University of Miskolc

Faculty of Mechanical Engineering and Informatics

Extended Sentence Parsing Method for Text-to-Semantic Application

Ph.D Dissertation

Author:

Walelign Tewabe Sewunetie
M.Sc in Computer Science

József Hatvany Doctoral School of Information Science, Engineering and Technology

Research Area
Applied Computer Science

Research Group
Knowledge Base and Data Analysis

Head of Doctoral School:
Prof. Dr. Jenő SZIGETI
Academic Supervisor:
Prof. Dr. László KOVÁCS

Miskolc, Hungary 2024

DECLARATION

The author hereby declares that this thesis has not been submitted, either in the same or in a different form, to this or any other university to obtain a Ph.D. degree. The author confirms that the submitted work is his own, and the appropriate credit has been given where reference has been made to the work of others.

Miskolc, 2024.

Walelign Tewabe Sewunetie

Acknowledgments

First and foremost, I would like to express my deepest gratitude to Almighty God. I extend my sincere gratitude to my supervisor, **Prof. Kovács Lászlo**, for his invaluable guidance, scholarly insights, and continuous support throughout the entire research process. His expertise and encouragement have been instrumental in shaping the trajectory of my work and elevating its quality.

I also acknowledge the financial support provided by Stipendium Hungaricum (SH), enabling the successful execution of my research and allowing me to concentrate on the academic pursuits integral to this dissertation. I am grateful to my colleagues and peers for their collaboration and shared insights, which have been pivotal in refining my ideas and contributing to the scholarly discourse.

To my wife, S/r Birtukan Yemane, my son Dagim and daughters Hilina and Amen, my mother Enana, and my brothers and sisters, I extend my deepest appreciation for their unwavering support, understanding, and patience. Their encouragement has been a constant source of motivation, and their belief in my abilities has sustained me through the challenges of doctoral research.

Finally, I express my gratitude to the countless individuals, mentors, and friends who have contributed to my academic and personal growth. Your influence has left an indelible mark on this dissertation and my overall scholarly development. This dissertation is the culmination of the collective efforts and support of these remarkable individuals, and for that, I am truly grateful.

Walelign Tewabe

Table of Contents

List of Figures

List of Algorithms

List of Tables

Abbreviations

ADJP adjective phrase. 42

ADVP adverb phrase. 42

AQG Automatic Question Generation. 1

BLEU Bilingual Evaluation Understudy. 17

CBOW Continuous Bag of Words. 37

ChatGPT Chat Generative Pretrained. 2

GER Grammar Error Rate. 18

IC Information Content. 58

IR Information Retrieval. 7

LCS longest common subsequence. 40

LDA Latent Dirichlet Allocation. 57

LLMs Large Language Models. 50

METEOR Metric for Evaluation of Translation with Explicit ORdering. 29

ML Machine Learning. 9, 12

MLP multilayer perceptron. 2

NER Named Entity Recognition. 6

NLG Natural Language Generation. 9

NLP Natural Language Processing. 1

NLTK Natural Language ToolKit. 26

NLU Natural Language Understanding. 9

NN neural network. 36

NP noun phrase. 42

NQG Neural Question Generation. 12

POS Part-of-Speech. 9

PP prepositional phrase. 42

QASC Question-Answer Sentence Composition. 42

RDF Resource Description Framework. 52

RNNs recurrent neural networks. 5

SPARQL SPARQL Protocol and RDF Query Language. 52

SRL Semantic Role Labeller. 14

VB verb phrase. 42

VQG Visual Question Generation. 12

Chapter 1

Introduction

1.1 Motivation

The field of Natural Language Processing (NLP) has advanced significantly in recent years, resulting in the creation of complex systems that indicate a deep comprehension of human language in terms of both generation and comprehension. The expanding significance of NLP in various applications has led to focused research endeavors on enhancing language model capabilities, particularly in the field of text-to-semantic applications[1]. With the increasing significance of language understanding in learning and information retrieval settings, Automatic Question Generation (AQG) has gained attention. AQG can accomplish two things: it makes it possible to create an automated assessment system from educational materials and improves information retrieval system performance.

The dissertation aims to solve important issues and innovate in the field of sentence parsing, driven by the deep significance of AQG in educational and information retrieval contexts. This investigation is sparked by AQG, which offers revolutionary possibilities for improving information retrieval systems and producing instructional content. The research is motivated by the need to expand and improve on current sentence parsing techniques, with a particular emphasis on their use in the larger context of text-to-semantic applications, as language comprehension becomes more and more important.

1.2 Objective and Research Question

The primary goal of this dissertation is to develop and evaluate extended sentence parsing methods that enhance the accuracy and efficiency of text-to-semantic applications, with a particular emphasis on AQG. To achieve this objective, the research seeks to answer the following key questions:

1. What are the limitations of existing sentence parsing methods in the context of AQG and text-to-semantic applications?

2. How can extended dependency parsing methods address the challenges faced in AQG?

3. What is the performance of multilayer perceptron (MLP)-based sentence parsing in comparison to traditional template-based approaches for AQG?

4. How can a hybrid parser, incorporating Chat Generative Pretrained (ChatGPT)-based sentence parsing, contribute to semantic graph induction in text-to-semantic applications?

1.3 Aim and Scope

This dissertation aims to contribute with novel insights and methods to the field of sentence parsing for text-to-semantic applications, focusing on AQG. The scope of the research encompasses the identification of limitations in current parsing methods, the development and evaluation of extended dependency parsing and multilayer perceptron-based models, and the exploration of a hybrid parser for utilizing ChatGPT-based techniques.

The intended contributions of this research include the advancement of sentence parsing techniques that improve the accuracy of AQG, ontology creation, and semantic graph induction. By addressing the identified limitations and proposing innovative methods, this dissertation aims to enhance the overall efficiency of text-to-semantic applications.

1.4 Dissertation Outline

The dissertation is organized as follows to facilitate the reader's understanding of the research exploration. In Chapter 2, an overview of NLP applications is provided, with an emphasis on the significance of sentence parsing, analysis, and semantic understanding. The chapter also explores AQG and semantic graph induction within the context of text-to-semantic applications. Chapter 3 delves into the challenges of dependency parsing for AQG, introduces the extended dependency parsing approach, and evaluates its effectiveness using predefined metrics. Chapter 4 introduces a novel MLP-based approach for AQG, compares its performance with template-based methods, and presents the research methodology along with key findings. Chapter 5 focuses on a Hybrid Parser for Semantic Graph Induction, explaining the basics of semantic graphs and their induction, detailing ChatGPT-based sentence parsing, and providing insights into the hybrid parser-based method through experiments and analysis. Chapter 6 explores the practical application of sentence parsing, including AQG, ontology creation, and semantic graph induction. Chapter 7 summarizes the significant findings and contributions of the dissertation. Throughout this systematic exploration, the dissertation aims to contribute valuable insights to the enhancement of text-to-semantic applications, particularly in educational and information retrieval contexts, thereby advancing the field of NLP.

Chapter 2

Background

2.1 Natural Language Processing and its Role

The dynamic field of NLP is focused on the relationship between human language and computers. It has evolved throughout time to become a cornerstone of AI, vital to a wide range of uses. NLP makes it possible for machines to understand, interpret, and produce language similar to humans, bridging the gap between computational systems and the many aspects of natural communication[2]. NLP is important because of its many applications, which affect how we use technology and handle large volumes of textual data. NLP has influenced many aspects of our digital lives, including sentiment analysis, machine translation, and speech recognition, information extraction, question answering, and engaging in lengthy conversations with humans[3].

Complex sentence parsing, analysis, and semantic understanding are at the core of NLP[4]. Investigating sentences into their constituent parts, interpreting syntactic patterns, and drawing conclusions from word choices are all part of parsing. Text comprehension is enhanced by this process, which is essential for understanding the meanings hidden in natural language. In the broader context of text-to-semantic applications, NLP plays a central role. The ability to parse sentences and extract semantic meaning is fundamental for tasks like AQG and semantic graph induction[5]. These applications require a deep understanding of language structures and relationships between entities, which NLP endeavors to provide.

2.1.1 Syntactic Analysis

A fundamental aspect of NLP is syntactic analysis, which aims to understand every aspect of grammar and sentence structure. Syntactic analysis techniques parse the source text to identify grammatical structures and dependencies[6]. By understanding the syntax of the text, the system can generate questions that maintain grammatical correctness and coherence with the source material. Understanding the placement of words in a sentence and their grammatical relationships depends heavily on this crucial component. Syntactic parser evolution: from classical rule-based techniques to modern probabilistic and neural network-based methodologies[7]. Prior rule-based systems were limited in their ability to handle the complexity of natural language and relied on human-crafted grammatical rules and linguistic knowledge, which helped to provide the foundation for basic understanding.

Probabilistic techniques emerged with machine learning advances, enabling parsers to make defensible judgments based on statistical patterns in language[8]. This was a breakthrough in syntactic analysis that made it possible for systems to adjust the differences between different linguistic forms. Syntactic analysis experienced a paradigm shift with the introduction of neural network-based techniques, such as transformer models and recurrent neural networks (RNNs)[9]. By capturing complex syntactic connections and patterns inside phrases, these models improve the precision and adaptability of language interpretation. They achieve this by utilizing deep learning techniques[10].

In this way, dependency parsing is one area of syntactic analysis that has gained popularity and is especially relevant to AQG. Creating inquiries that are both coherent and appropriate for the situation requires an understanding of word relationships. Dependency parsing makes it easier to determine the grammatical connections between words, which enables AQG systems to produce questions that follow the input text's syntactic structure[11]. Dependency parsing enhances the accuracy and importance of generated questions in the context of AQG. AQG systems can create questions that demonstrate a profound comprehension of the underlying structure and content of the input text by recognizing the syntactic connections between words[12].

2.1.2 Semantic Understanding

Simplified understanding is a key concept in NLP that overcomes language structure and focuses on revealing the meaning that is contained in words, phrases, and sentences[3]. Simplified comprehension explores the essential meanings that words and constructs within a particular context convey, whereas syntax analysis focuses on extracting the meaning and semantics of the text. It involves understanding implicit meanings, details, and plain meanings that all add to the complexity of human communication. Simplified analysis addresses the clarity of meaning as opposed to syntax, which is concerned with the structure and order of words. Capturing the intended simplicity of a text can be difficult due to ambiguities, context-dependent interpretations, and language's dynamic character[13].

The need for deep, simplified understanding in the process is emphasized in the imperative of deep, simplified understanding. When it comes to AQG, creat-

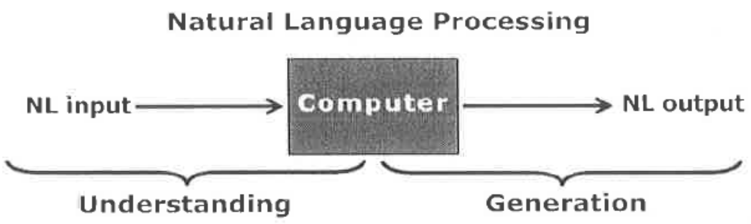


Figure 2.1: Natural Language Processing [15]

ing questions that are more than just grammatically correct necessitates a deep understanding of the fundamental simplicity. These methods aim to create questions that capture the core concepts, relationships, and semantic nuances present in the content[14]. Since deep simplified understanding allows the system to create questions that not only follow grammar rules but also show a sophisticated comprehension of the context, it is clear that deep simplified understanding is essential to AQG. This goes beyond superficial readings, making sure that the questions that are created are meaningful, contextually relevant, and consistent with the information that the input text intends to communicate. Simplified understanding essentially acts as a link between language's form and meaning. It establishes the imperative of incorporating deep, simplified understanding in the landscape of AQG, promising a more straightforward and contextually adept approach to question formulation. As shown in 2.1, NLP has two basic parts, i.e. natural language input called NLU and natural language output called NLG.

2.1.3 Named Entity Recognition

Named Entity Recognition (NER) is a critical component of NLP that focuses on identifying and classifying entities, such as names of people, places, organizations, and more, within a given text[16]. NER techniques contribute significantly to AQG by enabling the extraction of information about these entities and facilitating the creation of relevant questions. NER techniques analyze the text to locate and categorize entities. For example, in the sentence "Apple Inc. is headquartered in Cupertino, and Tim Cook is the CEO," NER would identify "Apple Inc." as an organization, "Cupertino" as a location, and "Tim Cook" as a person.

Once named entities are identified, NER can be used to extract attributes associated with these entities. This involves recognizing specific characteristics or details related to each entity. For instance, attributes for "Tim Cook" might include his position as the CEO. Questions can then be generated by inquiring about these attributes, such as "What is Tim Cook's role at Apple Inc.?" NER can also assist in identifying relationships between entities in the text. For example, recognizing that "Tim Cook" is associated with "Apple Inc." allows for the generation of questions about their relationship, such as "What is the connection between Tim Cook and Apple Inc.?"

In addition, NER outputs can be used to create templates for generating questions. These templates can be populated with specific entities, attributes, or relationships to form a variety of questions[17]. For instance: Template: "What is the [attribute] of [entity]?"

Question: "What is the headquarters location of Apple Inc.?" Template: "How is [entity1] related to [entity2]?"

Question: "How is Tim Cook related to Apple Inc.?"

Regarding the diversity in Question Generation, NER allows for the generation of diverse questions by identifying a range of entities and their attributes or relationships. This diversity enhances the comprehensiveness of the questions generated by the system.

2.1.4 Information Retrieval

Information Retrieval (IR) techniques play a crucial role in getting important information from texts. They do this by using processes like finding keywords, making summaries, and picking out important details[18]. One big part of IR is finding keywords. This is important because it helps create questions that are focused on the most important parts of the information[19]. By pulling out keywords, we make sure that the questions are about the most vital elements in the text. Summarization is also important in IR. It helps to shorten the important information, making it easier to create questions based on the main points.

IR is not just about finding information; it's also about keeping the context in mind. By identifying and getting information that is important in the given context, IR makes sure that the questions stay connected and make sense[20]. Bringing in relevant details from the source text while making questions not only makes them more connected but also adds to the overall meaning of the questions. The flexibility of IR techniques in dealing with different types of texts is important. This adaptability makes them useful in generating questions in various areas, showing that they can work well with different kinds of texts. To put it simply, the teamwork between IR techniques and the main focus on extended sentence parsing builds a strong base for creating questions automatically. These questions are not only well-connected but also make sense in different areas.

2.1.5 Ontology Creation

In this domain, the establishment of ontologies highlights the critical role that sentence parsing plays in natural language processing[21]. Sentence parsing serves as the foundation for the identification of entities, relationships, and contextual information by deconstructing complicated linguistic structures into their component pieces. This first stage forms the basis for creating a precise knowledge representation that captures the main ideas contained within the text. With the ability to extract, analyze, and organize information from unstructured textual input, NLP becomes a pivotal component in the vast array of applications[3]. As experts of understanding, NLP approaches are essential for deriving significant insights from unstructured text. This sets the foundation for the creation of complex semantic links and the systematic arrangement of information.

In this sense, NLP powers the process of building ontologies in addition to the complex language-parsing procedure. Together, these enable us to understand the linguistic structures and offer the foundation for a robust and meaningful knowledge representation, which advances our understanding of the semantic graph of unstructured textual material[22].

2.1.6 Semantic Graph Induction

Within text-to-semantic applications, Semantic Graph Induction and AQG emerge as pivotal elements. Semantic Graph Induction focuses on generating graphical structures that precisely depict the connections among concepts, entities, and actions within the text[23]. The synergy of various applications serves as catalysts, advancing semantic comprehension. The significance of Semantic Graph Induction is underscored by its critical role in advanced sentence-parsing techniques. It contributes substantially to crafting knowledge representations that are both structured and semantically rich[24]. Another key aspect of text-to-semantic applications is Semantic Graph Induction. This involves the construction of semantic graphs that visually represent the relationships and connections between entities and concepts within a body of text[25].

In the dynamic landscape of NLP, the concept of text-to-semantic applications represents a pivotal intersection where linguistic understanding meets semantic comprehension. This paradigm involves the transformation of raw textual data into a deeper semantic understanding, unraveling the intricate layers of meaning embedded in language[26]. The primary objective is to bridge the gap between textual information and its semantic representation, enabling intelligent systems to extract, interpret, and generate content with a nuanced understanding of context.The process of semantic graph induction helps in gaining a deeper comprehension of the semantic structure present in the textual input. Although there is potential for text-to-semantic applications, existing approaches frequently struggle to capture the complex semantic meaning within sentences[27].

2.2 Automatic Question Generation System

AQG relies on NLP as a foundational element, offering indispensable tools for automated language comprehension. This section discusses the evolution of NLP, tracing its journey from historical roots to contemporary methodologies that shape

AQG research. AQG, distinguished by its capacity to transform declarative statements into interrogative forms, facilitates a comprehensive exploration of the underlying material. NLP, situated at the intersection of artificial intelligence and linguistics, empowers computers to understand human language expressions. It encompasses two primary facets, as depicted in Figure 2.1 Natural Language Understanding (NLU) and Natural Language Generation (NLG)[15]. NLU involves grasping input expressed in natural language, while NLG centers on generating natural language responses.

The origins of NLP can be traced back to the mid-20th century, marked by early rule-based systems attempting to automate language understanding[28]. Despite their limitations, these systems laid the groundwork for subsequent developments. Text processing, a fundamental NLP task, involves tokenization, stemming, and lemmatization to deconstruct textual data, supporting tasks like Part-of-Speech (POS) tagging and syntactic analysis[29].

Within NLP, sentence parsing, analysis, and semantic understanding play pivotal roles. Sentence parsing involves the dissection of a sentence into its grammatical components, the establishment of word relationships, and the extraction of syntactic structure. To extract the meaning contained in natural language text, this procedure is essential[6]. Semantic understanding goes beyond syntax, understanding context and meaning through complexities, references, and implicit relationships in text. Particularly in text-to-semantic applications like AQG and semantic graph induction, these processes are essential to an extensive variety of NLP applications[30].

AQG, defined as the generation of syntactically sound, semantically correct, and relevant questions from diverse input formats, hinges on technological advancements[31]. The shift from manual to automated systems in education exemplifies this evolution, where traditional question generation by academicians has transformed. This section explores diverse techniques for question generation, emphasizing that the choice of technique depends on application requirements, source text quality, and desired question quality and diversity. Modern AQG systems often leverage a combination of techniques, incorporating NLP, ontology, and Machine Learning (ML) to enhance question relevance. The following discussion outlines common AQG techniques, showcasing the intersection of NLP with ontology and machine learning in question generation[32].

2.2.1 Template-Based Method

Template-based AQG techniques rely on predefined question templates that contain placeholders for specific information extracted from the source text. These templates serve as a structured framework for generating contextually relevant questions[33]. For example, consider a template tailored to information about Ethiopia, such as: ”What is the capital city of [country]?” In this template, the placeholder [country] can be filled in with the extracted information from the source text about Ethiopia. The system identifies the relevant content, such as "Ethiopia," and populates the template, resulting in a specific question like "What is the capital city of Ethiopia?" Template-based techniques offer a straightforward and systematic approach to question generation, making them especially useful for scenarios where specific types of questions need to be consistently generated from similar structures in the source text[12]. These techniques can be adapted to various domains and information types, providing a flexible solution for automatically generating questions tailored to the content at hand.

The template-based techniques use templates taken from the training set to generate questions, which are subsequently filled with specific topic items. A study employed a template-based method to create questions from key sentences and an adapted Text Rank to extract key sentences[34]. After analyzing their findings, they conclude that due to the limited number of templates, the sorts of queries generated are equally limited. The study work[35] in sentence-to-question transformation uses external rules or internal templates to modify the syntactic representation of the supplied input sentence. Syntax-based techniques use a standard procedure to decode a phrase to assess its syntactic structure, simplify it if necessary, recognize significant phrases, apply syntactic transformation rules, and request word substitution[36]. Jouault et al [37] proposed to construct semantics-based questions by accessing semantic information from the Wikipedia database to improve learners' self-directed learning, in contrast to approaches that use text as an input. Previous research on question generation has mostly relied on hard heuristic principles to convert a sentence into questions [38, 39].

AQG is defined as the process of producing relevant, accurate, and syntactically sound[40] questions from various input formats, and it is the current challenging research in the field of NLP. Among various AQG methods [41], a template-based method is the oldest, and it can be easily implemented. Template-based method use templates containing fixed text, and some placeholders that are populated from the given content. According to the thesis report [42], the templates are created by focusing on the events (actions, happenings), and existents (characters, settings).

In addition, I observed that most current templates ask about the subject, the predicate, and the object of the events and existents [38]. In the development of the template-based question generation method, first, I need to prepare a quality and representative dataset. While, due to the complexity of natural language structure, it is very difficult to create general-purpose templates. [43]. In this study, I have developed an open-ended template-based system.

After analyzing the structure of many sentence question pairs collected from various domains, I developed a template by considering most of the wh-question words (what, how, who, how many, where do, which, what kinds, when, why). Finally, I developed 48 open domain rule sets, according to this, I have also created our syntax-based template. For testing, I have prepared our 15-sentence-question pair test dataset collected from various sources. I used this test dataset for all systems that I analyzed for this study. From our test result, I observed that most generated questions have naturalness and meaningfulness problems. For this reason, I focused on evaluating by using naturalness and meaningfulness as basic parameters.

In the implementation of AQG, I used spaCy's improved Named Entity Recognition (NER)[44], which, how, how much, how many, which is capable of labeling more entity types, including money, dates/times, e.t.c, and it is important for selecting a proper wh-question word i.e what, who, where, when, [12]. Even if NER can identify the named entities automatically, [45] it is incapable to identify all categories of 'person', 'organization', 'location', and so on.

2.2.2 Rule-Based Method

Rule-based AQG techniques, employing dependency parsing, rely on predefined grammatical rules and templates to systematically generate questions. These rules provide instructions on how to extract or transform information from the source text into question forms[5]. For instance, consider a rule that identifies sentences beginning with interrogative words like "Who," "What," or "Where" and dictates their transformation into questions. In the context of information related to Ethiopia, a rule could identify a sentence like "Ethiopia is known for its rich cultural heritage," and based on the rule, generate the question "What is Ethiopia known for?" The use of dependency parsing enhances the sophistication of rule-based techniques by considering the syntactic dependencies between words in a sentence[46]. By understanding the grammatical relationships, these techniques can more accurately transform statements into questions. Rule-based approaches are valuable for maintaining grammatical correctness and generating questions that adhere to specific linguistic structures[47]. While they may require careful crafting of rules, they offer a systematic and interpretable method for AQG across diverse types of texts. Based on their domain, this template-based question generation can be categorized into two different groups i.e close domain and open domain[35].

In closed-domain template-based question generation, predefined question templates are designed specifically for a narrow or well-defined domain[48]. These templates are crafted to capture the particular types of information present in that domain. For instance, in a closed domain related to sports, a template could be structured as "Who won the [event] in [year]?" The system would then fill in the placeholders, such as the specific sports event and the corresponding year, by extracting information from the source text. This approach is effective when dealing with texts that consistently contain certain types of information, allowing for the creation of tailored question templates that suit the characteristics of the closed domain. Contrastingly, open-domain template-based question generation involves more general and adaptable templates to a broader range of topics or domains[49]. These templates are designed to handle various types of information and may have placeholders that can be filled with different types of entities or attributes. For example, a template like "What is the significance of [concept]?"

is more flexible and can be applied to diverse subjects. Open-domain templates are beneficial when dealing with texts spanning various topics, making creating specific templates for each domain impractical. The adaptability of open-domain templates allows for a more versatile approach to generating questions across a wide range of content and domains.

2.2.3 Neural Network-based Method

Researchers from other disciplines have recently become interested in the research topic of AQG for educational objectives. Cohen [50] proposed that the substance of a question can be represented as an open formula with one or more unbound variables in one of the first works on questions. While question generation research has been done for a long time, the use of AQG for educational purposes has attracted the attention of several academic communities in recent years[51]. Questions have also been a major topic of study in computational linguistics where models of the transformation from answers to questions have also been developed[52]. Previous studies have specifically addressed the generation of questions for educational objectives, as evidenced by Heilman et al[53], who showed that a combination of AQG and manual correction can be more time-efficient compared to solely manual authoring. Authors [54] created an automated reading tutor that uses AQG to help students improve their comprehension skills while reading a text. They looked at ways to automatically construct self-questioning instruction based on assertions in narrative texts about mental states (e.g., belief, intention, hypothesis, emotion). This system employs a template-based approach to question generation.

The study work [55] suggested that a neural network method for generating factual questions from structured data instead of producing questions from texts. Authors [56] performs a preliminary investigation on question creation from text using neural networks, dubbed the Neural Question Generation (NQG) framework, to produce natural language questions from text without the use of pre-defined criteria. The advanced NLP techniques employed for the textual question creation include NLU and NLG[57]. First, the system has to understand the input text which is NLU, and then it has to generate questions also in the form of text that is NLG. The article[57] presented a system for generating factual inquiries from unstructured material. They combine numerous Machine Learning (ML) algorithms with classical linguistic methodologies based on sentence patterns. In the disciplines of NLP and computer vision, generating natural language queries for picture understanding is a hot topic[58]. Regarding the implementations of the learning modules the most dominant solution is neural network based architecture, specially the MLP and RNNs[59]. ML was primarily employed in the Visual Question Generation (VQG) method to produce image captions. Using NLP algorithms, the image caption is converted into a question. VQG blends NLP, which allows the inquiry to be generated, with computer vision techniques, which allow the image's content to be understood[57].

The expected benefits of question generation from a given text using a neural network are: the training data should require little or no human effort and should reflect commonly-asked question intentions; the questions are generated based on natural language passages and should be of good quality, and the generated questions should be useful to QA tasks. According to previous research[59], neural-based AQG obtains large-scale, high-quality training data via the Community-QA (CQA) website. The use of deep neural networks to extract target responses from a given article or paragraph and generate questions based on the target answers is known as NQG[60].

2.2.4 Semantic Based Method

The paradigm of text-to-semantic applications encompasses various methodologies aimed at extracting deeper meaning from textual content[61]. This application goes beyond traditional question-generation approaches by incorporating semantic understanding to formulate questions that reflect a more profound grasp of the underlying meaning within the text. In the context of semantic-based AQG, the emphasis is on leveraging advanced language understanding to extract not only syntactic structures but also the semantic details present in the text[62]. This approach aims to generate contextually relevant questions and align with the deeper meaning embedded in the content.

Within the semantic-based framework, AQG goes beyond conventional syntactic analyses. It leverages advanced language understanding, including semantic relationships, to formulate questions that are not only grammatically correct but also contextually relevant and aligned with the deeper semantic meaning encapsulated in the text[63]. The integration of sentence parsing plays a pivotal role in semantic-based AQG. Advanced parsing methods contribute to the extraction of semantic structures, entities, and relationships within sentences[41]. This, in turn, enhances the precision and context-awareness of the questions generated, moving beyond surface-level understanding to capture the nuanced semantics of the text. Analyzing sentences, especially when improved for semantic comprehension, greatly enhances the accuracy of generating contextual questions[64]. This is achieved by capturing the entities and relationships that are contextually relevant within the sentences.

2.2.5 Challenges of Automatic Question Generation

AQG is a challenging task that involves creating natural and contextually relevant questions from given content[11, 12]. Annotated datasets for training AQG models are often limited. This scarcity makes it challenging to build models that generalize well across different domains and contexts. Addressing these challenges involves a combination of advanced natural language processing techniques, machine learning models, and domain-specific knowledge[3]. Ongoing research in these areas aims to enhance the capabilities of AQG systems. The process of automatically generating factual questions for reading assessment involves several computational and linguistic challenges. As shown in Figure 2.2, three major problems have been identified and addressed in the process to enhance the effectiveness of question generation, i.e., sentence simplification, question transformation, and question ranking[65].

1. **Sentence Simplification**

   One key challenge lies in the need for effective sentence simplification; specifically, I can say target selection. "Target selection" in the context of AQG refers to the process of identifying specific elements or entities within a given text that will become the focus of generated questions[66]. It involves selecting the key information, often in the form of named entities, attributes, or relationships, around which questions will be formulated. The goal is to choose targets that are relevant to the overall purpose of the question generation, whether it's for educational assessments, information extraction, or other applications[67].

   The selection of targets is influenced by the objectives of the question generation system and the nature of the text being analyzed. For instance, in educational contexts, targets may include important concepts, events, or characters within a passage[68]. In information extraction applications, targets could be specific entities, attributes, or relationships that the system aims to query or elaborate upon. The effectiveness of the question-generation process relies heavily on accurate and contextually relevant target selection. It involves techniques such as NER, syntactic analysis, and semantic understanding to identify the entities or information considered crucial for constructing meaningful and coherent questions[69]. Successful target selection contributes to the overall quality of the generated questions, ensuring they align with the goals of the task and provide valuable insights or assessments based on the selected targets[68].

   Understanding the context of the given text is crucial for generating relevant questions. Ambiguities, implicit information, and context-dependent meanings pose challenges in accurately capturing the context[13]. A lot of work has been done in this field with the help of various tools like Semantic Role Labeller (SRL), POS Tagger, and Annotated corpora tools[70]. In this section, I present the difficulties of automatic target selection using unannotated document sources. As a result of the analysis, I identified the following key issues: hard to measure the topic relevance, the large segmentation of the information content, and the incomplete background knowledge bases [71]. Question generation depending upon the target complexity, can be mainly categorized into two categories, deep question generation and shallow question generation. Deep QG generates deep questions that involve more logical thinking (such as why, why not, what-if, what-if-not and how questions) whereas shallow QG generates shallow questions that focus more on facts (such as who, what, when, where, which, how many/much and yes/no questions).

However, most of existing approaches to question generation have focused on generating questions from a single sentence, relying heavily on syntax only shallow semantics [72]. A problem with this approach is that the majority of questions generated from single sentences tend to be too specific and low-level to properly measure learners' understanding of the overall contents of text. In other words, what is assessed by such question generation system ends up essentially being the ability to compare sentences, just requiring learners to find a single sentence that has almost the same surface form as a given interrogative sentence. Results of simple sentence comparisons do little to contribute towards the goal of assessing learners' reading comprehension. Considering the different question types, the multi-choice question is the most widely used question format in the AQG systems. The main benefit of this format is the simplicity of the evaluation and the relatively high unambiguity. In the construction of multi-choice questions, I face the following challenges:

- selection of the target sentence
- selection of the target concept / phrase
- selection of the distractors.

In our study, research is mainly focused on the sentence level. I present this approach through different methodologies.

The engine selects elementary sentences without complex clauses. Using the Charniak parser [73], a syntactic tree is constructed for the selected sentence. Based on the associated POS and NE tagged information, the subject, object, preposition and verb parts are located in the sentence. Then, the POS parts are assigned to one of the following classes: Verb, Human, Entity, Location, Time, Count. The engine uses 90 predefined sentence schemas, like "Human Verb Entity" or "Human Verb Human Time". The sentence schema infer also a question type like "Whom/Who" or "Who/Where". According to the test experiments on TREC-2007 (Question Answering Track) [74] dataset, the engine could achieve a 0.12 - 0.55 recall ratio depending on the question type. Another direction is to select also the target sentence from a larger text. A good example is the Python project Wikipedia-question-generator [75]

The project uses besides Wikipedia also the WordNet ontology database to determine the synonyms of the target concept. Having a topic, the engine will generate a multi-choice question. For example, for the keyword 'Tony Bennett', I get the following question:

{ *"question": "Bennett is also an accomplished ________, having created works under the name Anthony Benedetto that are on permanent public display in several institutions.", "answer": "painter", "similar_words": ["classic", "classicist", "constructivist", "decorator", "draftsman", "etcher", "expressionist", "illustrator"]* }

The target sentence is selected on a relatively simple algorithm:

- Only the summary section is considered for sentence selection; at other parts, the sentences are partially and strongly related.
- Omit the first sentence of the summary, it is too straightforward to make interesting trivia.
- Avoid sentences starting with an adverb, as these are strongly depending on the previous sentences.
- Select the first common noun in the sentence as target concept.

According to the test experiments, these simple approaches can provide a relatively good result.

2. **Question Transformation**

The process of transforming information from a given text into a question format presents another significant challenge. This involves understanding the context, identifying key elements, and structuring the question appropriately. Achieving question transformation requires advanced NLP techniques to ensure that questions generated are contextually relevant, grammatically correct, and effectively elicit the desired information from the reader[48].

The success of these approaches hinges critically on the existence of well-designed rules for declarative-to-interrogative sentence transformation, typically based on deep linguistic knowledge. To improve quality over a purely rule-based system[53] introduced a generate-and-rank approach that generates multiple questions from an input sentence using a rule-based approach and then ranks them using a supervised learning-based ranker. Although the ranking algorithm helps to produce more acceptable questions, it relies heavily on a manually crafted feature set, and the questions generated often overlap word for word with the tokens in the input sentence, making them very easy to answer.

3. **Question Ranking**

After generating a set of questions, determining their relevance and appropriateness for a given context is a non-trivial task. Question ranking involves assessing the quality of generated questions and prioritizing them based on factors such as clarity, informativeness, and relevance to the text. Developing effective algorithms for question ranking is crucial to providing users with a curated set of questions that align with the goals of the reading assessment and provide a meaningful evaluation of comprehension.

Recently, researchers from multiple disciplines have been showing their common interest in AQG for educational purposes. In this paper, I review the state of the art of approaches to developing educational applications of question generation. I conclude that although a great variety of techniques on AQG exists, just a small amount of educational systems exploiting question generation has been developed and deployed in real classroom settings. I also propose research directions for deploying the question technology in

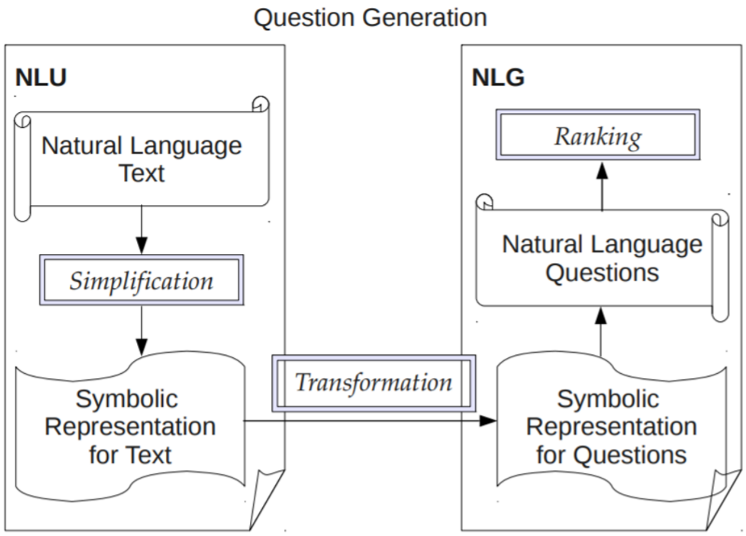


Figure 2.2: Three major problems in the process of question generation [65]

computer-supported educational systems [53]. Several challenges are associated with AQG, and addressing them is crucial for the development of effective question-generation systems. Here are some main challenges:

2.3 Evaluation Metrics of Automatic Question Generation

Defining appropriate evaluation metrics for AQG is challenging. Metrics should consider aspects such as question relevance, diversity, and grammatical correctness. Developing comprehensive and widely accepted evaluation criteria is an ongoing challenge[76]. Here are some key evaluation metrics commonly used for assessing AQG:

Relevance Metrics Relevance in generated questions is assessed through several key metrics. Precision and Recall evaluate accuracy by comparing generated questions to a reference set, with precision representing the ratio of correctly generated questions to the total generated, and recall representing the ratio of correctly generated questions to the total reference questions. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)[5, 76], primarily designed for summarization tasks, ROUGE measures the overlap of n-grams between the generated and reference questions. It can be adapted for AQG evaluation. The F1 Score combines precision and recall into a single metric, offering a balanced measure of relevance. Additionally, Bilingual Evaluation Understudy (BLEU)[77], originally designed for machine translation, assesses the overlap of n-grams between generated and reference questions, providing a widely used metric for natural language generation tasks[78]. Together, these metrics comprehensively evaluate the relevance of the generated questions.

Diversity Metrics The evaluation of diversity in generated questions employs two key metrics. The Diversity Score measures the extent of diversity among the questions, with methods like cosine similarity [79] or distance metrics used to measure dissimilarity between questions. Complementing this, Distinct Metrics quantify the number of distinct unigrams or n-grams present in the generated questions, with higher values indicating increased diversity. Together, these metrics provide a nuanced assessment of the diversity in the generated question set.

Grammatical Correctness Metrics Grammatical correctness is assessed through two key metrics. The Grammar Error Rate (GER) [80] measures the percentage of generated questions with grammatical errors, and it can be calculated using automated tools like language parsers or grammatical checkers. Complementing this, the Fluency Score evaluates the naturalness and coherence of the generated questions, relying on subjective assessments from either human evaluators or language models. Together, these metrics provide a comprehensive evaluation of the grammatical accuracy and linguistic fluency of the generated content.

Question Type Metrics The metric of type accuracy evaluates the precision in generating questions of various types, such as factual, reasoning, or opinion-based [76]. This assessment relies on annotated data with labeled question types to measure the accuracy of the generated questions in aligning with their designated types.

Human Evaluation Human evaluation employs crowdsourcing, wherein human evaluators assess the quality of generated questions, considering factors such as relevance, diversity, and grammatical correctness. To ensure consistency in evaluation, measures of inter-annotator agreement are employed. Additionally, informativeness evaluation involves humans assessing how well the generated questions capture essential information from the source text. Context appropriateness evaluation focuses on determining whether the generated questions are contextually appropriate and meaningful.

Developing a comprehensive evaluation framework may involve a combination of these metrics, as no single metric can capture all aspects of question quality. Researchers and practitioners often use a combination of automated metrics and human evaluations to obtain a more holistic understanding of AQG system performance. Ongoing research aims to refine existing metrics and develop new ones to better address the unique challenges of evaluating question-generation tasks.

2.4 Shortcomings of Current Sentence Parsing

Online learning is becoming more common and it allows students to access online materials anywhere at any time. In this information era, many organizations and institutions provide a variety of training alternatives to their employees or learners[81]. Due to the radical expansion of the internet over the past two decades, more individuals have got access to online resources [82]. As a result of this development, e-learning is quickly gain popularity as a teaching method, particularly in higher education. The assessment phase, which is used to gauge the academic success of the students, is one of the key difficulties in e-learning. The process of automatically creating questions from different inputs, such as raw text, databases, semantic representations, ontologies, taxonomies, knowledge bases, images, or audio and videos known as AQG[83].

Researchers [12], from various fields have recently demonstrated a common interest in using AQG for educational reasons. An important function in educational assessment played by AQG is to generate questions and their answers [84]. According to the survey [85], the main challenges in AQG development are the following issues: question generation from multiple sentences, short and long-type answer assessment, question generation, and assessment using machine learning tools. In this paper, I focus on the presentation of a neural network-based approach.

Despite the progress in NLP, current sentence parsing methods encounter difficulties in effectively capturing the detailed semantics of natural language text. Challenges arise from ambiguity, context-dependent meanings, and the dynamic nature of language. This study aims to overcome these limitations by introducing extended sentence parsing methods, to improve the precision and efficiency of text-to-semantic applications.

These challenges are particularly evident when it comes to accurately categorizing adverb subtypes like Time, Place, Manner, Degree, and Frequency, where common shortcomings persist. Ambiguity is a notable issue, as adverbs can often have multiple interpretations. For example, the adverb "fast" can indicate both manner (He runs fast) and degree (He is fast). Context Dependency further complicates matters, as understanding the context of a sentence is crucial for accurate adverb subtype categorization. Parsing tools may struggle to grasp the full context, leading to misclassifications, as seen with the adverb "soon," which can indicate both time (I will leave soon) or degree (I will finish the work soon).

Complex Sentence Structures add another layer of difficulty, especially in sentences with intricate constructions. Parsing tools may find it challenging to discern specific roles in complex constructions where adverbs modify different elements. The issue of Newly Coined Words and Expressions arises due to the dynamic nature of language, where parsing tools may not be equipped to handle newly coined adverbs or those used in novel ways. Lack of Pragmatic Understanding is also a concern, as parsing tools may struggle to infer implied meanings or the speaker's intent, impacting their ability to identify adverb subtypes correctly.

Disambiguating Homonyms is a common challenge, as adverbs that share the same form but have different meanings can create difficulties in categorization. For instance, "hard" can indicate manner (He works hard) or degree (The problem is hard), and disambiguating these cases is not always straightforward. Lastly, Dynamic Language Evolution poses a continuous challenge, as language is constantly evolving, and parsing tools may struggle to keep up with the rapid changes in usage patterns and linguistic trends. Addressing these shortcomings requires ongoing advancements in NLP techniques, improved training datasets, and a deeper understanding of the contextual and pragmatic aspects of language. Researchers and developers are continuously working to enhance the capabilities of parsing tools to overcome these challenges.

As I embark on a journey through extended dependency parsing, multilayer perceptron-based models, and hybrid parsers, the subsequent chapters will unravel innovative solutions designed to overcome the shortcomings of current sentence parsing methods. The ultimate goal is to contribute to the refinement of AQG and semantic graph induction, fostering a deeper and more accurate understanding of textual information. While the potential of semantic-based AQG is substantial, existing methodologies face challenges in accurately capturing nuanced semantic meanings. The limitations of current sentence-parsing techniques, particularly in the context of semantic understanding, highlight the need for advancements in the field.

2.5 Conclusion

This chapter provides a comprehensive exploration of NLP and its various roles. It delves into essential components such as syntactic analysis, semantic understanding, named entity recognition, information retrieval, ontology creation, and semantic graph induction. In addition, the depth analysis of AQG Systems covers methods like template-based, rule-based, neural network-based, and semantic-based approaches. Challenges within this domain, including sentence simplification, question transformation, and question ranking, are discussed. Furthermore, the chapter addresses the crucial aspect of evaluating AQG through specific metrics. and the existing shortcomings in current sentence parsing techniques. As the chapter concludes, it sets the stage for further exploration and development in the field of NLP, emphasizing the need for addressing challenges and refining methodologies in AQG. .

Chapter 3

Sentence Parsing with Extended Dependency Parsing

3.1 Challenges of Dependency Parsing for AQG

As the field of AQG continues to evolve rapidly, future research should focus on developing more advanced models that can generate a wider range of questions, especially for complex sentence structures. The current system serves as a valuable foundation for further advancements in AQG, offering potential applications beyond educational settings. The implications of this research extend beyond the immediate scope, providing a stepping stone for future AQG developments. The following sections will demonstrate how the integration of dependency parsing, NER, adverb, and noun subtype analysis improves the identification of target concepts and question words. Furthermore, I discuss how these improvements impact the quality of the generated questions.

This study proposes an innovative Extended Dependency Parsing approach. This method extends traditional dependency-parsing techniques by incorporating additional contextual and semantic information. By enriching the parsing process, the aim is to improve the accuracy of dependency-based structures, consequently enhancing the quality of questions generated through AQG. One notable challenge of AQG is the complexity of generating meaningful questions from parsed dependencies. Achieving this requires addressing issues such as ambiguous syntactic structures and ensuring the coherence of the generated questions.

3.2 Query Generation using Similarity Ranking with Embeddings

An alternative strategy for AQG is utilizing similarity ranking with embeddings. This method involves assessing the similarity between parsed dependencies and predefined templates to formulate relevant questions. In this approach, I used direct sample selection to predict the corresponding query sentence. The method selects the sample triplet (s_i, p_i, q_i) with the highest similarity with the query sentence input pair (s_i'', p_i''). The similarity value is measured with a semantic similarity using the language embedding model. The main benefits of this similarity approach are:

- context sensitivity
- high-level independence from the surface layer
- simple application.

The formal model of the proposed system:

1. Construction of the sample dictionary from the training set $T : \{t_i\}$ where t_i is equal to a triplet (s_i, p_i, q_i). The first component is the input sentence; the second component is the position of the target word, and the third tag is equal to the generated query sentence
2. The target input is a pair (s'', p'').
3. The input and target sentences are transformed into a lemmatized form to increase the unambiguity of the sentence form.
$$s_i' = lemma(s_i), q_i' = lemma(q_i), s''' = lemma(s_i'') \tag{3.1}$$
In this preprocessing step, every variant of the same word is mapped to the same word format.
4. Calculate the embedding level similarity for the dictionary samples:
$$c_i = embedding\_similarity(s_i', s'') \tag{3.2}$$
This measure takes the whole sentences into account, but it does not consider which is the target word in the sentences.
5. Sorting the dictionary sample in decreased order of similarity.
6. Process the dictionary samples in this order to measure a finer similarity value considering also the target word selection.

7. I calculated word-level similarity matching for the components in the investigated pair (s'_i, s''), each word is assigned to the nearest partner word using a pairing algorithm.

8. Select the first sentence in this order, where the target words are pairs in the finer-level similarity matching.

9. Calculate the output query sentence based on the winner sample using corresponding substitutions.

For the evaluation of the engine accuracy, I have used only one measure, the acc_2 accuracy value.

3.2.1 Testing and Evaluation

In the efficiency comparison framework, the following four methods were implemented:

- AQG using sentence matching with syntax-templates (M_ST)
- AQG using similarity ranking with embeddings (M_SE)
- AQG using MLP neural network without embeddings vector (M_NN)
- AQG using MLP neural network with embeddings vector (M_NNE).

The training and test datasets were generated from three sources: synthetic data generation, manual data generation, and external databases.

In the first generation approach, I have constructed a world domain with a limited number of words, and the sentences were constructed with the application of sentence templates. Here are some examples of the training items:

```
s: Peter is going to farm afternoon
p: 2
q: What is doing Peter afternoon

s: Anna is travelling to city today
p: 3
q: Where is Anna travelling to today
```

The algorithms were implemented in Python framework using the following standard libraries: NLTK, spaCy, and Keras.

Regarding the implementation of the method using syntax templates, I have used the template ruleset presented in [40].

The tests were evaluated for the following training set sizes: Small: $N = 80$, Medium: $N = 800$, Large: $N = 8000$

The test results for acc_2 measure are summarized in Table 3.1- Table 3.3. The cells show the measured accuracy in percentage (%). The presented values are the average values of 6 measures.

Table 3.1: Accuracy with small training set

N	E	M_ST	M_SE	M_NN	M_NNE
80	5	46	85	44	43
80	20			42	41
80	100			38	39
80	300			36	37

Table 3.2: Accuracy with medium training set

N	E	M_ST	M_SE	M_NN	M_NNE
800	5	47	92	47	48
800	20			48	49
800	100			51	51
800	300			52	51

Table 3.3: Accuracy with large training set

N	E	M_ST	M_SE	M_NN	M_NNE
8000	5	47	92	47	48
8000	20			51	51
8000	100			52	51
8000	300			53	52

The test results show some interesting experiences, namely:

- There is no significant differences between the tested MLP neural network models with embeddings and without embeddings. This fact shows that the syntax level attributes (like POS value, syntax role) have enough power to achieve a moderate accuracy, the embedding vector will carry information similar to the syntax-oriented attributes.
- The embedding module is very effective in the AQG method using similarity matching. This method could achieve the best results, over 90% accuracy. The main benefit of this approach, that it can detect the dictionary items

most similar to the input sentence for query generation. In this case, the similarity measure covers also the semantic viewpoint, too.

- The syntax-template approach could provide an accuracy similar to the accuracy value of the neural network approach, especially for smaller data sets and smaller epoch numbers. The main problem of this approach is that it is very rigid and it is very time-consuming to extend the applied ruleset.

3.3 Extended Dependency Parsing

Dependency parsing, a fundamental aspect of syntactic analysis, plays a crucial role in understanding the relationships between words in a sentence. However, when applied to the task of AQG, dependency parsing encounters specific challenges that impact its effectiveness. This chapter explores these challenges, highlighting the complexities of dependency parsing in generating meaningful questions from textual content. Dependency-based syntax, with functional relations, became more widely used in computational models compared to the phrase-structure-based constituency[86]. It identifies semantic connections between words in a sentence. It retrieves the sentence's syntactic structure from a linear sequence of word tokens by analyzing the relationships between words and determining each word's syntactic category. Recently, dependency-based syntactic parsing has gained popularity[87]. The increased interest in dependency-based parsing has led to research into various parsing algorithms. The key difference between dependency and syntactic parsing is that dependency parsing builds a parse tree, while syntactic parsing constructs a syntax tree [88].

Dependency parsing is a crucial task in NLP, and recent years have seen significant advancements in this field [89]. According to Kübler et al. [88], dependency parsing models can be broadly classified into two major groups: grammar-based dependency parsing and data-driven dependency parsing. Grammar-based models are based on formal grammar and can be further divided into context-free dependency parsing and constraint-based dependency parsing. In contrast, data-driven approaches differ in the type of parsing model adopted, the algorithms used to learn the model data, and the algorithms used to parse new sentences with the model.

Data-driven dependency parsing models can be further categorized into transition-based and graph-based dependency parsing models[90]. Both transition-based and graph-based models are developed using supervised machine-learning techniques from linguistic data. Transition-based dependency parsing, also known as shift-reduce parsing, learns a model for scoring transitions from one parser state to the next, conditioned on the parse history. Parsing is then performed by greedily taking the highest-scoring transition out of every parser state until a complete dependency graph is derived. Figure 3.1 shows an example of transition-based dependency parsing for the sentence "Budapest is the capital of Hungary."

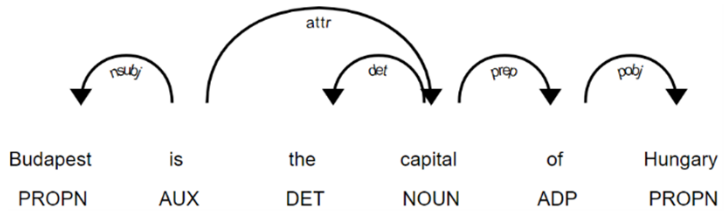


Figure 3.1: Dependency parsing example (Budapest is the capital city of Hungary)

The second important type of dependency parsing is the graph-based dependency parsing models, which were introduced by MacDonald[91]. These models learn scoring functions in one shot and then perform an exhaustive search over the entire tree space for the highest-scoring tree. Currently, there are various dependency parsing tools available in Python that can be used to analyze sentence structure, some of the popular tools are spaCy, Natural Language ToolKit (NLTK) with Stanford CoreNLP, and Stanza.

This study opens new avenues for AQG by addressing key limitations in existing systems. The integration of word2vec solutions for rule-matching calculation enhances the flexibility of this system, making it applicable across various domains[92]. Extended rule-based dependency parsing involves the use of explicit rules and linguistic knowledge to perform a more detailed analysis of sentence structure, capturing not only basic syntactic dependencies but also additional semantic and contextual relationships. While traditional dependency parsing typically relies on grammatical rules to establish syntactic dependencies, extended rule-based dependency parsing goes beyond this by incorporating additional linguistic phenomena. Extended Dependency Parsing involves incorporating extra linguistic information or features to augment the depth and sophistication of the analysis. The process encompasses various aspects, starting with fundamental syntactic dependency rules that establish grammatical connections among words, such as subject-verb and verb-object relationships. Additionally, Semantic Dependency Rules are introduced to expand the rule set, encompassing the identification of semantic dependencies.

Incorporate NER rules to identify and label named entities such as persons, organizations, locations, etc. These named entities can then be included in the dependency structure to represent relationships involving these entities. Include rules to handle potential errors or ambiguous cases. This might involve fallback mechanisms or heuristics to address challenges that arise during parsing. Building an extended rule-based dependency parser requires expertise in linguistics, as well as a deep understanding of the specific linguistic phenomena relevant to the task or domain. It involves crafting rules that capture the intricate relationships within sentences and across textual units. While rule-based approaches provide transparency and interpretability, they may require constant refinement to handle the complexity of natural language.

3.3.1 Method

Our methodology is grounded in a strategic fusion of dependency tree parsing and NER techniques. These choices are underpinned by their proven effectiveness and versatility in addressing the core challenges outlined in the introduction. Here, I have provided the necessary details, algorithms, and techniques to allow readers to confirm and replicate our findings. In this regard, dependency tree parsing is a cornerstone of our approach and provides the means to analyze the grammatical structure of sentences by establishing dependency relations between words[93]. The choice of dependency parsing is justified by its inherent ability to handle various language constructs and ambiguous inputs effectively. NER is another integral component of our methodology. NER automates the extraction of valuable information from unstructured natural language documents by categorizing named entities into predefined groups[94]. These groups include person names, organizations, locations, and more. Though conventional, I emphasize that these choices are essential to this method.

As reported by Mazidi et al. [95] reference, dependency labels provide valuable information for extracting the meaning of the relationships between words. This technique constructs a tree structure that represents the syntactic dependency relationships between words, allowing us to identify the key semantic building blocks of the sentence. However, it was recognized that dependency parsing alone is insufficient for AQG, and additional tools like NER needed to be incorporated. GATE, OpenNLP, and spaCy are notable NER platforms.

For this study, spaCy NER was employed as a fast, statistical, and open-source named entity visualizer. The system assigns labels to groups of contiguous tokens, which encompass named or numerical entities, including person, organization, language, and event, among others. Our proposed system is illustrated in Figure 3.2 and is categorized into distinct modules: Pre-processing: The initial module, involving the removal of stop words and tokenization of the remaining words from the input sentence. NER, POS, and Dependency Parsing: The subsequent modules process the tokenized data, identifying named entities, extracting POS tags, and performing dependency parsing. These elements form the foundation for subsequent stages. The output of this module serves as input for the NER, POS, and dependency parsing modules. The NER module identifies named entities within the input, while the POS module extracts the noun components of the sentence, which are also essential for the ruleset mapping and question generation stages.

The Ruleset adopted from previous work [96] is extended to include named entities, POS tags, and dependency parsing. This enhancement acknowledges the importance of these elements for generating high-quality questions. However, the main limitation of the ruleset was its lack of categorization for adverbs and noun types. To address this limitation, I have developed Algorithm 1 and Algorithm 2, which depict the essential steps in our methodology: Algorithm 1: Ruleset Mapping for Question Generation: This algorithm maps rules to dependency tag lists and selects the best matching rule. It is an essential component of our innovative

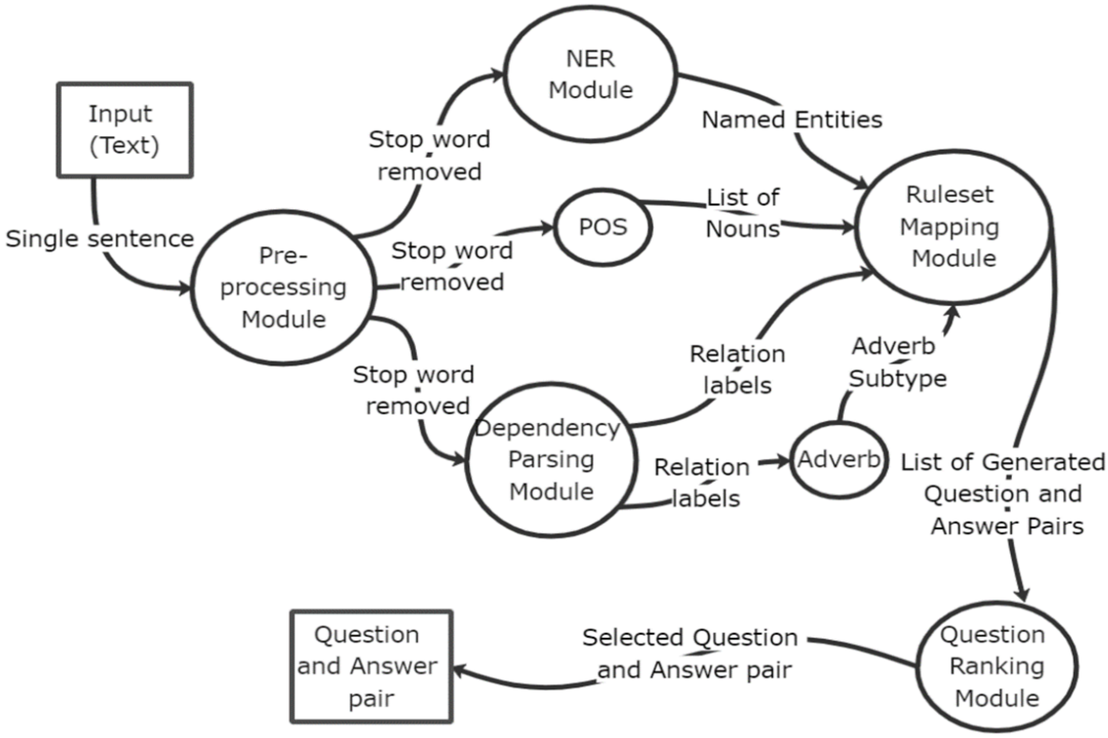


Figure 3.2: Proposed system block diagram

approach. Algorithm 2: Question Word Selection for Question Generation: This algorithm determines the appropriate question word (Wh_QTypeWord) based on inputs, including NER, adverb subtype, noun subtype, and dependency tags. This step contributes significantly to question generation.

In response to the limitations of conventional rule-based systems, our methodology innovatively integrates word2vec[97] a powerful word embedding technique. This integration augments the flexibility and effectiveness of our system, making it applicable across diverse domains. The authors noted that the state-of-the-art best match analysis calculation is commonly used to perform rule-set matching. Nevertheless, this mechanism for selecting the best match is rigid, and there are numerous scenarios in which sentences may express the same meaning but are written differently.

A distinctive feature of our approach is the inclusion of adverb subtypes (Time, Place, Manner, Degree, and Frequency)[98] and noun subtypes (Human, Animal, and Thing) for question generation. These subtypes play a pivotal role in crafting high-quality questions. I have provided comprehensive tables (Table 3.4 and Table 3.5) that detail the combinations of these subtypes with their corresponding question words.

Algorithm 1 Ruleset mapping for question generation algorithm

```
function RulesetMapping(Ruleset, List_of_Sent_DependencyTag)
    Input: Ruleset, List_of_Sent_DependencyTag
    Output: Question, Answer
    QuestionList ← empty
    for R ← Rule to Ruleset do
        Sim ← similarity(Rule, DependencyTagList)
        if Sim is in BestSimilarityScore then
            WinnerRule ← Rule
            BestSimilarityScore ← Sim
        end if
    end for
    QuestionList ← apply(WhQTypeWord, WinnerRule, DependencyTagList)
    Return QuestionList
end function
```

Table 3.4: Noun SubTypes and Corresponding Question Words

Noun SubType	**Question Word**
Human	Who
Animal	What
Thing	What

3.3.2 Evaluation Metrics

In recent years, numerous models, and evaluation techniques for AQG have been presented in the literature. Various metrics, including BLEU, ROUGE, and Metric for Evaluation of Translation with Explicit ORdering (METEOR), have been utilized to evaluate the effectiveness of AQG systems by measuring the similarity of the generated questions to the reference questions[5]. However, the evaluation process also gave significant importance to the answerability and naturalness of the questions, as they are essential factors in determining the quality of generated questions. A human evaluation was conducted to assess the answerability of the generated questions, with the significance of question types (Wh-types), named entities, and content words (often relations) being determined in various AQG tasks. Furthermore, the grammar structure and naturalness of the generated questions were considered fundamental parameters in human evaluation.

The evaluation process involved a Google form questionnaire that allowed participants to rate the generated questions on a scale of 1 to 5, where 1 denoted poor and 5 indicated excellently. The overall outcome of the human evaluation was encouraging, with a score of 3.67. The researchers compare system-generated questions with human-generated questions and use automatic evaluation techniques. The results of the evaluation, as it is presented in Table 3.6, suggest that the system performs well, especially in short sentences. The average BLEU-N score of 0.718 indicates that the system-generated questions have a reasonable level of

Algorithm 2 Algorithm for Question word Selection for Automatic Question Generation

```
function QuestionGeneration(Ruleset, List_of_sent_NER, List_of_sent_AdverbSubType, List_of_sent_NounSubType, List_of_sent_DependencyTag)
    Input: Ruleset, List_of_sent_NER, List_of_sent_AdverbSubType, List_of_sent_NounSubType, List_of_sent_DependencyTag
    Output: Wh_QType
    Set DependencyTagList ← List_of_sent_DependencyTags
    Set QuestionList ← Empty
    BestSimilarityScore ← empty
    Wh_QTypeWord ← empty
    BestScore ← empty
    if List_of_sent_NER is not empty then
        if List_of_sent_NER == "PERSON" then
            Wh_QTypeWord ← "Who"
        else if List_of_sent_NER == "LOC" then
            Wh_QTypeWord ← "What"
        else if List_of_sent_NER == "DATE" then
            Wh_QTypeWord ← "When"    ▷ ... add more conditions based on NER types
        end if
    else if List_of_sent_AdverbSubType is not empty then
        if List_of_sent_AdverbSubType == "PLACE" then
            Wh_QTypeWord ← "Where"
        else if List_of_sent_AdverbSubType == "TIME" then
            Wh_QTypeWord ← "When"
        else if List_of_sent_AdverbSubType == "MANNER" then
            Wh_QTypeWord ← "How"
        else if List_of_sent_AdverbSubType == "FREQUENCY" then
            Wh_QTypeWord ← "How Often"    ▷ ... add more conditions based on AdverbSubType
        else    ▷ Handle other cases
        end if
    else
        if List_of_sent_NounSubType == "PERSON" then
            Wh_QTypeWord ← "Who"
        else if List_of_sent_NounSubType == "ANIMAL" then
            Wh_QTypeWord ← "What"
        else if List_of_sent_NounSubType == "OBJECT" then
            Wh_QTypeWord ← "Which"    ▷ ... add more conditions based on NounSubType
        else    ▷ Handle other cases
        end if
    end if
    Return Wh_QTypeWord
    RulesetMapping(/* Arguments for RulesetMapping function */)
end function
```

Table 3.5: Adverb SubTypes and Corresponding Question Words

Adverb SubType	**Question Word**
Time	When
Place	Where
Manner	How
Degree	How
Frequency	How often

similarity to the human-generated questions. However, it is important to keep in mind that limitations exist with BLEU-N scores, and they may not necessarily reflect the quality of the questions in terms of their informativeness, relevance, and coherence.

Table 3.6: Sample sentence question pairs from our dataset.

No	Sentence	Question
1	Ethiopia defeated Italy at the Battle of Adwa	Who won the battle of Adwa?
2	GERD is the largest dam in Africa	Which is the largest dam in Africa?
3	Beads of water can be formed by clouds	What type of water formation is formed by clouds?
4	Limestone is formed by deposition	What kind of rock is formed by deposition?
5	Ethiopia is a country comprised of 13 months	Which country has 13 months in a year?
6	Bacteria are found in soil	Where are bacteria found?
7	A fish can breathe in the water	Which can breathe in the water?

On the other hand, the fact that ROUGE had the highest F1-Score suggests that the system-generated questions had a high level of overlap with the human-generated questions in terms of n-gram sequences, although it doesn't consider different words with the same meaning. It was shown through the experimental analysis that the combination of dependency parsing with NER is effective in identifying the subject, verb, object, and adverb parts of a sentence [99] which are essential for question generation. The effectiveness of identifying the subject, verb, object, and adverb parts of a sentence, which are essential for question generation[100] is revealed through experimental analysis of the combination of dependency parsing with NER. For instance, consider the following two sentences, which have the same meaning and can generate the same question. The subject, verb, object, and adverb parts of a sentence are extracted using dependency parsing in the system.

Table 3.7: BLEU–N and ROUGE-N metrics of Automatic evaluation result

Metrics	**Type**	**Score**
BLEU	1-gram	0.862654
	2-gram	0.785234
	3-gram	0.773411
	4-gram	0.751432
Rouge-1	F1 score	0.619192
	Precision	0.59619
	Recall	0.65
Rouge-2	F1 score	0.533333
	Precision	0.52
	Recall	0.55
Rouge-L	F1 score	0.619192
	Precision	0.59619
	Recall	0.65

3.3.3 Test results

AQG has seen significant advancements in recent years, with the development of various models that use deep learning techniques to generate questions from different types of textual data. For instance, the recent works of Zhao et al. [101] have proposed NN-based models that utilize contextual embedding and attention mechanisms for question generation. Furthermore, a crucial NLP objective is to extract significant sentences from a given text, and another objective is to generate extractions based on the original text. In this context, rule-based systems play a vital role in extracting pertinent words for generating uncomplicated and domain-specific questions. In this chapter, a rule-based AQG system was proposed that employs dependency parsing and considers various types of wh-question words. The system uses a combination of NER, POS, dependency tags, and adverb subtypes for rule-set mapping and question generation. While the proposed system has demonstrated good performance for simple sentence structures, future research could explore the integration of neural network-based models to improve the system's ability to generate complex questions. Moreover, future work could also focus on enhancing the system to include paragraph-based question generation. Overall, the proposed rule-based system provides a foundation for developing more sophisticated AQG systems that can generate questions from a wide range of textual data.

3.4 Summary

In conclusion, this chapter presented a rule-based AQG system that utilized dependency parsing and a comprehensive analysis of English sentence structure. The system was evaluated using both automatic and human evaluation techniques, and the results showed that the quality of the generated questions was highly dependent on the complexity of the sentence, with better quality and more natural questions generated for sentences with simple structures. Recent advances in AQG have led to the introduction of new models that utilize machine learning techniques, including neural networks, to generate questions from the text. These models can generate questions from both single sentences and paragraphs and have the potential to generate more complex and diverse questions. Furthermore, machine learning techniques, including neural networks, have been applied to question-generation models for various domains, including medical and scientific question generation. In conclusion, the field of AQG is rapidly evolving, and future work will likely focus on developing more advanced models that can generate more diverse and complex questions. The current rule-based system presented in this paper serves as a baseline for future research in the field. The new scientific findings of this chapter are summarized as follows:

Thesis 1. *A novel extended dependency parsing technique has been developed. For the proposed system, I have developed two algorithms that address the current limitations of sentence parsing, which depict the essential steps in our methodology: Algorithm 1: Ruleset Mapping for Question Generation, which selects the best matching rule. Algorithm 2: Question Word Selection for Question Generation to determine the appropriate question word (Wh QTypeWord) based on inputs, including NER, adverb subtype, noun subtype, and dependency tags. This extended dependency parsing method emerges as a promising avenue for enhancing the accuracy and effectiveness of sentence parsing in text-to-semantic applications. The test results show that the proposed algorithm provides questions with acceptable quality.* **[2][5][10][12][13]**

Chapter 4

Multilayer Perceptron-Based Sentence Parsing

4.1 MLP-Based Sentence Parsing Model

The utilization of neural networks, particularly MLPs, presents a paradigm shift from traditional template-based methods. This chapter explores the design, implementation, and evaluation of an MLP-based model tailored for AQG. Even though the AQG researchers began their work a decade ago, they still have serious shortcomings. The main goal of my investigation is to compare two very different question-generation techniques in a closed domain. The first version is a traditional template-based method that requires manual rule generation. Although the template-based solution is not an option for an open domain, I can generate an appropriate ruleset for a closed limited domain. The second approach is the neural network-based method using an MLP architecture.

The main contribution of this study is to develop an AQG model using phrase-based MLP and template-based approaches. In addition, the author tests and analyzes the efficiency of both approaches. The paper presents an efficiency comparison of two AQG methods, the template-based and the neural network-based methods on a closed domain. The performed test experiences show that both methods can provide a good question set, but the neural network-based method using NLP dominates the classic template-based approach not only in the open world domain but also in the closed word domain.

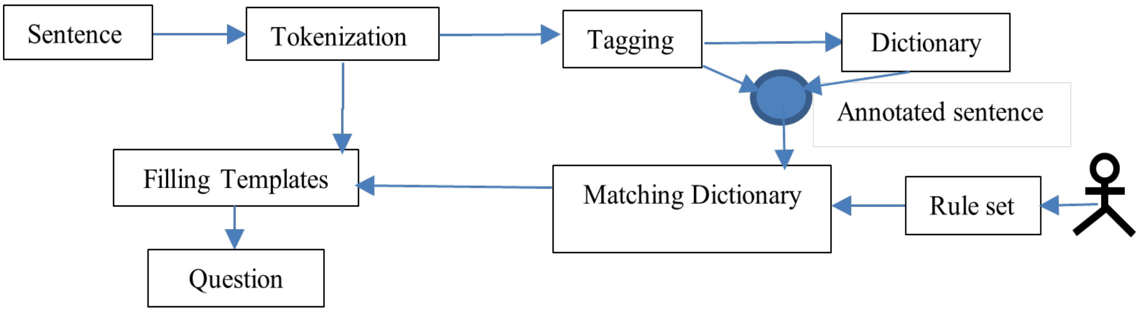


Figure 4.1: Block diagram of Template-based AQG

4.2 MLP Based Model for POS-Level AQG using Keyword Extraction

The notion behind template-based question generating is that a question template can capture a class of context-specific category questions. The block diagram of template-based AQG process from sentence to question generation has been illustrated in Figure 4.1. In the development of a template-based question generation system, the rule-set construction phase is the core module.

A template rule is a pair of sentence pattern and question patterns; R [ST, QT], where the sentence pattern and the question pattern are given with a list of tokens:

$$ST = [T_1^S, T_2^S, \ldots, T_m^S],$$
$$\text{where } T_i^S \in \text{Tokens}$$

This denotes the set of tokens that are references for the POS tag,¡– a position description value, and a symbol Words denote the set of valid words of the language.

$$QT = [T_1^Q, T_2^Q, \ldots, T_m^Q],$$
$$\text{where } T_i^Q \in \text{Words or } T_i^Q \in \text{ST}$$

In the question generation, the first step is to determine the token list T for the input sentence S.

$$T = [T_1, T_2, \ldots, T_m], \quad T_i \in \text{Tokens}$$

Then the best matches rule is selected from the rule set to generate the question sentence:

$$R_w = \arg\max_r \{\mathrm{sim}(T, \mathcal{ST}_r)\}$$

The symbol sim () means a similarity calculation of sentence patterns. In our experiment, I have used the formula for the similarity calculation of two strings.

$$\mathrm{sim} = \frac{1}{1 + \text{edit distance}(T, \mathrm{ST}_r)} \tag{4.1}$$

The idea behind the measure is that it would approximate a post-editor, in that it is based on edit distance, i.e. the minimum number of deletions, substitutions and insertions of words needed to turn the candidate translation into the reference translation [102].

The output question sentence is constructed with the application of R_w on S:

$$Q = \mathrm{apply}(R_w, S) \tag{4.2}$$

where the applied method converts QT_w into a sequence of words.

$$Q = \begin{cases} T_i^Q \in QT_w & \text{if } T_i^Q \in \text{Words} \\ \mathrm{sub}(T_i^Q) & \text{if } T_i^Q \notin \text{Words} \end{cases}$$

The symbol $\mathrm{sub}(T)$ denotes the function to determine the corresponding word for token T from S.

4.3 Query Generation using Neural Network with Embeddings

To tackle the challenges posed by dependency parsing in AQG, one approach involves leveraging neural networks with embeddings. Neural networks, equipped with carefully crafted embeddings, enable the generation of queries that capture detailed relationships between words. The first thing that happens when a piece of text is passed through a network is to transform the words into a representation that the network can manipulate [103]. The words can be transformed into real-valued vectors of a specific constant dimensional space to do this. The term ”word embedding” refers to the vector representation of words. These word embedding's can be randomly initialized while training a neural network (NN), and their weights can be adjusted along with the network's other trainable parameters [104].

The most common resources are trained based on the following techniques with a considerable amount of data. Word2vec [105] is one of the word embedding techniques that use a statistical method to extract word embedding from a given text corpus. To get context based on the words present in a particular window, it may be based on the Continuous Bag of Words (CBOW)[106]. The Skip-Gram Model [107], on the other hand, may also be used in this situation to forecast a word's surroundings. This approach has previously produced pre-trained word embedding that is based on the Google News corpus, which contains over 100 billion words. Vectors with 300 features or other values are used to represent these words.

In this approach, a neural network classifier is used to predict the query sentence. The proposed model uses an MLP neural network model, where the MLP network can predict a single value. To predict a word sequence, the system contains more neural networks, each model is responsible for a single word position. If M denotes the maximal length of the investigated sentences, the architecture involves M MLP modules and the output sentence is constructed as the sequence of the generated outputs.

Considering the MLP units, each neural network unit has the same input, namely the feature vector of the corresponding sentence. The content of the feature vector involves the following elements for every word in the input sentence: POS category, NER category, morphological units, and Word context description. For the description of the word context, I used the Word Embedding model. The embedding model is used mainly in NLP, where the model generates a high-dimensional vector for every word, including its context. In the generated vector space of the embeddings, words with similar meanings (like dog and cat) have positions near each other.

The formal model of the proposed algorithm is based on the following elements.

1. training set $T : \{t_i\}$ where t_i is equal to a triplet (s_i, p_i, q_i). The first component is the input sentence; the second component is the position of the target word and the third tag is equal to the generated query sentence.

2. The input and output sentences are transformed into a lemmatized form in order to increase the unambiguity of the sentence form.

$$s'_i = lemma(s_i), q'_i = lemma(q_i). \tag{4.3}$$

   In this preprocessing step, every variant of the same word is mapped to the same word format.

3. In the case of a query sentence, I performed a simplification phase to reduce the set of possible target words. This step is based on the consideration that most words in the query are related to some word in the input sentence. In the preprocessing step, every word related to the input sentence is converted

to a relative format $\_n$ where n denotes the related position in the input sentence.

4. N: the set of MLP neural networks, with $N_i : T \rightarrow W$, where W is the union of two sets: a) the set of relative words from the input sentence $(\_1, \_2, ...)$ b) the set of words not given in the input sentence.

5. The training set $T : \{t_i\}$ is converted into (X, Y), where X denotes the calculated feature set for $\{s_i\}$ and Y is the encoded $\{q_i\}$ set.

6. Split the (X, Y) into (X_i, Y_i) where i denotes a position index and $X_i = X$ for every position.

7. Train the generated N_i MLP networks.

In the case of prediction, the following steps are executed:

1. Input is a pair (s, p) where s is the base sentence and p is the position of the target word.

2. $s' = lemma(s)$

3. For every positions $l \in [0, ..., M]$, it performs a prediction: $c_i = N_i.pred(s'_i)$ the output categories c_i are converted into word lemmas w'_i using the s_i and the additional dictionary.

4. $q'''_i = [w'_0, w'_1, ..., w'_L]$, where $w'_{L+...}$ is the first terminal symbol in the sequence $(L < M)$.

5. Converting q''_i into the output q''_i using a NLP engine.

For the evaluation of the engine accuracy, I have used two measures:

- $acc_1 = Avg(\sum_{l=1}^{M} d_{nl})$, where d_{nl} denotes characteristic variable for the n-th sample at the l-th position. Its value is equal to 1 if the predicted word is equal to the real word at the given position, otherwise, the value is equal to 0.

- $acc_2 = Avg(1 - edit\_distance(q'_i, q''_i)/M)$. In this case, the edit distance value is used to measure the similarity between the predicted and measured values. The main motivation for acc_2 is the fact that acc_1 is too strict, it requires position-level equivalences.

In the test framework, I have used the medium language model of the spaCy Python framework. In this model, the output vector of the embedding module has 300 dimensions.

4.4 Research Method

Delving into the architecture of the MLP model designed for sentence parsing in AQG. Understanding the layers, nodes, and activation functions that contribute to the model's learning and inference processes. Comparison of Template-Based and MLP-Based. Evaluating the performance of the MLP-based approach against traditional template-based methods. Highlighting the advantages of employing neural networks in capturing complex linguistic patterns and semantic relationships. In this study, I have defined metrics to assess the performance of the MLP-based model in comparison to template-based approaches. Metrics include accuracy, precision, recall, and F1 score in the context of sentence parsing for AQG. I have developed our proposed system using Google Colaboratory [108], or "Colab" for short, which allows us to write and execute Python in our browser with no additional configuration. Then I divided the dataset into a training set (90 percent) and a testing set (10 percent) using random sampling techniques. The implementation of the template-based and MLP-based question generation is available on GitHub[1].

After I have implemented the proposed system, I need to measure and compare its efficiency. According to our observation, most scholars do not know which methodologies to use for the evaluation techniques of question generation. It is hard to quantify the generated question as "good" because good questions tend to be significant, syntactically correct, semantically sound, and natural. As a result, recent QG research tends to utilize human evaluation. However, human evaluation can be labor-intensive, time-consuming, inconsistent, and hard to reproduce. Due to these, researchers[109] still use automatic evaluation metrics, even though studies have shown that automatic evaluation metrics do not correlate well with fluency and coherence.

In our evaluation methodology, I have used human raters to blindly compare automatically-generated questions with human-generated (golden questions) rating (1-5) marks for all testing questions and BLEU and ROUGE automatic evaluation metrics. The BLEU is a metric to evaluate a generated sentence to a reference sentence. BLEU [77] was originally created to measure the quality of machine translation with respect to human translation. It computes an N-gram precision difference between the two sequences, as well as a penalty for machine sequences being shorter than human sequences. A perfect match receives a 1.0 score, whereas a perfect mismatch receives a 0.0 value. The most you can do is get a 0.6 or 0.7 on the scale. This score was created primarily to assess the accuracy of automatic machine translation systems' predictions[109]. On a set of references, BLEU calculates the average n-gram precision. A BLEU-n score is a BLEU score that has been calculated using up to n-grams.

The other metric employed is ROUGE, a set of evaluation metrics proposed in the context of automatic summarization[78]. ROUGE is a collection of metrics

[1] https://github.com/waleligntewabe/MLP-based-AQG

rather than a single metric. ROUGE-N is the one that is most likely to be used. The N in ROUGE-N stands for the n-gram that we're employing. I would measure the match rate of unigrams in ROUGE-1, and bigrams between our model output and reference in ROUGE-2. We'll calculate the ROUGE recall, precision, or F1 score once we've determined which N to utilize. The last metric is ROGUE-L [110], which is based on the length of the longest common subsequence (LCS) between our model output and the reference sequence. It calculates the final as the F-measure of these values above, using the fractional length of LCS over sequence length as precision/recall for one and vice versa for the other. To implement these metrics I have used the Python rouge library[2]. To further inspect the capabilities of our proposed QG models, I also perform human evaluation on our template-based and MLP based question generation models.

4.5 Comparison of Template-Based and MLP-Based Approaches

In this experiment, the rule set is constructed for the general domain by considering the most common English question patterns and the different structures of the sentences. Regarding the preprocessing phase, the first step is tokenization. Tokenization is the mechanism by which a given expression is split into words or other significant elements called tokens. Another operations steps in the preprocessing phase are sentence segmentation, tokenization, POS tagging, and rule matching. Rule set construction and template matching is based on the POS tag feature vector of the tokens. Rules holds both sentence template and their question template. To apply on concrete sentence, the POS tag feature is determined for matching. Demonstration that the neural network-based method using NLP outperforms the template-based approach in both open world and closed word domains.

Sentence template is given by list of POS tags with position index to differentiate the similar POS tags within the sentence. e.g. [NN1, VBZ1, VBN1, IN1, NN2]. Question template[35] is given by list of common question words and POS tags with position index to differentiate the similar POS tags with in the sentence e.g. [where, NNS1, VBP1, VBN1, IN1, DT1, NN1].

The next task is to evaluate each generated questions with the original sentences and return the best scorer question as a final result. Finally, the system generates a question with all possible constructed templates. Then the system automatically evaluates each generated question with the given sentence using the BLEU metric and takes the maximum score as the final output question.

Example 1

Let us assume the rule set contains the following three rules having different

[2] https://github.com/pltrdy/rouge

numbers of question templates.

Rule 1

$$\begin{aligned}\text{ST} &= ['NNS1','VBP1','VBN1','IN1','DT1','NN1'];\\ \text{QT} &= ['Where' +' VBP1' +' NNS1' +' BN1'+'?']\end{aligned}$$

Rule 2

$$\begin{aligned}\text{ST} &= ['VBG1','NN1','VBZ1','NNS1'];\\ \text{QT} &= ['Which' +' VBZ1' +' NNS1'+'?']\end{aligned}$$

Rule 3

$$\begin{aligned}\text{ST} &= ['NN1','VBZ1','VBN1','IN1','NN2'];\\ \text{QT1} &= ['How' + \text{NN1} +' VBZ1' +' VBN1'+'?'];\\ \text{QT2} &= [\text{NN1} +' VBZ1' +' VBN1' +' IN1' +' what'+'?'];\\ \text{QT3} &= ['Which' +' VBZ' +' VBN' +' IN' +' NN2'+'?']\end{aligned}$$

The input sentence is the following:

$$S = \text{Limestone is formed by deposition;}$$

In the first step, it generates the token list and yields the following list:

$$T = \text{`NN1','VBZ1','VBN1','IN1','NN2'.}$$

Based on the similarity calculation using edit distance, I have got the following similarity values for the rules:

$$\begin{aligned}\text{sim}(T, \text{ST}_{r1}) &= 0.14,\\ \text{sim}(T, \text{ST}_{r2}) &= 0.08,\\ \text{sim}(T, \text{ST}_{r3}) &= 1,\end{aligned}$$

Based on the best similarity score, the winner is Rule 3.

Using the substitutions, I have got the concrete variants for the question templates of the winner rule:

$$\begin{aligned}Q1 &= \text{How limestone is formed?; BLEU scores 0.55}\\ Q2 &= \text{Limestone is formed by what?; BLEU scores 0.668}\\ Q3 &= \text{Which is formed by deposition?; BLEU scores 0.56}\end{aligned}$$

Based on the BLEU score, the winner question is "Limestone is formed by what?"

MLP architecture

MLP is a supplement of a feed-forward neural network and consists of three types of layers: the input layer, output layer, and hidden layer[111]. The neural network architecture learns any function $f(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output.

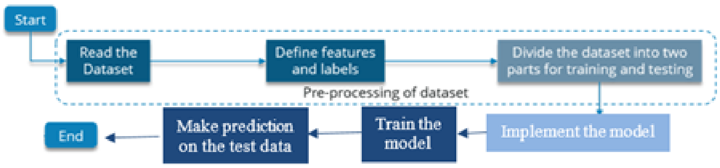


Figure 4.2: Training and prediction steps of MLP model

Figure 4.2 illustrates the training and prediction steps of the MLP model. First, I read and pre-processing the dataset then train the model on the implemented model next train and predict the test data. I have created datasets both manually and from the Question-Answer Sentence Composition (QASC)[3] dataset to prepare the MLP training model. The QASC dataset is a question-and-answer set that focuses on sentence composition. It includes a corpus of 17 million sentences and 9,980 multiple-choice questions regarding grade school science (8,134 train, 926 dev, 920 test). I have discovered several null values in the QASC dataset, as well as very long and useless sentences. Due to this, I have tried to preprocess and clean up the dataset and selected only the top 700 short and more meaningful sentence question pairs. In addition, I have constructed 300 sentence question pairs manually from general truth and common sentences, and finally, built 1000 sentence question pair datasets for our training. Table 4.1 shows sample sentence question pairs from our dataset.

In our experiment, the first preprocessing step is to convert the sentence into a sequence of phrases. Phrases are a combination of two or more words that can take the role of a noun, a verb, or a modifier in a sentence. In the English language, there are five phrase types i.e. noun phrase (NP), verb phrase (VB), adjective phrase (ADJP), adverb phrase (ADVP), and prepositional phrase (PP). I have used chunking to extract phrases from sentences. To construct the input matrix for the MLP model build a vocabulary with a combination of English phrases and unique words that exist only in questions. Then I extend the vocabulary with the WH question words and the most frequently unique words. In the test, I built up a vocabulary containing 40 unique words. The sentence and question vector representation for MLP training model has the following form:

[3] https://github.com/allenai/qasc

Table 4.1: Sample sentence question pairs from our dataset.

No	Sentence	Question
1	Ethiopia defeated Italy at the Battle of Adwa	Who won the battle of Adwa?
2	GERD is the largest dam in Africa	Which is the largest dam in Africa?
3	Beads of water can be formed by clouds	What type of water formation is formed by clouds?
4	Limestone is formed by deposition	What kind of rock is formed by deposition?
5	Ethiopia is a country comprised of 13 months	Which country has 13 months in a year?
6	Bacteria are found in soil	Where are bacteria found?
7	A fish can breathe in the water	Which can breathe in the water?

$$\left[\begin{array}{ll} sp1x0, sw1x1, sw1x2, \ldots, sp1x40 & qp1x0, qp1x1, qp1x2, \ldots, qp1x40 \\ sp2x0, sw2x1, sw2x2, \ldots, sp2x40 & qp2x0, qp2x1, qp2x2, \ldots, qp2x40 \\ sp3x0, sw3x1, sw3x2, \ldots, sp3x40 & qp3x0, qp3x1, qp3x2, \ldots, qp3x40 \\ \ldots & \ldots \\ sp10x0, sp10x1, sp10x2, \ldots, sp10x40 & qp10x0, qp10x1, qp10x2, \ldots, qp10x40 \end{array}\right]$$

Then I converted the phrase tags of each sentence into vector form based on their vocabulary. Finally, the matrix of the training set is created using a one-hot encoding method. Based on our observation from our dataset the maximum length of phrases in a sentence or question is nine, which means the vector length of each sentence and question would be 40*9=360. To read all the vector form datasets, I have used a loop to combine in one array and form a none*1*360 matrix. Then the model starts to train the vector-matrix values of a single sentence train with every vector equivalent of the question phrase tag. For our experiment, I defined the structure of a MLP network model. The model has 360 inputs, 3 hidden layers with 1000, 2000, and 1000 neurons, and an output layer with 360 output. Rectified linear activation functions are used in each hidden layer and a sigmoid activation function is used in the output layer.

A plotted graph and input and output shapes of each layer are illustrated in Figure 4.3. For regularization, I have tried dropout and there is no effect on the accuracy. A line plot of model classification accuracy for the training and validation datasets over training epochs is illustrated in Figure 4.4. The line plot demonstrates that the model learns the problem quickly, achieving an accuracy of about 80% in roughly 25 epochs rather than the 100 epochs. The line plot also illustrates that throughout training, train and test performance are comparable.

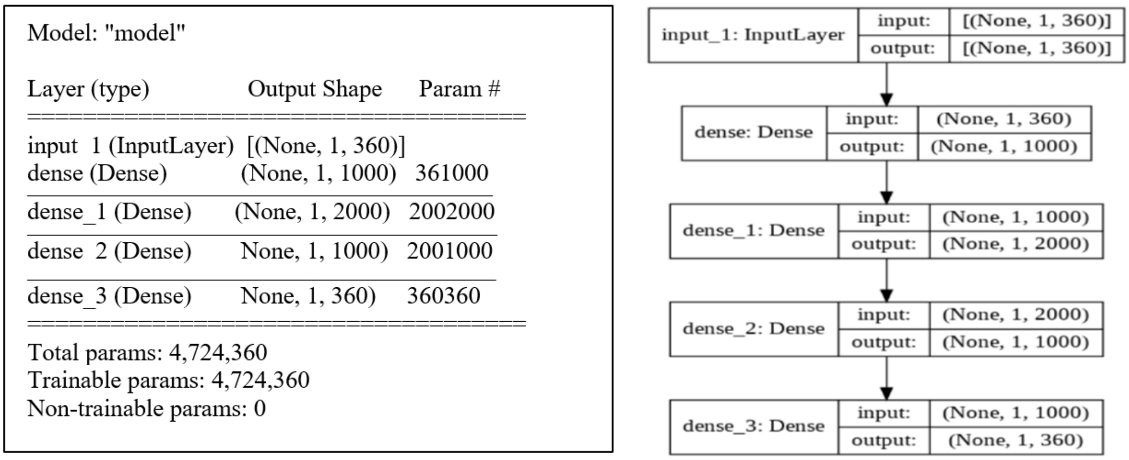


Figure 4.3: A model and plotted a graph of our MLP neural network

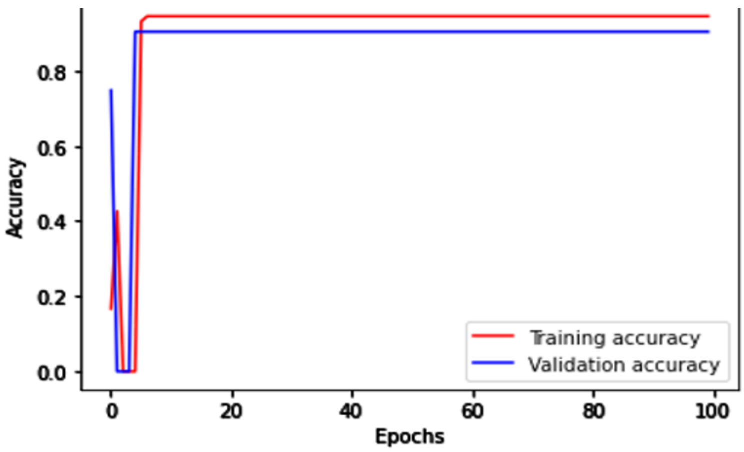


Figure 4.4: Accuracy on train and validation datasets

4.6 Results and Analysis

For human evaluation, I prepared 100 sentences randomly from different sources and I asked three annotators to score them on a scale of [1-5] independently, with the following three metrics: Fluency: whether a question is grammatical and fluent. Relevancy: whether the question is semantic relevant to the passage. Answerability: whether the question can be answered by the right answer.

Table 4.2: Sample test sentence question pair with generated questions.

No	Sentence	Human Generated Question	Template Based Generated Question	MLP based Generated Question
1	Ethiopia defeated Italy at the Battle of Adwa	Who won the battle of Adwa?	Who defeated Italy at battle Adwa?	What Ethiopia defeated the battle?
2	Hearing is the fastest human sense	Which is the fastest human sense?	What does Hearing do fastest?	What is the fastest human sense?
3	The fastest bird is the Peregrine falcon.	What is the fastest bird?	What does the bird is the?	What is peregrine falcon?
4	Dragonflies are one of the fastest insects	What are the fastest insects?	Who are the fastest of the insects?	What are the fastest insects?

Table 4.2 shows the sample sentences and questions generated by human, template-based, and MLP-based systems. The questions generated using the proposed system are evaluated using both automatic metrics and human evaluators regarding the gold questions. The survey was executed on Google Forms and evaluated by 10 fluent English speakers. Before starting the survey, the evaluators were informed about the purpose of the study and the questionnaire.

Table 4.3: Human-based evaluation result.

	Fluency	Answerability	Relevancy
Gold Question	4.4	4.387	4.262
Template Based	3.825	3.762	3.805
MLP based	3.85	3.837	3.875

However, human evaluators rate all questions generated by humans and the proposed system as shown in Table 4.3. The BLEU and ROUGE automatic metrics evaluation result is displayed in Table 4.4. A BLEU implementor's is to compare the candidate's n-grams to the reference's n-grams and count the number of matches. Position has no impact on these matches. The higher the number of matches, the better the candidate [109].

Table 4.4: Automatic metrics evaluation result.

		Template Based AQG Model	Phrase-Based MLP AQG Model
BLEU	1 gram	0.2778	0.331
	2 gram	0.556	0.496
	3 gram	0.8	0.451
	4 gram	0.8	0.582
Rouge-1	F1_score	0.327160714	0.350872356
	Precision	0.351731602	0.333549784
	Recall	0.307359307	0.381818182
Rouge-2	F1_score	0.056565655	0.163636362
	Precision	0.063636364	0.163636364
	Recall	0.051515152	0.163636364
Rouge-L	F1_score	0.296645774	0.350872356
	Precision	0.318398268	0.333549784
	Recall	0.279220779	0.381818182

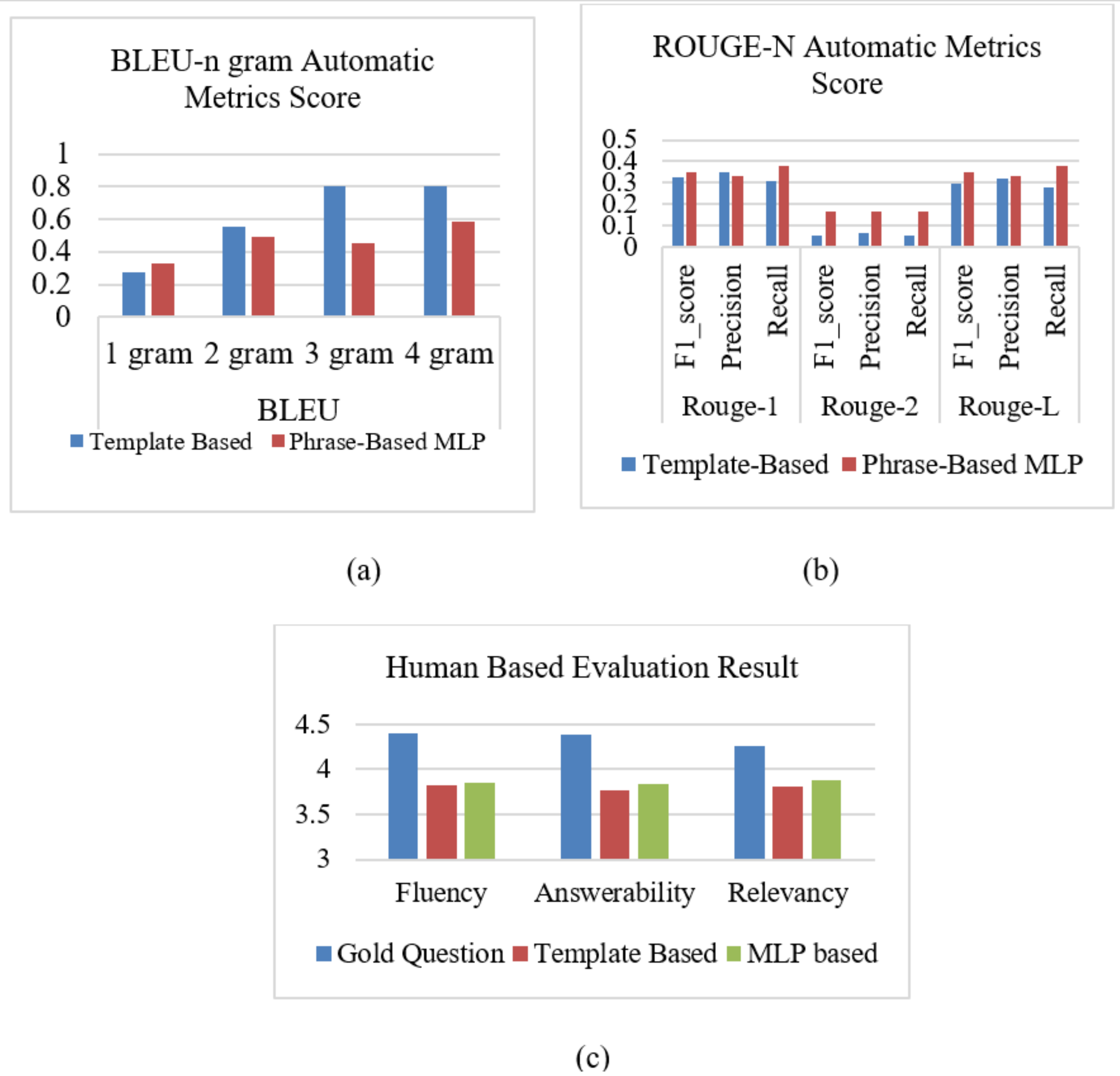
As shown in Table 4.4 and in Figure 4.5 (a) for better visualization, questions generated by the template-based system have got the highest score in BLEU 3 and 4-gram quality measure, but weaker results for the rest metrics. Figure 4.5 (b) shows that in all ROUGE-N automatic metrics phrase-based generated questions score better results than template-based generated. I have evolved human evaluators who consider the Fluency, Answerability, and Relevance of each sentence about the questions. In this evaluation, all generated question types i.e. gold question, template-based, MLP based need to be evaluated. The evaluation result presented in Table 4.4 shows that the MLP-based approach has better results than the template-based. As shown in Figure 4.5 (c) gold questions have scored the best result in human evaluation.

According to the overall evaluation result, the MLP-based approach got a better score. In addition, I have observed that humans can understand and answer the automatically generated questions. From this, I notice that our proposed MLP-based system is more encouraging than a template-based approach.

4.7 Summary

This chapter presents an efficiency comparison of two-generation methods on a closed domain. The first method is the template-based approach and the second is the neural network-based method. The quality of the generated questions was evaluated by both automatic metric scores (BLEU and ROUGE) and by human experts. The performed test experiences show that both methods can provide a good, 80%-88% accuracy in the test generation compared to human-generated questions. Based on the test results, the neural network based method using NLP dominates the classic template-based approach not only in the open world domain

Figure 4.5: Comparison of Template-Based and Phrase-based QG using (a) BLEU-n gram (b) ROUGE-N and (c) Automatic Evaluation



but also in the closed word domain. This chapter highlights the improvements and new ideas that were generated by using the Multilayer Perceptron-based approach. The new scientific results are summarized as follows:

Thesis 2.

A novel MLP-based Sentence Parsing Model has been developed and used for the improvement of parsing accuracy. The model can handle complex linguistic structures and it is in general more effective in generating questions than the rule-based approaches. The developed MLP-based approach emerges as a promising avenue for enhancing the capabilities of AQG. **[3][4][13][14][15]**

Chapter 5

Hybrid Parser for Semantic Graph Induction

This chapter presents a novel Hybrid Parser intended for Semantic Graph Induction, continuing the attempt to advance sentence parsing for text-to-semantic applications. Recognizing the difficulties in analyzing the complex relationships found in textual data, the Hybrid Parser combines conventional parsing approaches with state-of-the-art ChatGPT-based methodologies. This introduction provides the foundation for analyzing the Hybrid Parser's architecture, methods, and findings from experiments.

Semantic Graph Induction is a computational approach in NLP and AI that aims to extract and represent structured knowledge and semantic relationships from unstructured textual data. Semantic Graph visually represents the semantic structure of a document extracted from sentences[112]. Semantic graphs play a multifaceted role in various applications, spanning information retrieval, knowledge representation, question answering, text summarization, document clustering, and classification. Furthermore, I find diverse real-world applications across healthcare, finance, cybersecurity, and entertainment industries [113] as illustrated in Figure 5.1. In healthcare, semantic graphs are instrumental in integrating medical knowledge [114], providing a centralized view of IoT information, and enhancing virtual assistants and chatbots by delivering more contextually relevant responses to medical queries[115]. Furthermore, in the finance industry, semantic graphs have emerged as a crucial tool for managing financial knowledge securely[116], enabling applications like transaction surveillance, financial crime detection and prevention, and non-compliant user detection [117]. 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[118]. Moreover, semantic graphs play a vital role in cybersecurity by mapping historical cyber attacks and predicting potential future breaches, thus bolstering cyber defense strategies[119].

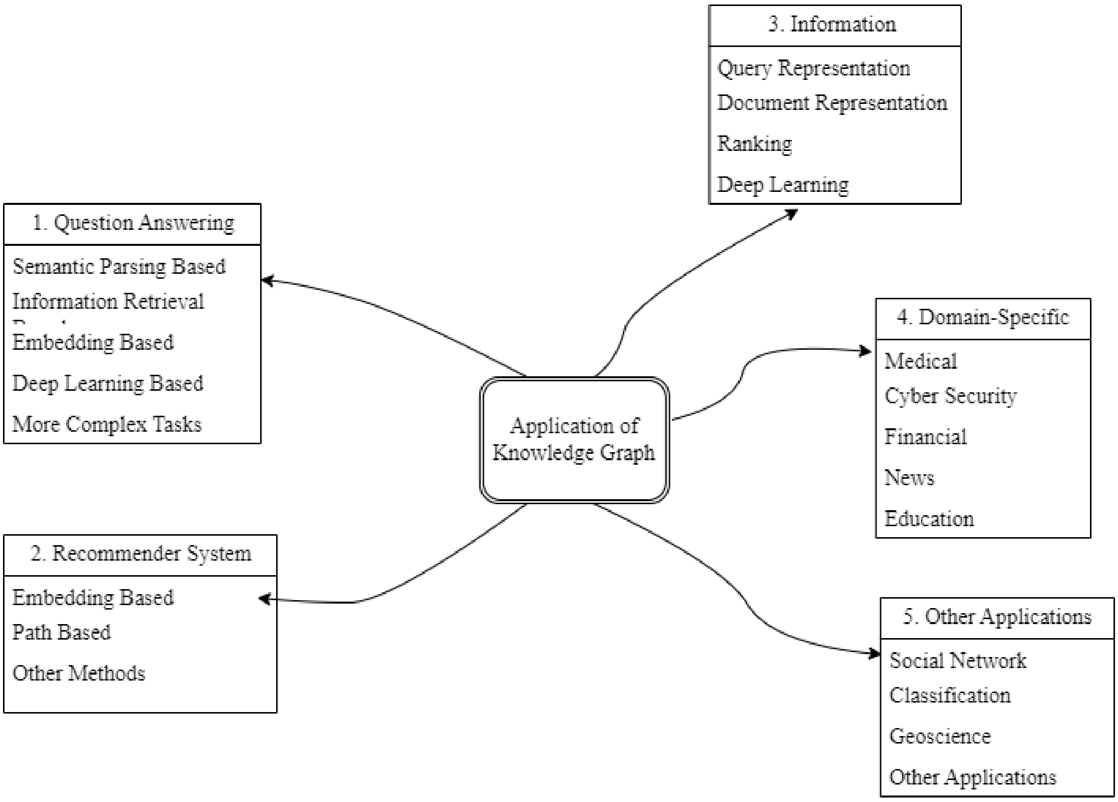


Figure 5.1: Real-world application fields of knowledge graphs [113]

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 an innovative 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 of events, specific locations, frequency, and the degree of attributes [120]. 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 Large Language Models (LLMs)[121], 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[122]. 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[123]. It has demonstrated remarkable skill in a wide array of language understanding tasks, including question-answering, language generation, and text summarization[124]. 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.

A pivotal factor of this work is the exploration of cutting-edge techniques for semantic graph construction, a process that explains the complex relationships and meanings embedded within textual data. The contributions of this work are as follows:

1. **Semantic Graph Construction Enhancement:** The primary objective of this study is to advance the field of semantic graph-based Construction, which serves as a fundamental resource in NLP and AI. The innovative approach presented in this study extends beyond traditional limitations to encompass more comprehensive structures. By adding elements like Subject, Predicate, Direct Object, Indirect Object, and Conjunction and exploring and predicting adverb types such as Time, Place, Manner, Degree, and Frequency, this work bridges the gap in capturing nuanced event descriptions and relationships within text. The work identifies the critical limitations of the current NLP parsing tools the inability to adequately represent detailed event descriptions involving aspects like time and place. This limitation serves as the primary motivation for the study, prompting the development of an innovative solution. By expanding the structure of semantic graphs and introducing adverb types, the study offers an effective solution to enhance the representation of complex textual data.

2. **Testing a novel ChatGPT-based parsing for functional sentence parsing:** It involves evaluating the model's performance in breaking down sentences into their functional components, such as subjects, predicates, objects, adverbs, and conjunctions. This assessment aims to determine the accuracy and effectiveness of ChatGPT in this specific parsing task. The testing process may involve using a dataset of sentences with known structures and comparing ChatGPT's parsing results against these ground truths to measure its parsing capabilities. Additionally, evaluating the model's ability to predict adverb types, including Time, Place, Manner, Degree, and Frequency, is essential in understanding its overall effectiveness in functional sentence parsing.

3. **Comparative Analysis of Methodologies:** A significant focus of this work is a comparative study of the two methodologies for semantic graph-based induction. The study meticulously dissects and evaluates ChatGPT-based methods and Hybrid Parser-based techniques. ChatGPT, a state-of-the-art language model, is examined for its potential to address the challenges of semantic graph-based induction, while hybrid sentence parsing techniques leverage rule-based and machine-learning components to enhance adaptability and robustness. This comparative analysis offers insights into the strengths, weaknesses, and real-world applications of these methodologies.

5.1 Basics of Semantic Graph

A semantic graph is a graph model where nodes represent concepts and edges (or arcs) represent relationships between those concepts[125]. 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, and a set $E \subseteq V \times V$ of directed edges (or arcs)[126]. An edge going from node $u \in V$ to node $v \in V$ is denoted as $(u, v) \in E$ and has a start (tail) vertex u and an end (head) vertex v.

Building semantic graphs is essential for many practical uses and ongoing research [114, 127, 128]. As I 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. 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 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[129].

Knowledge graphs have also got substantial attention in recent years, serving as vital tools for organizing and connecting vast amounts of information from diverse sources, including text corpora, databases, and the web[130]. Some well-known knowledge graphs, such as DBpedia, Freebase, and Wikidata, have been crucial in this effort. We're also using some smart techniques like word embeddings and word vector representations to make semantic graphs even better[131]. Resource Description Framework (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[132]. 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[133]. 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 Protocol and RDF Query Language (SPARQL) queries to create big semantic graphs that cover a lot of the web[134]. But there are challenges too. I need better ways to handle big sets of data, put together text and visual data, and make sure the knowledge graphs I have created are complete and correct. Researchers have used new techniques, like word embeddings and entity embeddings, to help to understand the fine details of how words and things are related[113].

In general, in the fields of RDF, ontologies, and the ideas behind the Semantic Web initiative, semantic graphs play an important role that understanding and managing information. The semantic graphs serve as a crucial foundation for knowledge representation and data integration, facilitating the consistent 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

5.1.1 Adverbs in Sentence Parsing

Adverbs play a crucial role in sentence parsing by providing valuable information about manner, time, place, frequency, and degree [135]. Understanding the role of adverbs in sentence structure and meaning is essential for accurate adverb-type categorization, which has implications for various natural language processing tasks. This section provides an overview of the significance of adverbs in sentence parsing and their impact on language understanding. Adverbs modify verbs, adjectives, and other adverbs in a sentence, influencing the overall semantics and conveying additional details[136]. They provide information about how an action is performed (manner), when an action occurs (time), how often an action happens (frequency), and the intensity or extent of an action (degree). For example, in the sentence "She sings beautifully," the adverb "beautifully" modifies the verb "sings" to indicate how she sings.

In sentence parsing, the process of analyzing the grammatical structure and assigning syntactic roles to words, adverbs contribute to the overall meaning and interpretation of a sentence[137]. They help disambiguate sentence structures and provide contextual clues for understanding the relationships between different elements. Adverb type categorization is essential for language understanding tasks such as sentiment analysis, question generation, and information extraction[138]. By categorizing adverbs into specific types based on their semantic properties, systems can better understand sentence meaning and generate more accurate and contextually appropriate responses.

Recent research in adverb type categorization has focused on the use of ML techniques and deep learning models to automate the categorization process[139]. Approaches like neural networks, support vector machines, and decision trees have been employed to classify adverbs into predefined categories, leveraging features such as part-of-speech tags, syntactic dependencies, and contextual information. The accurate categorization of adverb types contributes to improved language understanding and enables more sophisticated natural language processing applications[140]. By capturing the nuances of different adverb types, systems can better interpret and generate sentences, leading to enhanced performance in various tasks.

5.1.2 Impact on Language Understanding

Understanding the role of adverbs in sentence parsing is vital for language understanding. Adverbs provide crucial contextual information that helps determine the meaning and intent of a sentence. By analyzing adverbs, systems can infer temporal relationships, identify important events, and comprehend the nuances of language.

For example, in machine translation systems, adverbs influence the translation process by indicating the time, frequency, or manner of an action[141]. Similarly, in sentiment analysis, adverbs play a significant role in determining the sentiment expressed in a sentence. By categorizing adverbs accurately, systems can better grasp the intended sentiment and provide more precise sentiment analysis results. In information retrieval systems, adverbs help refine search queries and improve the accuracy of search results. Understanding the intended meaning behind adverbs allows systems to retrieve documents or information that align with the user's specific requirements[142].

5.1.3 Semantic Graph Induction

The term graphs refer to a common data format as well as a universal language for describing complicated systems. A common data structure and language for characterizing complex systems is called a graph. In its most basic form, a graph is just a set of objects or nodes, and the interactions (or edges) that exist between pairs of these nodes. For instance, we can utilize edges to signify the friendship between two individuals and utilize nodes to symbolize each person, effectively encoding a social network. This is illustrated in Figure 5.2, the famous Zachary Karate Club Network represents the friendship relationships between members of the karate club studied by Wayne W. Zachary from 1970 to 1972[143].

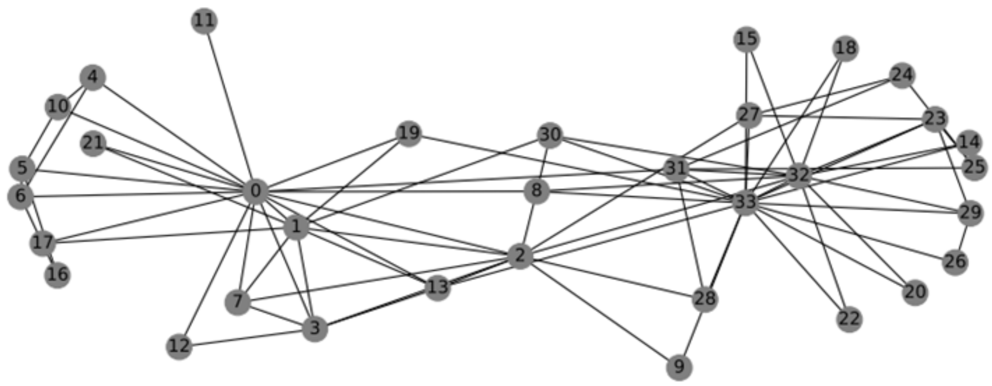


Figure 5.2: Application fields of knowledge graphs Example[126]

An edge that connects two individuals if they socialize outside of the club. During Zachary's study, the club split into two factions centered around nodes 0 and 33 and Zachary was able to correctly predict which nodes would fall into each faction based on the graph structure [126]. Graphs do more than just provide an elegant theoretical framework, however. They offer a mathematical foundation that we can build upon to analyze, understand, and learn from real-world complex systems[113, 126].

Semantic Graph Induction is the process of automatically building a semantic graph from unstructured data, like textual documents or datasets, is known as semantic graph induction[144, 145]. It involves taking information from unstructured data, such as entities, concepts, and their relationships, and putting it in an organized manner. This procedure frequently depends on NLP and machine learning approaches to discover and link entities, infer relationships, and build the graph. Constructing large-scale semantic graphs from vast and diverse datasets is a significant challenge. Researchers are continually developing more efficient algorithms and technologies to handle big data[146]. Recently, word embeddings and entity embeddings have become effective in capturing semantic relationships, and the advancements in embedding techniques continue to improve graph construction[147]. Ensuring the completeness and accuracy of knowledge graphs is an ongoing challenge[148, 149], with methods for knowledge base completion and alignment being actively explored[150].

5.2 Hybrid Parser-Based Method

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[151], 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. Figure 5.3 provides an overview of the structural framework of the Hybrid Parser-based approach. It illustrates the key components, including text preprocessing, rule-based and machine learning-based parsing, adverb-type prediction, and semantic graph construction, highlighting their interconnections.

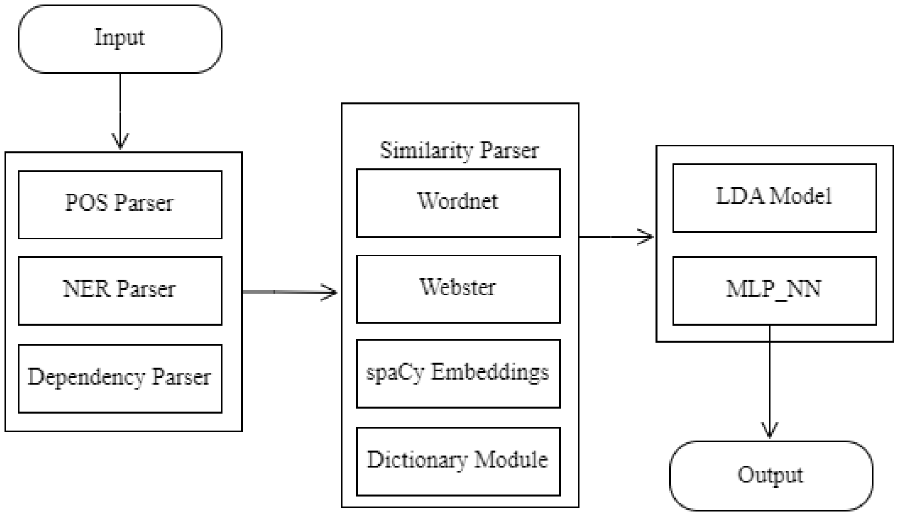


Figure 5.3: The structural framework of the proposed Hybrid parser-based

5.2.1 Adverb Type Categorization

Adverb-type categorization is a crucial task in NLP, playing a significant role in various language understanding tasks. This paper presents an evaluation of the efficiency of ChatGPT 3.5, a language model, for adverb-type categorization. By analyzing the performance of ChatGPT 3.5 and comparing it with a dictionary and ML-based method, the effectiveness and potential of ChatGPT 3.5 for adverb type categorization can be assessed[152]. Adverbs provide valuable information about manner, time, frequency, and degree in English sentence parsing, making their accurate categorization essential for understanding sentence structure and meaning.

Traditional rule-based approaches and curated dictionaries have been widely used for adverb-type categorization. However, recent advancements in language models, such as ChatGPT 3.5, offer new possibilities for improving the accuracy and efficiency of this task. ChatGPT 3.5, a state-of-the-art language model, has shown remarkable performance in various natural language understanding tasks [153]. Its ability to generate contextually relevant and coherent responses makes it a promising candidate for adverb-type categorization. By leveraging its language generation capabilities, ChatGPT 3.5 can potentially capture the contextual information and linguistic patterns necessary for accurate adverb-type predictions.

To evaluate the efficiency of ChatGPT 3.5 for adverb type categorization, a comparative analysis with a dictionary and ML-based method is conducted[154]. This approach combines curated dictionaries of adverb types with ML techniques to classify adverbs into predefined categories. By contrasting the performance of ChatGPT 3.5 with this established method, insights into the strengths and weaknesses of ChatGPT 3.5 can be gained. The evaluation aims to assess the effectiveness of ChatGPT 3.5 in accurately categorizing adverb types and to explore its potential applications in sentence parsing and other language understanding tasks. The findings from this evaluation contribute to a deeper understanding of the capabilities and limitations of ChatGPT 3.5 for adverb-type categorization.

5.2.2 Methodology

I utilized a free cloud-based platform called Google Collaboratory for running and writing Python code. For text analysis and parsing, I have used essential parsing tools such as spaCy and NLTK. To improve the analysis and understanding of language, I have 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 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, I have considered 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, $s = w_1, w_2, ..., w_l$. where symbol w denotes a word inside the sentence. The set of accepted functional unit types is given by

T = *Predicate, Subject, Direct Object, Indirect Object, Time, Place, Manner, Frequency, Degree, Conjunction*

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 the two input synsets:

$$l_l(s_1, s_2) = \frac{2 \cdot IC(s_c)}{IC(s_1) + IC(s_2)} \tag{5.1}$$

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 that contains some selected words with the related unit type labels:

$$l_l(s_1, s_2) = \frac{2 \cdot IC(s_c)}{IC(s_1) + IC(s_2)} \tag{5.2}$$

I have divide this dictionary into two parts:

$$D = D_B \cup D_L$$

where D_B is the set of baseline words, I have used 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 outputs the predicted unit label.

$$cat = NN(l(w))$$

For the training of the MLP unit, the D_L dataset is used as the training and test dataset. The MLP neural network unit under consideration comprises five layers, with one dedicated to model regularization (as depicted in Figure 5.4). The trained MLP unit demonstrated a commendable average accuracy of 92% on the tested datasets.

Figure 5.5 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.

```
=================================================================
dense_4 (Dense)              (None, 1100)              49500
_________________________________________________________________
dropout_1 (Dropout)          (None, 1100)              0
_________________________________________________________________
dense_5 (Dense)              (None, 440)               484440
_________________________________________________________________
dense_6 (Dense)              (None, 132)               58212
_________________________________________________________________
dense_7 (Dense)              (None, 6)                 798
=================================================================
Total params: 592,950
Trainable params: 592,950
Non-trainable params: 0
```

Figure 5.4: MLP architecture

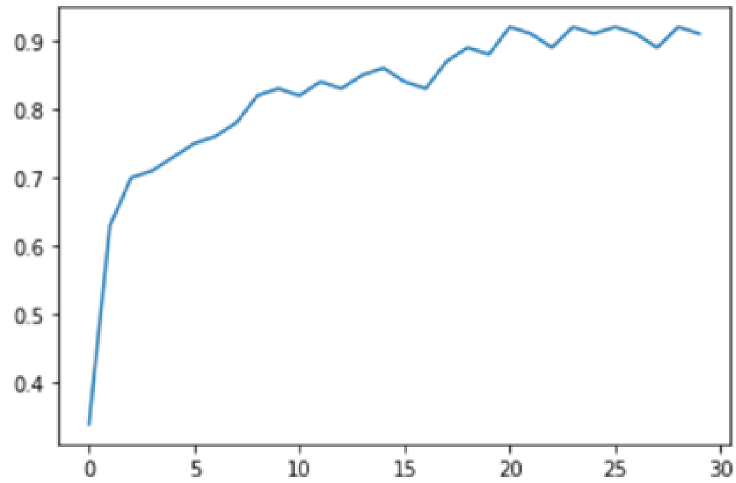


Figure 5.5: Validation accuracy curve in the training process

5.2.3 Dictionary and ML-based Method

The dictionary and ML-based method for adverb type categorization leverages a curated dictionary of adverb types and employs ML techniques to classify adverbs into predefined categories. One component of our prediction engine is the LDA. The LDA is a generative probabilistic model that can be used to generate random mixtures over latent topics for the description of the documents, where each topic is characterized by a distribution over words [155]. In LDA, I assume that there are k underlying latent topics according to which documents are generated and that each topic is represented as a multinomial distribution over the Ω words in the vocabulary. A document is generated by sampling a mixture of these topics and then sampling words from that mixture. More precisely, a document D of N words $D = w_1, ..., w_N$ is generated by the following process.

1. Select the value of N using a Poisson distribution
2. Choose a Dirichlet random variable θ

3. For each of the words in the document D:
   - Choose a topic z_i from Multinomial(θ)
   - Choose a word w_i from $p(w_i|z_i, \beta)$, a multinomial probability conditioned on the topic z_i.

In this model, the probability of a document is calculated with the following mixture:

$$p(D|\alpha, \beta) = \int p(\theta, \alpha) \left(\prod_{i=1}^{N} \sum_{z_i} p(z_i|\theta) p(w_i|z_i, \beta) \right) d\theta \tag{5.3}$$

The first step in implementing this method is to compile a comprehensive dictionary of adverb types. The dictionary includes a list of adverb types, along with their definitions and examples. This curated dictionary serves as a reference for categorizing adverbs into specific types. Next, ML techniques are utilized to train a classification model for adverb-type categorization. Features such as part-of-speech tags, syntactic dependencies, and contextual information are extracted from the adverbs in the training dataset. These features capture the linguistic properties and contextual patterns associated with different adverb types.

Using the extracted features, a decision tree classification model is trained on the labeled dataset[156]. The model learns to classify adverbs into predefined categories based on the extracted features and the corresponding adverb types from the curated dictionary. During the inference phase, the trained model takes an input adverb and predicts its type based on the learned patterns. By utilizing the curated dictionary and the trained classification model, the method assigns the most appropriate adverb type to the given adverb.

5.3 ChatGPT-based Sentence Parsing

5.3.1 Basics of ChatGPT-based Models

The ChatGPT-based method leverages the transformer architecture, specifically the GPT (Generative Pre-trained Transformer) model, for sentence parsing tasks. The GPT model is a variant of the transformer architecture that has been pre-trained on a large corpus of text data, enabling it to generate coherent and contextually relevant responses to input prompts[157].

Model Architecture: The ChatGPT-based method utilizes a multi-layer transformer architecture comprising encoder and decoder layers. The encoder processes the input text by attending to the surrounding context and generating a contextualized representation for each token. The decoder then uses this contextual information to generate the output sequence, which in the case of sentence parsing, corresponds to the parsed sentence structure.

Training Process:

The ChatGPT-based method is trained using a variant of the unsupervised pre-training and fine-tuning paradigm. During pre-training, the model is exposed to a diverse range of text data and learns to predict the next token in a sequence given the preceding context. This pre-training phase helps the model capture general linguistic patterns and semantic relationships present in the data.

Following pre-training, the model is fine-tuned on task-specific data for sentence parsing. This fine-tuning process involves updating the parameters of the pre-trained model using supervised learning techniques, where the model learns to map input sentences to their corresponding parsed structures. Fine-tuning allows the model to adapt its representations to the specific characteristics of the parsing task, thereby improving performance on downstream evaluation metrics.

Hyperparameters:

Several hyperparameters influence the performance and behavior of the ChatGPT-based method during training and inference. These include:

- Learning Rate: The rate at which the model's parameters are updated during optimization.
- Batch Size: The number of training examples processed in a single forward and backward pass.
- Number of Layers: The depth of the transformer architecture, i.e., the number of encoder and decoder layers.
- Hidden Dimension: The dimensionality of the model's hidden states, which determines the expressive capacity of the model.
- Dropout Rate: The probability of dropping out units during training to prevent overfitting.

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. 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[158]. The pre-trained ChatGPT model is fine-tuned on a labeled dataset of adverbs to improve its categorization accuracy.

The ChatGPT-based approach utilizes the language generation capabilities of ChatGPT 3.5 to predict the type of adverbs based on contextual information[159]. The pre-trained ChatGPT model is fine-tuned on a labeled dataset of adverbs to improve its categorization accuracy. The ChatGPT-based approach mentioned here leverages the language generation abilities of ChatGPT 3.5 for predicting the type of adverbs using contextual information. Initially, a pre-trained ChatGPT 3.5 model is used, which is a language model capable of generating human-like text based on the given input[160].

Suppose take the following sentence: "She ran quickly to catch the bus."

- Input to ChatGPT 3.5: The input sentence "She ran quickly to catch the bus" is passed to the ChatGPT model.
- Contextual understanding: ChatGPT 3.5 analyzes the contextual information in the sentence to generate a prediction for the type of adverb used.
- Prediction generation: Based on its training on a labeled dataset of adverbs, the fine-tuned ChatGPT 3.5 model predicts the type of adverb in the sentence. In this case, it identifies the adverb "quickly."
- Output: The ChatGPT 3.5 model generates the predicted adverb type, which is "manner" in this example. The "manner" category refers to adverbs that describe how an action is performed.

The ChatGPT-based approach for adverb-type categorization has its limitations[161, 162]:

- Data dependency: The accuracy of the ChatGPT 3.5 model heavily relies on the quality and representativeness of the labeled dataset used for fine-tuning. If the dataset is limited or biased, the model's predictions may be less reliable.
- Overreliance on context: While the contextual understanding is a strength of ChatGPT, it can also be a limitation. In some cases, the model's prediction may be influenced by the surrounding context, leading to incorrect adverb type categorization.
- Lack of explicit rules: The ChatGPT 3.5 model learns patterns and associations from the training data but may not explicitly understand grammatical or linguistic rules. This can result in occasional incorrect predictions for adverb types.

5.3.2 Methodology

The methodology for this study involves the following steps.

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.

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.

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 in Figure 5.6 the researcher developed and used representative prompt templates from the collected dataset in this regard. This fine-tuning process helps the model to learn and recognize the functional structure of a sentence including the adverb types based on contextual information.

Response Generation: With the ability to predict the functional structure of the sentence, ChatGPT can generate coherent and contextually relevant responses. These responses are informed by the adverb-type predictions, making them more precise and contextually appropriate. Throughout the methodology, emphasis is placed on the quality and representativeness of the labeled dataset, as this significantly influences the accuracy of adverb categorization and response generation. This ChatGPT-based methodology combines the power of pre-trained language models with fine-tuning on a domain-specific dataset to enhance adverb type prediction and response generation. It is a dynamic approach that leverages ChatGPT's natural language understanding and generation capabilities, making it a valuable tool for various NLP applications.

A significant aspect of language models is the 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[163]. 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[164].

```
# Prompt for generating a semantic graph description
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 2: {'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,
Clause parts of the following sentence:- Sentence: 'The child reads
the book carefully and attentively  at the library everyday. """

prompt = prompt_template + custom_prompt
```

Figure 5.6: Sample prompt template

Functioning as autoregressive language models, LLMs operate by taking an input text and iteratively predicting subsequent tokens or words[165]. Until the year 2020, the primary approach to adapt these models for specific tasks was fine-tuning. However, with the emergence of larger models like GPT-3, they can now be engineered with prompts to achieve similar outcomes[166]. LLMs are believed to acquire an inherent understanding of syntax, semantics, and the "ontology" within human language corpora[167]. 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, I utilized the ChatGPT 3.5 OpenAI API for the sentence parsing.

5.3.3 Prompt Engineering Techniques

Prompt engineering is a crucial technique employed to guide the behavior of large-scale language models like ChatGPT[157]. By strategically constructing input prompts, researchers and developers aim to obtain more accurate and relevant responses from these models[168]. 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.

5.3.4 Functional English Sentence Structure Analysis for AQG

In functional English sentence structure, sentences are constructed around the main clause, which contains a subject and a predicate. The subject is typically a noun or pronoun that performs the action described by the verb in the predicate. The basic structure of an English sentence is typically subject-verb-object (SVO), with any additional information placed before or after the main clause.

- Subject: I
- Verb: love
- Object: Ethiopia.

The SVO structure can be expanded to include other elements such as adjectives, adverbs, prepositional phrases, and clauses. Functional sentence structure also plays a key role in AQG, as it provides a framework for generating meaningful and grammatically correct questions. AQG is the process of automatically generating questions from a given text, and understanding sentence structure is crucial for this task [84]. One approach to AQG is to use dependency parsing, a technique for analyzing the grammatical structure of sentences, to identify the relevant information needed for the question. This approach can be combined with the use of functional sentence structure, which can help identify the subject, verb, and object in a sentence and use this information to generate meaningful questions. Several studies have explored the use of functional sentence structure in AQG. For example, Hanif et al.[169] proposed a method for generating questions from sentences by analyzing the syntactic structure and identifying the relevant information needed for the question. Kuo et al.[170] suggested that pre-trained language models can capture functional sentence structure patterns and improve AQG accuracy.

According to a recent study by Hanif et al.[169], the use of functional English sentence structure can help improve the performance of AQG systems. The study proposes a method for generating questions from sentences by analyzing the syntactic structure and identifying the relevant information needed for the question. The authors state that "using functional English sentence structure, we can identify the subject, verb, and object in a sentence and use this information to generate meaningful questions." They also note that the use of dependency parsing, a technique for analyzing the grammatical structure of sentences, can further improve the accuracy of AQG systems.

A new model was created by the researchers to detect the functional structure of sentences using the spaCy dependency parsing model. To develop this model, they analyzed functional English sentence structures in depth and wrote lengthy Python code to extract the subject, verb, object, and adverb components of the sentences. To compare the performance of their model, the researchers also developed another model based on ChatGPT using the openai API and created a suitable prompt template. The following code was used for this purpose:

```
import openai
openai.api_key = "Private API key"

def categorize_adverb(adverb):
    response = openai.Completion.create(
      engine="text-davinci-002",
      prompt=f"Identify the sentence parts'{doc}' like Subject,
        Verb, Object, Adverb",
      temperature=0,
      max_tokens=60,
      top_p=1,
      frequency_penalty=0,
      presence_penalty=0
    )
    category = response.choices[0].text.strip()
    return category
```

5.4 Comparative Evaluation and Result Analysis

The evaluation and test results are presented to compare the performance of ChatGPT 3.5 with the dictionary and ML-based methods for adverb-type categorization. The evaluation is conducted using a labeled dataset of adverbs, where the ground truth adverb types are known. The performance of ChatGPT-based dictionary- and ML-based methods is evaluated on this dataset, and the results are compared.

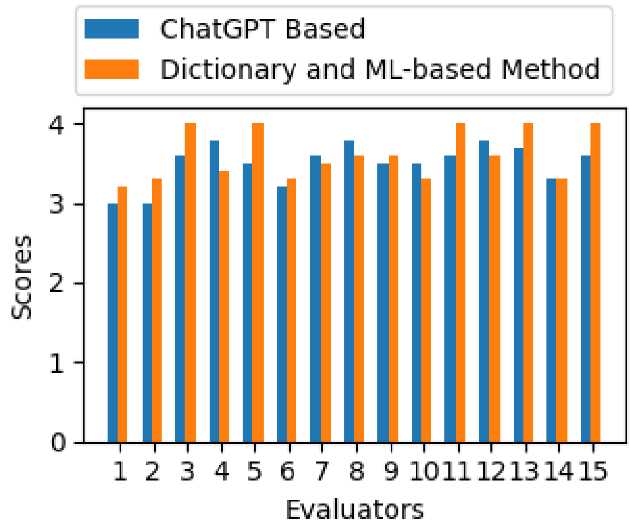


Figure 5.7: ChatGPT-Based and Dictionary ML-based Model Evaluation Result

The analysis of the evaluation and test results helps determine the strengths and weaknesses of each approach. It provides insights into the performance of ChatGPT 3.5 and the dictionary and ML-based method, highlighting their efficiency and effectiveness in adverb-type categorization. Figure 1 illustrates the overall evaluation results of linguistic experts for both methods, with scores of a minimum of 1.5 and a maximum of 4. The average test score for the ChatGPT-based method is 3.5 out of 5, indicating the overall evaluation by the experts. Similarly, the dictionary and ML-based method achieved an average test score of 3.6 out of 5, reflecting the collective assessment by the evaluators.

The evaluation results reveal that, from the tested data, in the case of the chatGPT-based model, 32 out of 40 adverbs are correctly classified based on the given scores, while the dictionary and ML-based model correctly classify 34 adverbs. Accuracy measures the overall correctness of the adverb-type categorization. It calculates the ratio of correctly classified adverbs to the total number of adverbs in the evaluation dataset.

Accuracy = (Number of correctly classified adverbs) / (Total number of adverbs).

For the chatGPT-based model: Accuracy= 0.8

For the dictionary and ML-based model: Accuracy= 0.85

These accuracy values suggest that the dictionary and ML-based method outperform the chatGPT-based model in adverb categorization.

5.4.1 Evaluation Method

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, the training dataset was constrained to 160 sentences, while the testing dataset comprised 40 sentences. Consequently, the dataset was partitioned, allocating 80% for training purposes and 20% for testing. The datasets are collected from various sources like academic history, biology, and world facts.

For the dictionary and ML-based method, the curated dictionary of adverb types is prepared, along with the extracted features from the training dataset. The ML model is trained on the dataset using appropriate algorithms and techniques. To evaluate the efficiency of the approaches, evaluation metrics for accuracy are employed. This metric quantifies the performance of the model in correctly categorizing adverbs into their respective types. The researchers didn't get any automatic evaluation methods; for this reason, only human-level evaluation methods were applied. This human-level evaluation methodology is applied to both the ChatGPT-based approach and the dictionary and ML-based method using the labeled adverb dataset. The results are analyzed to assess the efficiency of each approach in adverb-type categorization.

5.4.2 Results and Analysis

The model initially categorizes words and phrases in a given sentence into different functional structures, such as subject, verb, object, time adverb, place adverb, frequency adverb, and manner adverb. Due to the expensive nature of human-level evaluation, the training dataset was limited to 200 sentences, while the testing dataset consisted of 40 sentences. In this scenario, the dataset is divided into 80% for training and 20% for testing purposes.

Fifteen linguistic teachers from Higher Education institutions evaluate the overall categorization, and they rank the adverb-type categorization results on a scale of 1 (poor) to 5 (excellent). Additionally, they assess the classified adverb type as correct or incorrect. Among the evaluated adverbs, the maximum score is assigned to each adverb.

Examples of classifications that were evaluated as partially correct (grades 3 or 4):-

Sentence: The child reads the book carefully and attentively at the library everyday.

- ChatGPT 3.5 Based Method Generated result:- Predicate: read, Subject:

child, Object: the book, Time: everyday, Place: at library, Manner: carefully, and attentively.

- Dictionary and ML-based Method Generated result:- Subject: The child, Predicate: reads, Object: the book, Time: everyday, Place: at the library, Manner: carefully and attentively, Frequency: everyday.

Several evaluation metrics are employed to assess the performance of ChatGPT 3.5 and the dictionary and ML-based method for adverb type categorization. The accuracy and other evaluation metrics are analyzed to assess the efficiency and effectiveness of tested methods in accurately categorizing adverbs into their respective types. These evaluation metrics provide a comprehensive understanding of the performance of ChatGPT 3.5 and the dictionary and ML-based method in adverb-type categorization.

The performance of the two experimental models in functional English sentence structure analysis is evaluated using a 1-5 scale, with 5 being the best and 1 being the worst. The evaluation was carried out by 15 linguistic experts who assessed the correct extraction of functional sentence structure by the models. The evaluation metrics helped to determine the effectiveness of the models and their suitability for use in the educational domain. The evaluation sentences were selected based on their functional structure, and a total of 40 sentences were carefully chosen from various sources to ensure representativeness. Examples of these sentences are provided below.

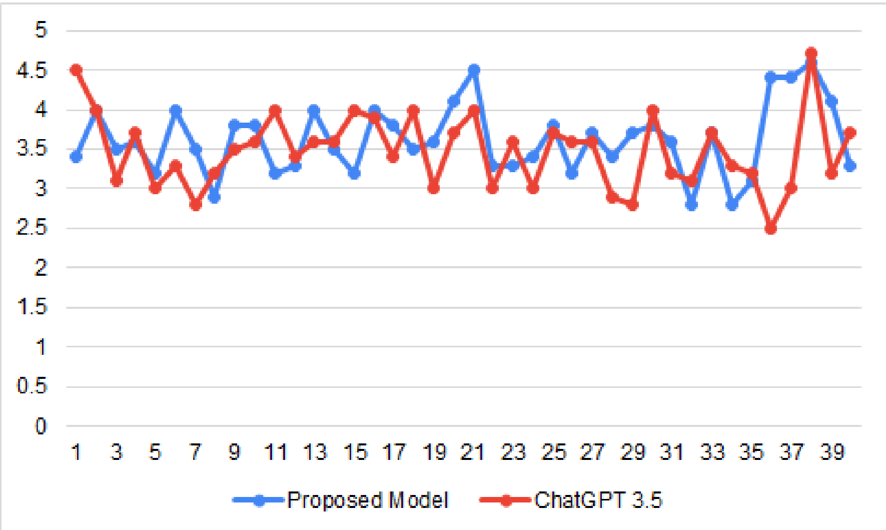


Figure 5.8: Linguistics expert evaluation result of the two models

Example one: Ethiopia is the source of the longest river in the world which is Nile.

A Proposed model:

. Subject=Ethiopia,

. Verb= is,

. Object=the source of the longest river in the world which is Nile,

. Adverb=longest

B ChatGPT model:

. Subject= Ethiopia,

. Verb= is,

. Object= the source of the longest river in the world,

. Adverb= which is Nile.

Example two: Tedy Afro is the famous Ethiopian artist.

A Proposed model:

. Subject=Tedy Afro,

. Verb= is,

. Object=the famous Ethiopian artist,

. Adverb=famous

B ChatGPT model:

. Subject= Tedy Afro,

. Verb= is,

. Object= the famous Ethiopian artist,

. Adverb= like

In Figure 5.8, we can observe the linguistics expert evaluation results of the two models based on the 40 prepared sentences. The average evaluation results of both models are summarized in Figure 5.9. The proposed model obtains a score of 3.62 out of 5, which is superior to that of ChatGPT 3.5. However, this result indicates that further improvements in the proposed model are necessary by considering various parameters. It is important to note that large language models such as ChatGPT 3.5 may not perform well in certain applications, such as functional sentence structure analysis.

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 engaged the expertise of linguistic professionals, educators, and students. The survey encompassed

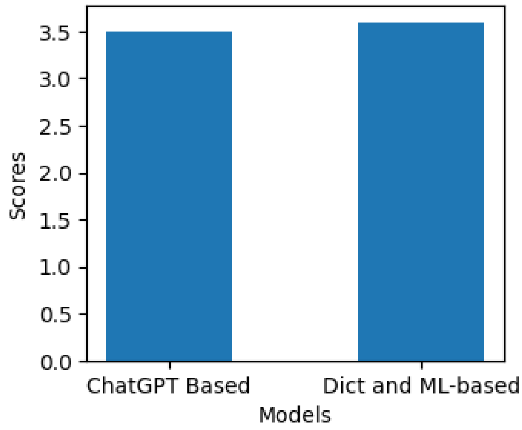


Figure 5.9: Average Linguistics expert evaluation result of the two models

five distinct rating categories: "Poor," "Below Average," "Average," "Above Average," and "Excellent.". I used similar test datasets for both approaches and make a comparative result analysis. In the evaluation process, used the ChatGPT efficiency for prompts of different lengths and complexity. Table 1 presents an analysis of the efficiency of ChatGPT models based on different prompt set sizes. 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 5.10 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.

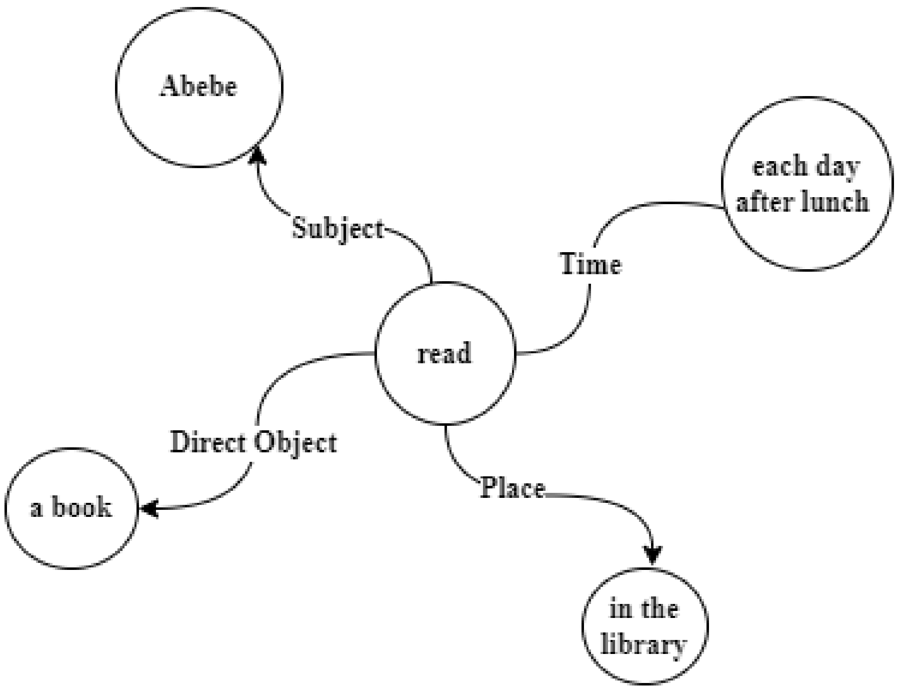


Figure 5.10: The overall evaluation result of linguistic experts

Table 5.1: 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

Efficiency, as reflected in the average quality rating of responses generated by these models, is a key measure. I have 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 disparities in performance between the OpenAI API and other models.

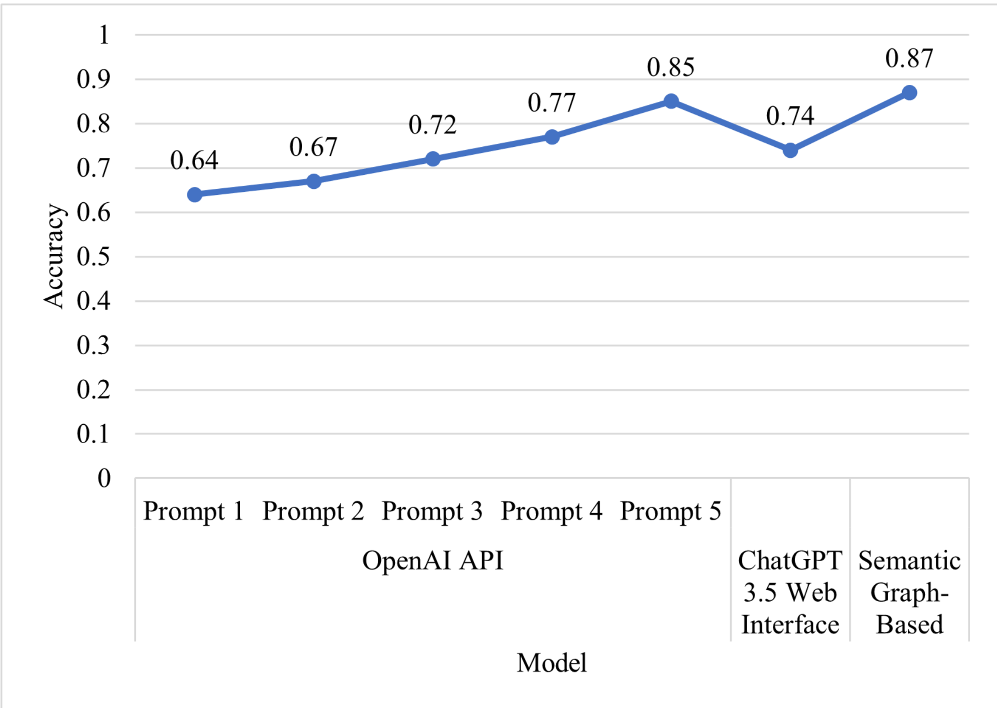


Figure 5.11: ChatGPT OpenAI and Hybrid parser-based Sentence Parsing Accuracy

Table 5.2: Efficiency of ChatGPT and Hybrid Parser-based Sentence Parsing method.

Model	Prompt size	Accuracy
OpenAI API	5	0.64
	15	0.67
	25	0.72
	35	0.77
	40	**0.85**
ChatGPT 3.5 Web Interface		0.74
Hybrid Parser-based Sentence Parsing		**0.87**

Figure 5.11 visually 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 5.1, 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.

Table 5.2 presents accuracy values, indicating that the Hybrid Parser-based sentence parsing method exhibit a slight advantage over the ChatGPT-based model (acc_GPT = 0.85, acc_hybrid = 0.87). This evaluation scenario provides valuable insights into the performance and effectiveness of both approaches in sentence parsing.

This experiment underscores that while ChatGPT 3.5 is a recent and versatile language model capable of generating diverse and interesting results, it has limitations, particularly in domains like sentence parsing. The observed accuracy values strongly advocate for the effectiveness of the proposed Hybrid Parser-based sentence parsing. This suggests that the proposed model may find broader applicability in sentence parsing tasks.

5.5 Summary

In summary, semantic graph construction serves as a foundational pillar in the areas of knowledge representation and artificial intelligence, providing structured meaning to the expansive domain of textual data. This process leverages a spectrum of foundational technologies, including NLP, dependency parsing, word embeddings, LDA, and the seamless integration of ontologies and knowledge graphs. These technological foundations empower the creation of semantic graphs, encom-

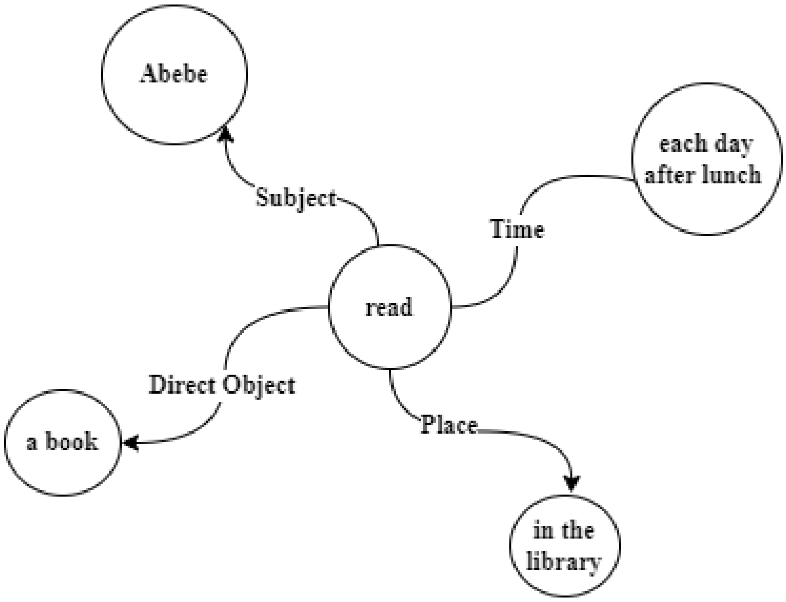


Figure 5.12: Semantic Graph Generated by the proposed model for the sentence “Abebe reads a book deeply in the library each day after lunch“

passing tasks from entity recognition to intricate topic modeling. Notably, two innovative approaches for sentence parsing in semantic graph induction have been introduced: the ChatGPT-based and Hybrid Parser-based methods. The following is a summary of the latest scientific findings.

Thesis 3. *I have developed two novel sentence parsing approaches that are based on deep neural network technologies. The first method uses the prompted version of ChatGPT while the second applies Hybrid Parser and neural network-based method. Through a comprehensive analysis, the Hybrid Parser-Based approach demonstrates a slight advantage in accuracy compared to ChatGPT in sentence parsing tasks. Notably, the Hybrid Parser consistently maintains "excellent" response quality, showcasing stability across various inputs, while ChatGPT's response quality varies with prompt sizes. The findings contribute to the broader field of natural language processing and offer valuable insights for practical applications, including information retrieval and knowledge graph development.* [**1**][**7**][**8**][**9**]

Chapter 6

Application of Sentence Parsing

In the area of Sentence Parsing applications, two important examples are AQG in ITS 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:

6.1 AQG in ITS

AQG within the Intelligent Tutoring System (ITS) is a tangible, real-world application of sentence parsing, highlighting the practical significance of advanced NLP techniques. ITS is a computer-based system that aims to offer direct and customized instruction or feedback to learners with personalized guidelines based on their cognitive skills, usually without requiring intervention from a human teacher[171]. Different researchers, designers, and developers define ITSs in different ways. According to Fletcher and Sottilare[172], intelligent tutoring may be viewed as "an effort to capture in computer technology the capabilities and practices of a human instructor who is an expert in both the subject matter and one-to-one tutoring.". In this application, sentence parsing plays a pivotal role in deconstructing instructional content into its syntactic and semantic components.

```
Subject=John
Verb=ate
Object=an apple
Time Adverb=yesterday
```

Figure 6.1: Functional structure for the the sentence "John ate an apple yesterday"

This parsing process is instrumental in breaking down sentences into syntactic and semantic elements, laying the foundation for effective question generation. In Figure 6.1, I observe the functional structure designed to represent the sentence "John ate an apple yesterday." This graphical representation illustrates how the different components of the sentence, such as the subject 'John,' the predicate 'ate,' the object 'an apple,' and the time reference 'yesterday,' are organized and interconnected.

The AQG process begins with the parsing of 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.

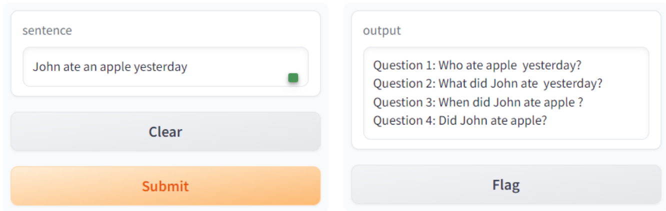


Figure 6.2: Sample Automatic Generated Questions for the sentence "John ate an apple yesterday"

Figure 6.2 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.

The functional structure provides a visual framework for understanding the syntactic and semantic elements within the sentence. This depiction aids in breaking down the sentence into its functional parts, offering insight into how each element contributes to the overall meaning. Applied in educational technology, assessment tools, and content generation systems, AQG involves parsing content to generate pertinent questions that assess the reader's understanding, demonstrating its practical relevance and impact on enhancing learning experiences.

- External Knowledge model
- Explicit knowledge of ontology
- Extending learner and tutor ML components
- Using NLP engines
- Question Generation Model.

To increase the performance of AI educational systems, more realistic student models and a better understanding of the pedagogical context are needed. Smart ITS systems will increase the possibilities for hard-working students to acquire more knowledge and skills. Most ITS systems have developed a focus on solving a single domain, and they try to create only a single course, like a C++ programming tutor or a Java object tutor. Most authoring tools like ITSB, CTAT GIFT, etc. use isolated database systems; these local databases can provide only a limited knowledge base. The limitations of an isolated database are a lack of reusability, a lack of standardization, a lack of flexibility, and limited knowledge. To overcome the problems mentioned before, I propose an extended ITS architecture model using a shared database. I expect that education in the future will use more insensible smart ITS tools.

The external databases can enhance the quality of the behavior models, both in tutor and student models. Implementation of this architecture is based on a new integration approach that includes existing methodologies and algorithms. The proposed extended ITS architecture is shown in Figure 6.3.

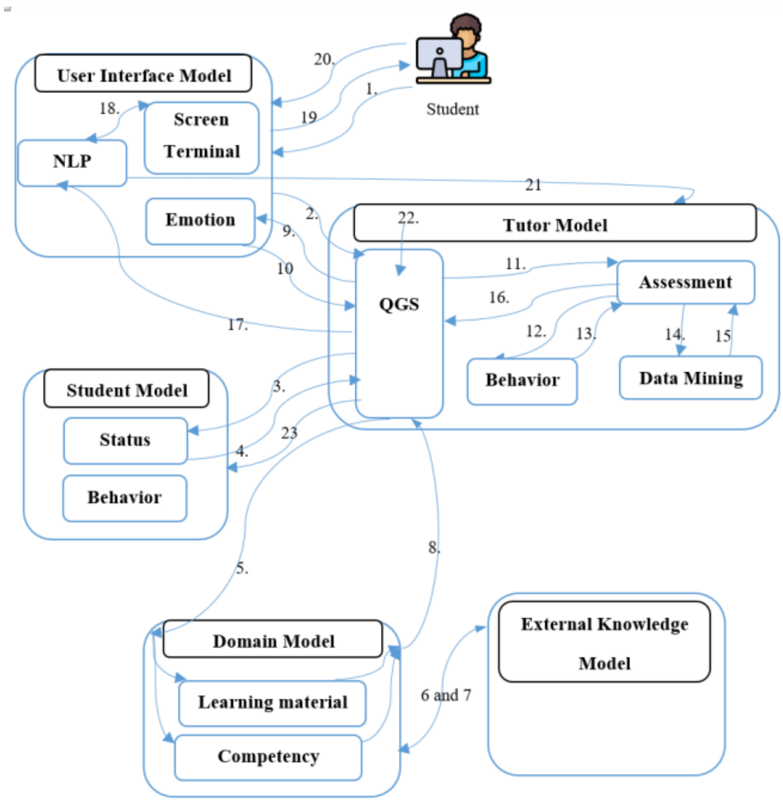


Figure 6.3: Extended ITS Architecture

The status in the student model is to indicate the current academic status of the student, which is upgraded in every assessment session. The status can be used to capture the current achievement of a student in a given topic. In the extended ITS architecture, I have used an adaptive behavior model. The knowledge model uses data mining and deep learning techniques for better decision-making, including classification and clustering of the student task.

The utilization of an extended sentence parsing method in the context of generating questions automatically based on the semantic content of the text has shown promising results. The evaluation carried out with the assistance of linguistic expertise, has indicated the effectiveness of the approach. However, there is still a need for improvements, particularly when dealing with long and complex sentences, as well as entire paragraphs.

To elaborate further, although the initial outcomes are positive, further enhancements are necessary to better handle intricate sentence structures and longer passages of text. This is essential to ensure that the system can accurately understand and generate questions across a wide range of linguistic complexities and contextual variations.

6.2 Generation of Ontology Semantic Graphs

Regarding the implementation of an ontology and NLP engine, Python is one of the most common languages used. It is an interpreted, object-oriented, extensible programming language[173], which provides an excellent combination of clarity and versatility in different disciplines. Information science offers many modules and packages for management and implementing ontology, data mining, and NLP. Many tools are available for building or managing an ontology. Regarding editing of the ontology by humans, Protégé is the most commonly used editor framework, which was created at Stanford University [174]. Protégé is free, open-source software to construct and update the ontology knowledge base. The tool has features for building, editing, and visualizing ontologies and importing and exporting capabilities of different ontology formats.

The proposed extended architecture model includes, besides the standard modules, a common shared database and knowledge-based background, too. The benefits of the global database are sharing a common understanding of the information structure, reusing the data, and mixing different sources of knowledge. 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 as input data in different data mining techniques, like NN.

In Figure 6.4, we can see a sample RDF graph corresponding to the sentence "Lalibela stands as an ancient rock-hewn church in Ethiopia." 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, ontology development often starts with

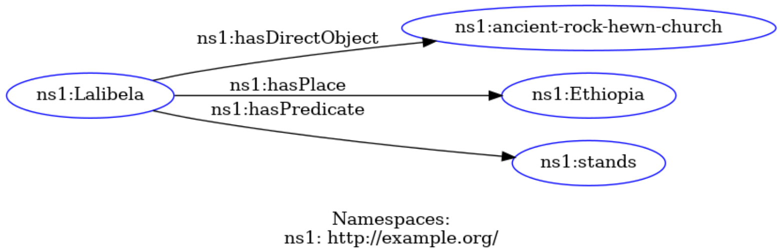


Figure 6.4: Sample RDF graph for sentence "Lalibela stands as an ancient rock-hewn church in Ethiopia"

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.

6.3 Evaluation and Result

The evaluation of the application of sentence parsing in the context of AQG in ITS and the creation of Ontology Semantic Graphs involves assessing the effectiveness, practicality, and impact of these applications. The results obtained from this evaluation shed light on the strengths, limitations, and potential areas for improvement in utilizing sentence parsing in educational technology and knowledge representation. The automated question-generation process yields positive outcomes in terms of generating relevant questions from parsed sentences. This indicates the potential for AQG in enhancing the efficiency of educational technology applications. While the initial results are promising, there is a recognized need for further enhancements, particularly in handling complex sentence structures and longer passages. Improvements are crucial to ensure the system's accuracy across a wide range of linguistic complexities and contextual variations.

Regarding the Generation of Ontology Semantic Graphs, the use of Python for ontology and NLP engine implementation is well-founded, considering its clarity and versatility. The choice aligns with industry standards and facilitates ease of development. The extended ITS architecture model incorporating a shared database is evaluated positively for its potential benefits, including common understanding, knowledge reuse, and integration of different knowledge sources. The RDF graphs, offer a clear and structured representation of information derived from natural language sentences. This visualization aids in understanding the encoding of content into RDF format. The connection between sentence parsing and ontology development is well-established. Sentence parsing serves as a foundational step in identifying key concepts, relationships, and attributes, contributing to the creation of structured ontology semantic graphs.

The evaluation underscores the importance of further research and development to fully harness the capabilities of sentence parsing in advancing intelligent tutoring and knowledge representation. The application of sentence parsing in both AQG in ITS and the creation of Ontology Semantic Graphs demonstrates substantial potential and positive outcomes. However, continuous improvement is essential to address challenges related to complex sentence structures.

6.4 Summary

Thesis 4.

I demonstrated the efficiency of the proposed method in two application domains. The first domain is the field of automatic question generation and the second refers to the automatic semantic graph induction. The AQG module was developed in Python as a prototype module in the Intelligent Tutoring System. The performance test result shows that the developed framework can be used in real applications.The second module can be used to generate RDF ontology descriptions from the free text data sources using our proposed sentence parser engines. The test result shows that the proposed module can be used to automate the process of ontology generation. [1][6]

Chapter 7

Summary

7.1 Contributions

This dissertation has made substantial contributions to the field of NLP and AQG within the context of text-to-semantic applications. The systematic exploration has led to the development and refinement of various sentence-parsing techniques, offering novel insights and advancements. The contributions can be summarized as follows:

Thesis 1: Extended Dependency Parsing

The introduction of a novel extended dependency parsing technique addresses challenges in dependency parsing within the context of AQG. The proposed system, featuring Ruleset Mapping for Question Generation and Question Word Selection for Question Generation algorithms, contributes significantly to the enhancement of sentence parsing methods. This advancement is crucial for achieving improved accuracy and effectiveness in text-to-semantic applications.

Thesis 2: Multilayer Perceptron-Based Sentence Parsing

The development of a Multilayer Perceptron-based Sentence Parsing Model represents a significant advancement in parsing accuracy, especially in handling complex linguistic structures. The model's ability to generate questions with a deeper semantic understanding showcases the potential of machine learning techniques in advancing AQG and text-to-semantic applications.

Thesis 3: Hybrid Parser for Semantic Graph Induction

The exploration of two distinct parsing approaches, the ChatGPT-based and Hybrid Parser-based methods, provides valuable insights into semantic graph induction. The Hybrid Parser-based approach, demonstrating a slight advantage in accuracy and stability across various inputs, contributes significantly to the broader field of NLP, particularly in AQG and semantic graph development.

Thesis 4:Application of Sentence Parsing

The dissertation extends its focus to the practical application of sentence parsing. A novel AQG has been developed using sentence parsing by including adverb type categorization and the creation of Ontology Semantic Graphs is explored in this chapter. The well-established connection between sentence parsing and ontology development reinforces the importance of parsing in identifying key concepts, relationships, and attributes for structured knowledge representation.

7.2 Future work

While the dissertation has made significant strides in the field of NLP and AQG, there are still a number of areas that could be explored and improved in the future:

1. Integration of Advanced Models: Further research can explore the integration of advanced machine learning models, including neural networks, to enhance question generation capabilities, particularly for complex sentence structures.
2. Hybrid Methodologies: Investigate hybrid approaches that combine rule-based systems with machine learning techniques to capitalize on their respective strengths, improving parsing accuracy and question generation.
3. Semantic Graph Refinement: Focus on refining the construction of semantic graphs, emphasizing detailed event descriptions and relationships within textual data for the development of more sophisticated knowledge graphs.
4. Response Verification Techniques: Address concerns about the accuracy of generated responses by exploring robust response verification techniques to ensure the quality and correctness of the output.
5. Ethical Considerations in Educational Applications: Conduct further research on the ethical implications of integrating technologies like ChatGPT into educational settings, and propose guidelines for responsible use to mitigate potential risks.

In conclusion, the dissertation sets the stage for continued advancements in NLP and AQG, offering an extended sentence parsing method that holds promise for text-to-semantic applications. The contributions made provide a solid foundation for future research, paving the way for refined language processing in the dynamic landscape of artificial intelligence.

Author's Publications

1 Journal Articles Related to the Dissertation

1. Sewunetie, W. T. & Kovács, L. Exploring Sentence Parsing: OpenAI API-Based and Hybrid Parser-Based Approaches. *IEEE Access* **12,** 38801–38815. doi:10.1109/ACCESS.2024.3360480 (2024).
2. Tewabe, W. & Kovacs, L. A Comparative Study of ChatGPT-based and Hybrid Parser-based Sentence Parsing Methods for Semantic Graph-based Induction. *International Journal of Advanced Computer Science and Applications (IJACSA)* **15.** doi:10.14569/IJACSA.2024.01501117. http://dx.doi.org/10.14569/IJACSA.2024.01501117 (2024).
3. Sewunetie, W. T. & Kovács, L. Automatic Question Generation Based on Sentence Structure Analysis and Dependency parsing approach. *Indonesian Journal of Electrical Engineering and Computer Science* **32.** Scopus Q3 Accepted for publication. doi:http://doi.org/10.11591/ijeecs.v32.i3 (2023).
4. Sewunetie, W. T. & Kovács, L. Comparison of template-based and multilayer perceptron-based approach for automatic question generation system. *Indonesian Journal of Electrical Engineering and Computer Science* **28.** Q3 Indexed, 1738–1748. doi:http://doi.org/10.11591/ijeecs.v28.i3.pp1738-1748 (2022).
5. Sewunetie, W. T. & Kovács, L. Evaluation of Python-Based NLP Frameworks for Template-Based Automatic Question Generation. *Production Systems and Information Engineering* **9,** 54–66. ISSN: 1785-1270. https://psaie.iit.uni-miskolc.hu/PSAIE2020.pdf (2020).
6. László, K. & Tewabe, S. W. Challenges in target selection for automatic question generation. *Multidiszciplináris Tudományok* **9,** 500–507. doi:https://doi.org/10.35925/j.multi.2019.4.53 (2019).
7. Tewabe Sewunetie, W., Hussein Ali Ahmed, G. & Kovács, L. The development and analysis of extended architecture model for intelligent tutoring systems. *GRADUS* **6,** 128–138. http://gradus.kefo.hu/archive/2019-4/2019_4_CSC_014_Sewunetie.pdf (2019).

2 Conference Proceedings Related to the Dissertation

8. Sewunetie, W. & Kovács, L. *ChatGPT in Education: Opportunities and Concerns for Functional English Sentence Structure Analysis for Automatic Question Generation* in *Proceedings of the International Conference on Advanced Science and Technology (ICAST)* Scopus Indexed Accepted Paper (2023). https://icast-conf.eai-conferences.org/2023/accepted-papers/.
9. Kovács, L., Csépányi-Fürjes, L. & Tewabe, W. *Transformer Models in Natural Language Processing* in *The 17th International Conference Interdisciplinarity in Engineering. Inter-ENG 2023* (eds Moldovan, L. & Gligor, A.) **929** (Springer, Cham, 2024). ISBN: 978-3-031-54673-0. doi:10.1007/978-3-031-54674-7_14. eprint: 978-3-031-54674-7. https://doi.org/10.1007/978-3-031-54674-7_14.
10. Sewunetie, W. T. & Kovács, L. *Efficiency Evaluation of ChatGPT for Adverb Type Categorization* in *2023 World Symposium on Digital Intelligence for Systems and Machines (DISA)* (2023), 199–203. doi:10.1109/DISA59116.2023.10308928.
11. Sewunetie, W. T. & Kovács, L. *Automatic Question Generation based on Sentence Structure Analysis* in *2023 24th International Carpathian Control Conference (ICCC)* Scopus Indexed (2023), 411–414. doi:10.1109/ICCC57093.2023.10178946. https://ieeexplore.ieee.org/document/10178946.
12. Sewunetie, W. T. & Kovács, L. *Comparison of Automatic Question Generation Techniques* in *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)* Scopus Indexed (2022), 000025–000030. doi:10.1109/CINTI-MACRo57952.2022.10029559.
13. Sewunetie, W. T. & Kovacs, L. *Rule-Based Automatic Question Generation Using Dependency Parsing* in *The 13th Conference of PhD Students in Computer Science* (2022), 192–196. https://www.inf.u-szeged.hu/~cscs/pdf/cscs2022.pdf.
14. Sewunetie, W. T. & László, K. *MLP Based Model for POS-Level Automatic Question Generation using Keyword Extraction* in *Abstract Book for the 17th Miklós Iványi International PhD & DLA Symposium: Architectural, Engineering and Information Sciences* (Pollack Press, Pécs, 2021).
15. Kovacs, L. & Walelign, T. S. *Automatic Question Generation using Template-Based Shallow NLP Approach* in *Doktorand Forum* (Nov. 2020).
16. László, K. & Sewunetie, W. *Closed Domain Template-Based Question Generation* in *16th Miklós Iványi International PhD & DLA Symposium* (2020), 127.

17. Kovács, L. & Walelign, T. S. *Evaluation of Template-Based Automatic Question Generation Techniques* in *XXIII. Tavaszi Szél Konferencia Absztrakt Kötet : Association of Hungarian PHD and DLA Students* (2020), 364. https://www.dosz.hu/__doc/dokumentumfile/2020/tavaszi-szel-2020-absztrakt-kotet-i-final.pdf.
18. Kovács, L. & Walelign, T. S. *A Comparative Survey of Automatic Question Generation Techniques* in *Doktoranduszok Fóruma* (Nov. 2019), 173–176.

3 Other Journal Articles

1. Gebretsadik, K. K. & Sewunetie, W. T. *Designing Machine Learning Method for Software Project Effort Prediction* 2019. doi:10.5923/j.computer.20190901.02.
2. Walelign, T. & Eshete, D. *Design and Architecture of Desktop LCD Monitor Having Built-in Webcam and Microphone with Multi-Angled Flexibility* 2016.
3. Walelign, T. & Eshete, D. *Design and Development of a Real-Time Virtualized Cloud-Based Mobile Phones Data Storage for Ethio Telecom* 2016. doi:DOI:10.9790/0050-03044751. https://api.semanticscholar.org/CorpusID:55505075.
4. Emiru, E. D. & Sewunetie, W. T. *Automatic Speech Segmentation for Amharic Phonemes Using Hidden Markov Model Toolkit (HTK)* 2016.
5. Sewunetie, W. T. & Emiru, E. D. *Real-Time Speech Synthesis by Handling Amharic Gemination Words Using Statistical Technique* 2016. https://api.semanticscholar.org/CorpusID:212510211.

4 Other Conferences

6. Sewunetie, W. T. *Amharic Text-to-Speech System* 2023. https://openreview.net/forum?id=h2GFfN9Oec.
7. Sewunetie, W. *et al.* *Development of A Virtual Store over Cloud for Mobile Phone and SIM Card Data* Raj Kumar Goal Institute of Technology, 2013.

References

1. Hinh, R. *Tool comparison of semantic parsers* Master's Thesis (Purdue University, Aug. 2016). https://docs.lib.purdue.edu/open_access_theses.
2. Rapaport, W. J. *Syntactic semantics: Foundations of computational natural-language understanding* in *Thinking computers and virtual persons* (Elsevier, 1994), 225–273.
3. Khurana, D., Koli, A., Khatter, K. & Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications* **82,** 3713–3744 (2023).
4. Gardner, M. *et al.* Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640* (2018).
5. Gašpar, A., Grubišić, A. & Šarić-Grgić, I. Evaluation of a rule-based approach to automatic factual question generation using syntactic and semantic analysis. *Language Resources and Evaluation,* 1–31 (2023).
6. Chowdhary, K. & Chowdhary, K. Natural language processing. *Fundamentals of artificial intelligence,* 603–649 (2020).
7. Agarwal, R. & Wadhwa, M. Review of state-of-the-art design techniques for chatbots. *SN Computer Science* **1,** 246 (2020).
8. Ghavami, P. *Big data analytics methods: analytics techniques in data mining, deep learning and natural language processing* (Walter de Gruyter GmbH & Co KG, 2019).
9. Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N. & Fox, E. A. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200* (2020).
10. Janiesch, C., Zschech, P. & Heinrich, K. Machine learning and deep learning. *Electronic Markets* **31,** 685–695 (2021).
11. Last, M. & Danon, G. Automatic question generation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10,** e1382 (2020).
12. Kurdi, G., Leo, J., Parsia, B., Sattler, U. & Al-Emari, S. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education* **30,** 121–204 (2020).

13. Chan, S. W. K. & Franklin, J. Dynamic context generation for natural language understanding: A multifaceted knowledge approach. *IEEE Transactions on systems, man, and Cybernetics-Part A: Systems and Humans* **33,** 23–41 (2003).

14. Dresscher, L. *Automated Generation of Assessment Questions from Textbook Models* MA thesis (2021).

15. Kumar, K. & Thakur, G. S. M. Advanced applications of neural networks and artificial intelligence: A review. *International journal of information technology and computer science* **4,** 57 (2012).

16. Goyal, A., Gupta, V. & Kumar, M. Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review* **29,** 21–43 (2018).

17. Kim, H., Yoo, J., Yoon, S., Lee, J. & Kang, J. Simple questions generate named entity recognition datasets. *arXiv preprint arXiv:2112.08808* (2021).

18. Zhu, R., Tu, X. & Huang, J. X. in *Deep learning for data analytics* 125–153 (Elsevier, 2020).

19. Hirschman, L. & Gaizauskas, R. Natural language question answering: the view from here. *natural language engineering* **7,** 275–300 (2001).

20. Borlund, P. The concept of relevance in IR. *Journal of the American Society for information Science and Technology* **54,** 913–925 (2003).

21. Cimiano, P., Unger, C. & McCrae, J. *Ontology-based interpretation of natural language* (Springer Nature, 2022).

22. Ding, Y., Ma, J. & Luo, X. Applications of natural language processing in construction. *Automation in Construction* **136,** 104169 (2022).

23. Žitko, B., Ljubić, H., *et al.* Automatic question generation using semantic role labeling for morphologically rich languages. *Tehnički vjesnik* **28,** 739–745 (2021).

24. Benamara, F., Taboada, M. & Mathieu, Y. Evaluative language beyond bags of words: Linguistic insights and computational applications. *Computational Linguistics* **43,** 201–264 (2017).

25. Mills, M. T. & Bourbakis, N. G. Graph-based methods for natural language processing and understanding—a survey and analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **44,** 59–71 (2013).

26. Jackson, P. & Moulinier, I. *Natural language processing for online applications: Text retrieval, extraction and categorization* (John Benjamins Publishing, 2007).

27. Galitsky, B. A., De La Rosa, J. L. & Dobrocsi, G. Inferring the semantic properties of sentences by mining syntactic parse trees. *Data & Knowledge Engineering* **81,** 21–45 (2012).

28. Dergaa, I., Chamari, K., Zmijewski, P. & Saad, H. B. From human writing to artificial intelligence generated text: examining the prospects and potential threats of ChatGPT in academic writing. *Biology of Sport* **40,** 615–622 (2023).

29. Thanaki, J. *Python natural language processing* (Packt Publishing Ltd, 2017).
30. Zhou, Y.-C., Zheng, Z., Lin, J.-R. & Lu, X.-Z. Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry* **142,** 103746 (2022).
31. Wu, S. *et al.* Structure-sensitive semantic matching for aggregate question answering over knowledge base. *Journal of Web Semantics* **74,** 100737 (2022).
32. Lee, U. *et al.* Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in english education. *Education and Information Technologies,* 1–33 (2023).
33. Ibrahim Teo, N. H. *Ontologies for automatic question generation* PhD thesis (University of Warwick, 2019).
34. Formica, A., Mele, I. & Taglino, F. A template-based approach for question answering over knowledge bases. *Knowledge and Information Systems,* 1–27 (2023).
35. Sewunetie, W. T. & Kovács, L. Comparison of template-based and multi-layer perceptron-based approach for automatic question generation system. *Indonesian Journal of Electrical Engineering and Computer Science* **28,** 1738–1748 (2022).
36. Ahmed, A. & Hanneman, G. Syntax-based statistical machine translation: A review. *Computational Linguistics* (2005).
37. Jouault, C. & Seta, K. *Building a semantic open learning space with adaptive question generation support* in *Proceedings of the 21st International Conference on Computers in Education* (2013), 41–50.
38. Heilman, M. *Automatic factual question generation from text* PhD thesis (Carnegie Mellon University, 2011).
39. Chali, Y. & Hasan, S. A. Towards topic-to-question generation. *Computational Linguistics* **41,** 1–20 (2015).
40. Sewunetie, W. T. & Kovács, L. Evaluation of Python Based NLP Frameworks for Template Based Automatic Question Generation. *Production Systems and Information Engineering* **9,** 54–66 (2020).
41. Danon, G. & Last, M. A syntactic approach to domain-specific automatic question generation. *arXiv preprint arXiv:1712.09827* (2017).
42. Fasya, E. L. *Automatic question generation for virtual humans* MA thesis (University of Twente, 2017).
43. Alsubait, T., Parsia, B. & Sattler, U. Ontology-based multiple choice question generation. *KI-Künstliche Intelligenz* **30,** 183–188 (2016).
44. Mhatre, K., Thube, A., Mahadeshwar, H. & Shrivas, A. Question Generation using NLP. *International Journal of Scientific Research & Engineering Trends* **5,** 2395–566 (2019).

45. Li, J., Sun, A., Han, J. & Li, C. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* **34,** 50–70 (2020).

46. Jakubíček, M. *Rule-based parsing of morphologically rich languages* PhD thesis (Masarykova univerzita, Fakulta informatiky, 2017).

47. Alkhawajah, F. I. *The Effect of Direct and Indirect Written Corrective Feedback on the Acquisition of Rule-Based and Item-Based Linguistic Features* Doctoral dissertation (York University, Toronto, Ontario, Aug. 2016).

48. Zhang, R., Guo, J., Chen, L., Fan, Y. & Cheng, X. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)* **40,** 1–43 (2021).

49. Kusuma, S. F., Siahaan, D. O. & Fatichah, C. Automatic question generation with various difficulty levels based on knowledge ontology using a query template. *Knowledge-Based Systems* **249,** 108906 (2022).

50. Cohen, F. S. What is a Question? *The monist,* 350–364 (1929).

51. Alshboul, J. & Baksa-Varga, E. A Review of Automatic Question Generation in Teaching Programming. *International Journal of Advanced Computer Science and Applications* **13** (2022).

52. Echihabi, A. & Marcu, D. *A noisy-channel approach to question answering* in *Proceedings of the 41st annual meeting of the association for computational linguistics* (2003), 16–23.

53. Heilman, M. & Smith, N. A. *Good question! statistical ranking for question generation* in *Human language technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics* (2010), 609–617.

54. Mostow, J. & Chen, W. *Generating Instruction Automatically for the Reading Strategy of Self-Questioning.* in *AIED* (2009), 465–472.

55. Serban, I., Sordoni, A., Bengio, Y., Courville, A. & Pineau, J. *Building end-to-end dialogue systems using generative hierarchical neural network models* in *Proceedings of the AAAI conference on artificial intelligence* **30** (2016).

56. Zhou, Q. *et al. Neural question generation from text: A preliminary study* in *Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6* (2018), 662–671.

57. Das, B., Majumder, M., Phadikar, S. & Sekh, A. A. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning* **16,** 1–15 (2021).

58. Sarrouti, M., Ben Abacha, A. & Demner-Fushman, D. Goal-driven visual question generation from radiology images. *Information* **12,** 334 (2021).

59. Yousuf, H., Lahzi, M., Salloum, S. A. & Shaalan, K. A systematic review on sequence-to-sequence learning with neural network and its models. *International Journal of Electrical & Computer Engineering (2088-8708)* **11** (2021).

60. Liu, B. *Neural question generation based on Seq2Seq* in *Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence* (2020), 119–123.
61. Boguslavsky, I. *Semantic Analysis based on linguistic and ontological resources* in *Proceedings of the 5th International Conference on Meaning-Text Theory (MTT'2011)* (2011), 25–36.
62. Deblanc, S. *Transformer based fill-in-the-gap question generation for online assessment* Master's thesis (Ghent University, Ghent, Belgium, 2021).
63. Fattoh, I. E. Semantic based automatic question generation using artificial immune system. *Computer Engineering and Intelligent Systems* **5,** 74–82 (2014).
64. Deng, Y. *et al.* Contextualized knowledge-aware attentive neural network: Enhancing answer selection with knowledge. *ACM Transactions on Information Systems (TOIS)* **40,** 1–33 (2021).
65. Yao, X., Bouma, G. & Zhang, Y. Semantics-based question generation and implementation. *Dialogue and Discourse* **3,** 11–42 (2012).
66. Sewunetie, W. T. & Kovács, L. *Automatic Question Generation based on Sentence Structure Analysis* in *2023 24th International Carpathian Control Conference (ICCC)* (2023), 411–414.
67. Chakankar, T., Shinkar, T., Waghdhare, S., Waichal, S. & Phadatare, M. M. M. Automated Question Generator using NLP. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* **11.** ISSN: 2321-9653 (3 Mar. 2023).
68. McKeown, K. *Text generation* (Cambridge University Press, 1992).
69. Jusoh, S. A study on NLP applications and ambiguity problems. *Journal of Theoretical & Applied Information Technology* **96** (2018).
70. Kaur, J. & Bathla, A. K. A review on automatic question generation system from a given Hindi text. *International Journal of Research in Computer Applications and Robotics (IJRCAR)* **3,** 87–92 (2015).
71. Fernández, M. *et al.* Semantically enhanced information retrieval: An ontology-based approach. *Journal of Web Semantics* **9,** 434–452 (2011).
72. Dhole, K. D. & Manning, C. D. Syn-QG: Syntactic and shallow semantic rules for question generation. *arXiv preprint arXiv:2004.08694* (2020).
73. McClosky, D., Charniak, E. & Johnson, M. *Reranking and self-training for parser adaptation* in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (2006), 337–344.
74. Bailey, P., De Vries, A. P., Craswell, N. & Soboroff, I. *Overview of the TREC 2007 Enterprise Track.* in *TREC* (2007).
75. Atbaker. *Wikipedia Question Generator* https://github.com/atbaker/wikipedia-questiongenerator. 2023.

76. Mulla, N. & Gharpure, P. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence* **12,** 1–32 (2023).
77. Wołk, K. & Marasek, K. Enhanced bilingual evaluation understudy. *arXiv preprint arXiv:1509.09088* (2015).
78. Owczarzak, K., Conroy, J., Dang, H. T. & Nenkova, A. *An assessment of the accuracy of automatic evaluation in summarization* in *Proceedings of workshop on evaluation metrics and system comparison for automatic summarization* (2012), 1–9.
79. Vijaymeena, M. & Kavitha, K. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal* **3,** 19–28 (2016).
80. Katinskaia, A. & Yangarber, R. *Assessing grammatical correctness in language learning* in *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications* (2021), 135–146.
81. Navimipour, N. J. & Zareie, B. A model for assessing the impact of e-learning systems on employees' satisfaction. *Computers in Human Behavior* **53,** 475–485 (2015).
82. Säljö, R. Digital tools and challenges to institutional traditions of learning: technologies, social memory and the performative nature of learning. *Journal of computer assisted learning* **26,** 53–64 (2010).
83. Pan, Z. *et al.* Recent progress on generative adversarial networks (GANs): A survey. *IEEE access* **7,** 36322–36333 (2019).
84. László, K. & Tewabe, S. W. Challenges in target selection for automatic question generation. *Multidiszciplináris Tudományok* **9,** 500–507 (2019).
85. Das, B., Majumder, M., Phadikar, S. & Sekh, A. A. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning* **16(1),** 1–15 (2021).
86. Novák, A. & Novák, B. Cross-lingual transfer of knowledge in distributional language models: Experiments in Hungarian. *Acta Linguistica Academica* **69,** 405–449 (2022).
87. Nivre, J., Basirat, A., Dürlich, L. & Moss, A. Nucleus composition in transition-based dependency parsing. *Computational Linguistics* **48,** 849–886 (2022).
88. Kübler, S., McDonald, R. & Nivre, J. in *Dependency parsing* 11–20 (Springer, 2009).
89. Osborne, T. *et al.* NPs, not DPs: The NP vs. DP debate in the context of dependency grammar. *Acta Linguistica Academica. An International Journal of Linguistics (Until 2016 Acta Linguistica Hungarica)* **68,** 274–317 (2021).
90. Zhang, X., Du, Y., Sun, W. & Wan, X. Transition-based parsing for deep dependency structures. *Computational Linguistics* **42,** 353–389 (2016).
91. McDonald, R. & Nivre, J. Analyzing and integrating dependency parsers. *Computational Linguistics* **37,** 197–230 (2011).

92. Duan, K. *et al.* Enhancement of Question Answering System Accuracy via Transfer Learning and BERT. *Applied Sciences* **12,** 11522 (2022).
93. Yusuf, A. A., Nwojo, N. A. & Boukar, M. M. *Basic dependency parsing in natural language inference* in *2017 13th International Conference on Electronics, Computer and Computation (ICECCO)* (2017), 1–4.
94. El Bazi, I. & Laachfoubi, N. Arabic named entity recognition using deep learning approach. *International Journal of Electrical and Computer Engineering (IJECE)* **9,** 2025–2032 (2019).
95. Mazidi, K. & Tarau, P. *Infusing nlu into automatic question generation* in *Proceedings of the 9th International Natural Language Generation conference* (2016), 51–60.
96. Sewunetie, W. T. & Kovács, L. *Comparison of automatic question generation techniques* in *2022 IEEE 22nd international symposium on computational intelligence and informatics and 8th IEEE international conference on recent achievements in mechatronics, automation, computer science and robotics (CINTI-MACRo)* (2022), 000025–000030.
97. Fauzi, M. A. Word2Vec model for sentiment analysis of product reviews in Indonesian language. *International Journal of Electrical and Computer Engineering* **9,** 525 (2019).
98. Fasold, R. W. & Connor-Linton, J. *An introduction to language and linguistics* (Cambridge university press, 2014).
99. Afzal, N. & Mitkov, R. Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing* **18,** 1269–1281 (2014).
100. Maheen, F. *et al.* Automatic computer science domain multiple-choice questions generation based on informative sentences. *PeerJ Computer Science* **8,** e1010 (2022).
101. Zhao, Y., Ni, X., Ding, Y. & Ke, Q. *Paragraph-level neural question generation with maxout pointer and gated self-attention networks* in *Proceedings of the 2018 conference on empirical methods in natural language processing* (2018), 3901–3910.
102. Forsbom, E. *Training a super model look-alike* in *Workshop on Systemizing MT Evaluation* (2003).
103. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521,** 436–444 (2015).
104. Takase, S. & Kobayashi, S. All word embeddings from one embedding. *Advances in Neural Information Processing Systems* **33,** 3775–3785 (2020).
105. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* **26** (2013).
106. Ling, W. *et al.* *Not all contexts are created equal: Better word representations with variable attention* in *Proceedings of the 2015 conference on empirical methods in natural language processing* (2015), 1367–1372.

107. Xu, C. *et al. Rc-net: A general framework for incorporating knowledge into word representations* in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (2014), 1219–1228.

108. Team, G. C. *Introduction to Google Colab* https://colab.research.google.com/notebooks/intro.ipynb. 2021.

109. Nema, P. & Khapra, M. M. Towards a better metric for evaluating question generation systems. *arXiv preprint arXiv:1808.10192* (2018).

110. Singh, J. & Sharma, Y. Encoder-decoder architectures for generating questions. *Procedia computer science* **132,** 1041–1048 (2018).

111. Csáji, B. C. *et al.* Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary* **24,** 7 (2001).

112. Rusu, D., Fortuna, B., Grobelnik, M. & Mladenić, D. Semantic graphs derived from triplets pplicwith aation in document summarization. *Informatica* **33** (2009).

113. Zou, X. *A survey on application of knowledge graph* in *Journal of Physics: Conference Series* **1487** (2020), 012016. doi:10.1088/1742-6596/1487/1/012016.

114. Shi, L. *et al.* Semantic health knowledge graph: semantic integration of heterogeneous medical knowledge and services. *BioMed research international* **2017.** doi:10.1155/2017/2858423 (2017).

115. Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S. & Sontag, D. Learning a health knowledge graph from electronic medical records. *Scientific reports* **7,** 5994. doi:10.1038/s41598-017-05778-z (2017).

116. Liu, J., Lu, Z. & Du, W. *Combining Enterprise Knowledge Graph and News Sentiment Analysis for Stock Price Volatility Prediction* in *Proceedings of the 52nd Hawaii International Conference on System Sciences* (2019). ISBN: 978-0-9981331-2-6. https://hdl.handle.net/10125/59565.

117. Elnagdy, S. A., Qiu, M. & Gai, K. *Cyber incident classifications using ontology-based knowledge representation for cybersecurity insurance in financial industry* in *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)* (2016), 301–306. doi:10.1109/CSCloud.2016.45.

118. Zablith, F. Constructing social media links to formal learning: A knowledge Graph Approach. *Educational technology research and development* **70,** 559–584. doi:10.1007/s11423-022-10091-2 (2022).

119. Jia, Y., Qi, Y., Shang, H., Jiang, R. & Li, A. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering* **4,** 53–60. doi:10.1016/j.eng.2018.01.004 (2018).

120. Cresswell, M. J. & Cresswell, M. Adverbs of space and time. *Adverbial Modification: Interval Semantics and Its Rivals,* 41–66. doi:10.1007/978-94-009-5414-4_3 (1985).

121. Liu, Y. *et al.* Summary of ChatGPT-Related Research and Perspective Towards the Future of Large Language Models. *Meta-Radiology,* 100017. doi:10.1016/j.metrad.2023.100017 (2023).

122. Abdullah, M., Madain, A. & Jararweh, Y. *ChatGPT: Fundamentals, applications and social impacts* in *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (2022), 1–8. doi:10.1109/SNAMS58071.2022.10062688.

123. Vassiliou, G., Papadakis, N. & Kondylakis, H. *SummaryGPT: Leveraging ChatGPT for summarizing knowledge graphs* in *European Semantic Web Conference* (2023), 164–168. doi:10.1007/978-3-031-43458-7_31.

124. Pichappan, P., Krishnamurthy, M. & Vijayakumar, P. Analysis of ChatGPT as a Question-Answering Tool. *Journal of Digital Information Management* **21,** 51. doi:10.6025/jdim/2023/21/2/50-61 (2023).

125. De Barros Pereira, H. B. *et al.* Systematic review of the "semantic network" definitions. *Expert Systems with Applications,* 118455. doi:10.1016/j.eswa.2022.118455 (2022).

126. Hamilton, W. L. *Graph representation learning* (Morgan & Claypool Publishers, 2020).

127. Chen, Y. *et al.* A Survey on Multimodal Knowledge Graphs: Construction, Completion and Applications. *Mathematics* **11,** 1815. doi:10.3390/math11081815 (2023).

128. Peng, C., Xia, F., Naseriparsa, M. & Osborne, F. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review,* 1–32. doi:10.1007/s10462-023-10465-9 (2023).

129. Kamath, A. & Das, R. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978* (2018).

130. Lehmann, J. *et al.* Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* **6,** 167–195. doi:10.3233/SW-140134 (2015).

131. Čebirić, Š. *et al.* Summarizing semantic graphs: a survey. *The VLDB journal* **28,** 295–327. doi:10.1007/s00778-018-0528-3 (2019).

132. Banu, A. in *Semantic Web Technologies* 223–244 (CRC Press, 2022). doi:10.1201/9781003309420.

133. Wohlgenannt, G. *Learning Ontology Relations by Combining Corpus-Based Techniques and Reasoning on Data from Semantic Web Sources* (Peter Lang International Academic Publishers, 2018).

134. Harth, A., Hose, K. & Schenkel, R. *Linked data management* (CRC Press, 2014).

135. Chauhan, S. Comprehensive review of coronavirus disease 2019 (COVID-19). *Biomedical journal* **43,** 334–340 (2020).

136. Goel, S. & Uzuner, O. Do sentiments matter in fraud detection? Estimating semantic orientation of annual reports. *Intelligent Systems in Accounting, Finance and Management* **23,** 215–239 (2016).

137. Wright, B. & Garrett, M. Lexical decision in sentences: Effects of syntactic structure. *Memory & Cognition* **12,** 31–45 (1984).
138. Singh, S. Natural language processing for information extraction. *arXiv preprint arXiv:1807.02383* (2018).
139. Chiche, A. & Yitagesu, B. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data* **9,** 1–25 (2022).
140. Brill, E. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics* **21,** 543–565 (1995).
141. Kocmi, T. Exploring benefits of transfer learning in neural machine translation. *arXiv preprint arXiv:2001.01622* (2020).
142. Kolomiyets, O. & Moens, M.-F. A survey on question answering technology from an information retrieval perspective. *Information Sciences* **181,** 5412–5434 (2011).
143. Zachary, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research* **33,** 452–473 (1977).
144. Pham, P., Nguyen, L. T., Pedrycz, W. & Vo, B. Deep learning, graph-based text representation and classification: a survey, perspectives and challenges. *Artificial Intelligence Review* **56,** 4893–4927. doi:10.1007/s10462-022-10265-7 (2023).
145. Kovacs, L. Mapping Between Semantic Graphs and Sentences in Grammar Induction System. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience* **1,** 100–112 (2010).
146. Hu, W. *et al.* Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430* (2021).
147. Goyal, P. & Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* **151,** 78–94. doi:10.1016/j.knosys.2018.03.022 (2018).
148. Ji, S., Pan, S., Cambria, E., Marttinen, P. & Philip, S. Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems* **33,** 494–514. doi:10.1109/TNNLS.2021.3070843 (2021).
149. Abu-Salih, B. Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications* **185,** 103076. doi:10.1016/j.jnca.2021.103076 (2021).
150. Singh, H., Jain, P., Chakrabarti, S., *et al.* Multilingual knowledge graph completion with joint relation and entity alignment. *arXiv preprint arXiv:2104.08804* (2021).
151. Desmarais, F.-X., Gagnon, M. & Zouaq, A. *Comparing a rule-based and a machine learning approach for semantic analysis* in *Proceedings of the 6th International Conference on Advances in Semantic Processing, Barcelona, Spain* (2012), 103–8.

152. Nanni, M. *Disambiguating Italian homographic heterophones with SoundChoice and testing ChatGPT as a data-generating tool* 2023.
153. Martínez-Cruz, R., López-López, A. J. & Portela, J. ChatGPT vs State-of-the-Art Models: A Benchmarking Study in Keyphrase Generation Task. *arXiv preprint arXiv:2304.14177* (2023).
154. Gondaliyaa, C. P., Patel, A. M. & Parikh, S. M. *A Comparative Study on Machine Learning Based Algorithms* in *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)* (2018), 26–27.
155. Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research* **3,** 993–1022 (2003).
156. Boz, O. *Extracting decision trees from trained neural networks* in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002), 456–461.
157. Vemprala, S., Bonatti, R., Bucker, A. & Kapoor, A. *ChatGPT for Robotics: Design Principles and Model Abilities* tech. rep. MSR-TR-2023-8 (Microsoft, Feb. 2023). https://www.microsoft.com/en-us/research/publication/chatgpt-for-robotics-design-principles-and-model-abilities/.
158. Ray, P. P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems* **3,** 121–154. ISSN: 2667-3452. doi:https://doi.org/10.1016/j.iotcps.2023.04.003. https://www.sciencedirect.com/science/article/pii/S266734522300024X (2023).
159. Li, K. *et al.* Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023).
160. Lund, B. D. *et al.* ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology* **74,** 570–581 (2023).
161. Kocoń, J. *et al.* ChatGPT: Jack of all trades, master of none. *Information Fusion,* 101861 (2023).
162. Ray, P. P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems* (2023).
163. Naveed, H. *et al.* A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435* (2023).
164. Kotei, E. & Thirunavukarasu, R. A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning. *Information* **14,** 187. doi:10.3390/info14030187 (2023).
165. Bowman, S. R. Eight things to know about large language models. *arXiv preprint arXiv:2304.00612* (2023).
166. Brown, T. *et al.* Language models are few-shot learners. *Advances in neural information processing systems* **33,** 1877–1901 (2020).

167. Manning, C. D. Human language understanding & reasoning. *Daedalus* **151,** 127–138. doi:10.1162/daed_a_01905 (2022).

168. Meskó, B. Prompt Engineering as an Important Emerging Skill for Medical Professionals: Tutorial. *Journal of Medical Internet Research* **25,** e50638. doi:10.2196/50638 (2023).

169. Hanif, M. A. *et al.* Highly efficient and sustainable ZnO/CuO/g-C3N4 photocatalyst for wastewater treatment under visible light through heterojunction development. *Catalysts* **12,** 151 (2022).

170. Chen, Y., Li, H., Qi, G., Wu, T. & Wang, T. Outlining and Filling: Hierarchical Query Graph Generation for Answering Complex Questions Over Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering* (2022).

171. Al Rekhawi, H. A. Android applications development intelligent tutoring system (2020).

172. Dermeval, D., Paiva, R., Bittencourt, I. I., Vassileva, J. & Borges, D. Authoring tools for designing intelligent tutoring systems: a systematic review of the literature. *International Journal of Artificial Intelligence in Education* **28,** 336–384 (2018).

173. Tewabe Sewunetie, W., Hussein Ali Ahmed, G. & Kovács, L. The development and analysis of extended architecture model for intelligent tutoring systems. *GRADUS* **6,** 128–138 (2019).

174. Horridge, M. *et al. Simplified OWL ontology editing for the web: is WebProtégé enough?* in *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12* (2013), 200–215.

175. Alkhatlan, A. & Kalita, J. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *arXiv preprint arXiv:1812.09628.* doi:https://doi.org/10.48550/arXiv.1812.09628 (2018).

176. Sottilare, R. A., Durlach, P. J. & Fletcher, J. - A REVIEW OF INTELLIGENT TUTORING SYSTEMS FOR COLLECTIVE (TEAM) TRAINING. *Assessment of Intelligent Tutoring Systems Technologies and Opportunities.*

177. Pogge, T. & Sengupta, M. The Sustainable Development Goals (SDGS) as drafted: Nice idea, poor execution. *Wash. Int'l LJ* **24,** 571 (2015).

178. Ling, Y., Cai, F., Liu, J., Chen, H. & de Rijke, M. Generating Relevant and Informative Questions for Open-Domain Conversations. *ACM Transactions on Information Systems* **41,** 1–30 (2023).

179. Soni, S., Kumar, P. & Saha, A. *Automatic Question Generation: A Systematic Review* in *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India* (2019).

180. Le, N.-T., Kojiri, T. & Pinkwart, N. *Automatic question generation for educational applications–the state of art* in *Advanced Computational Methods for Knowledge Engineering: Proceedings of the 2nd International Conference on Computer Science, Applied Mathematics and Applications (ICCSAMA 2014)* (2014), 325–338.
181. Swali, D., Palan, J. & Shah, I. Automatic question generation from paragraph. *International Journal of Advance Engineering and Research Development* **3,** 73–78 (2016).
182. Piñón, C. Adverbs of completion in an event semantics. *Perspectives on aspect,* 149–166. doi:10.1007/1-4020-3232-3_8 (2005).
183. Mabrouk, A., Hantach, R. & Calvez, P. in *Proceedings of the 4th day "Automatic Language Processing and Artificial Intelligence"-TALIA 2018* (CEUR-WS. org, 2018).
184. Richardson, K. New resources and ideas for semantic parser induction (2018).
185. Hormay, E. *et al.* The effect of loss of the glucose-monitoring neurons in the anterior cingulate cortex: Physiologic challenges induce complex feeding-metabolic alterations after local streptozotocin microinjection in rats. *Neuroscience Research* **149,** 50–60 (2019).
186. Zhou, K., Yang, J., Loy, C. C. & Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision* **130,** 2337–2348 (2022).
187. Wang, A. *et al.* GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).
188. Mochales, R. & Moens, M.-F. Argumentation mining. *Artificial Intelligence and Law* **19,** 1–22 (2011).
189. Li, Y. *et al. NTIRE 2023 challenge on efficient super-resolution: Methods and results* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 1921–1959.
190. Lund, B. D. & Wang, T. Chatting about ChatGPT: how may AI and GPT impact academia and libraries? *Library Hi Tech News* **40,** 26–29 (2023).
191. Ray, T. P. *et al.* Outflows from the youngest stars are mostly molecular. *Nature* **622,** 48–52 (2023).
192. De Ruiter, J. P. *Questions: Formal, functional and interactional perspectives* (Cambridge University Press, 2012).
193. Sewunetie, W. T. & Kovács, L. *Comparison of automatic question generation techniques* in *2022 IEEE 22nd international symposium on computational intelligence and informatics and 8th IEEE international conference on recent achievements in mechatronics, automation, computer science and robotics (CINTI-MACRo)* (2022), 000025–000030.

194. Sewunetie, W. & Kovács, L. *ChatGPT in Education: Opportunities and Concerns for Functional English Sentence Structure Analysis for Automatic Question Generation* in *Proceedings of the International Conference on Advanced Science and Technology (ICAST)* Accepted Paper (2023). https://icast-conf.eai-conferences.org/2023/accepted-papers/.

195. Kovács, L., Csépányi-Fürjes, L. & Tewabe, W. *Transformer Models in Natural Language Processing* in *Proceedings of the INTER-ENG 2023 Conference* Accepted Paper (Springer, https://inter-eng.umfst.ro/2023/, 2023).

196. Özateş, Ş. B., Özgür, A., Güngör, T. & Başaran, B. Ö. A Hybrid Deep Dependency Parsing Approach Enhanced With Rules and Morphology: A Case Study for Turkish. *IEEE Access* **10,** 93867–93886 (2022).