

A Machine Learning Approach for Generating a Recursive Object Model from a Natural Language Text

Amin Bayatpour

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

August 2023

© Amin Bayatpour, 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Amin Bayatpour**

Entitled: **A Machine Learning Approach for Generating a Recursive Object Model from a Natural Language Text**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

Dr. Jun Yan Chair

Dr. Mazdak Nik-bakht External Examiner

Dr. Jun Yan Examiner

Dr. Yong Zeng Supervisor

Approved by

Dr. Chun Wang, Director
Department of Concordia Institute for Information Systems Engineering

August 2023

Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

A Machine Learning Approach for Generating a Recursive Object Model from a Natural Language Text

Amin Bayatpour

This research investigates the potential of machine learning algorithms as an alternative approach to rule-based systems for generating Recursive Object Model (ROM) diagrams. The existing rule-based approach suffers from limitations and challenges, and this study aims to explore the possibility of overcoming these limitations by leveraging machine learning techniques. To achieve the research objectives, software was developed to gather labelled data for our supervised learning problem. A model comprised of Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) models was created and trained using the labelled data. The proposed model takes a pair of words and a sentence as inputs and classifies the appropriate relations among the pairs. Subsequently, a comprehensive evaluation was conducted to assess the effectiveness of the proposed model. The evaluation process involved a comparative analysis between the proposed model and a baseline model, an evaluation of the proposed model on unseen data, and an investigation into the capability of the design model in addressing the limitations of the rule-based system. The evaluation results demonstrate the superiority of the proposed model. Firstly, the proposed model achieved an exceptional accuracy of 97 percent in the training process, surpassing the baseline model's accuracy of approximately 61 percent. Secondly, the proposed model exhibited an accuracy of 96 percent on unseen data, thus showcasing its ability to generalize effectively to new instances. Lastly, when comparing the proposed intelligent system with the rule-based system, although the proposed methodology exhibited minor errors in generating ROM diagrams for certain scenarios, the findings underscore the potential of the proposed model in mitigating the limitations of the rule-based system.

Acknowledgments

I would like to take this opportunity to express my deep and sincere appreciation to my respected supervisor, Dr. Yong Zeng. His constant support and invaluable guidance have been a guiding light throughout my academic journey. His expert insights and continuous encouragement have not only enhanced my understanding of the subject but have also played a significant role in shaping me as a learner and an individual.

I am also immensely grateful to my family and friends who have been unwavering pillars of strength and motivation throughout these two remarkable years. Their constant presence and words of encouragement have made this academic pursuit all the more meaningful and enjoyable. Their belief in me has fueled my determination to overcome challenges and achieve milestones.

Moreover, I feel truly fortunate to have had the privilege of working within the inspiring atmosphere of the design lab. This nurturing environment has not only provided me with the resources necessary for my research but has also fostered an atmosphere of creativity and curiosity. The experiences gained within this space have broadened my horizons and added a rich dimension to my academic journey.

I extend my heartfelt gratitude to the esteemed members of the committee for their insightful feedback and thoughtful considerations during the thesis process. Their expertise and constructive critiques have played an instrumental role in refining and elevating the quality of my work.

Contents

List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Research Objective	4
1.4 Outline	5
2 Literature Review	6
2.1 Introduction	6
2.2 Environment Based Design and Recursive Object Model	7
2.2.1 Environment-Based Design	7
2.2.2 Recursive Object Model	10
2.3 Natural Language Processing	11
2.3.1 Overview	11
2.3.2 Techniques in Natural Language Processing	14
2.3.3 Applications and Limitations	15
2.4 Machine Learning in Natural Language Processing	17
2.4.1 Overview of Supervised, Unsupervised, Semi-supervised and Reinforce- ment Learning	17

2.4.2	Machine Learning Algorithms in NLP	20
3	Methodology	30
3.1	Introduction	30
3.2	Research Methodology	30
3.2.1	Data Collection	31
3.2.2	Proposed Model Architecture	37
3.2.3	Training Results	45
4	Validation of the proposed methodology	54
4.1	Introduction	54
4.2	Evaluation Methods	55
4.3	Testing Data	57
4.3.1	Data Collection	57
4.3.2	Data Prepration	58
4.4	Result and Evaluation	59
4.4.1	Baseline model vs Proposed model	59
4.4.2	Evaluating Proposed Methodology	63
4.4.3	Proposed Model vs Rule-based system	64
4.5	Discussion and Conclusion	70
5	Conclusion and future works	72
5.1	Conclusion	72
5.2	Limitations and Future works	73
	Bibliography	76

List of Figures

Figure 1.1	An example of Chomsky Tree for a sentence.	2
Figure 1.2	An example of a ROM diagram generated by the ROMA software.	3
Figure 2.1	Design Process	7
Figure 2.2	EBD Design Process Zeng (2015).	8
Figure 2.3	Machine Learning Techniques (Sarker, 2021)	18
Figure 2.4	A simple RNN (Lipton, 2015)	24
Figure 3.1	An illustration of inputting a sentence in the ROMWeb software.	32
Figure 3.2	An illustration of result the ROM diagram in the ROMWeb software.	32
Figure 3.3	An illustration of Editing the result of ROM diagram in the ROMWeb software.	33
Figure 3.4	Overview of the system.	38
Figure 3.5	Overview of the architecture of the designed system.	39
Figure 3.6	Overview of the designed model's components.	41
Figure 3.7	Accuracy of the proposed model in 10 folds	48
Figure 3.8	Loss of the proposed model in 10 folds	49
Figure 4.1	Baseline Model which is used for validating the proposed model.	55
Figure 4.2	Demonstrating a problem in the Chomsky tree in detecting some verbs as nouns.	56
Figure 4.3	Accuracy of the baseline model in 10 folds	61
Figure 4.4	Loss of the baseline model in 10 folds	62

List of Tables

Table 2.1	Five basic symbols in the ROM (Zeng, 2008).	11
Table 2.2	Summary of Machine learning algorithms	29
Table 3.1	The Predefined patterns for collected sentences	34
Table 3.2	Confusion Matrix	46
Table 3.3	Comparing predicted and actual relationship on validation data after training the proposed model	51
Table 3.4	Classification report for the evaluation of the proposed methodology for vali- dation data	51
Table 3.5	Compare the accuracy of the proposed model without and with regularization	52
Table 4.1	The Predefined patterns for collected sentences for testing the methodology. .	56
Table 4.2	Classification report for the evaluation of the baseline model for validation data	60
Table 4.3	The validation result for the baseline model for the best-performing fold. . .	60
Table 4.4	The accuracy of the proposed model vs the baseline model	63
Table 4.5	Classification report for the evaluation of the proposed methodology for test- ing data	65
Table 4.6	The result of evaluating the trained model for the best-performing fold on the testing data	65

Chapter 1

Introduction

1.1 Motivation

An essential tool in the Environment-based Design methodology is the Recursive Object model (ROM), which offers a graphical representation of text written in natural language (Zeng, 2008). This representation of natural language is interpreted as the "brain" of the methodology since it is crucial for decision-making and following processes. The rule-based software known as ROMA, which is employed in the current implementation of ROM, as seen in the picture 1.2, is built on the Chomsky tree. The Chomsky tree is a linguistic model that seeks to capture the hierarchical structure of sentences in natural language, as seen in figure 1.1. It organizes words and phrases into a tree-like structure, with each node representing a constituent part of the sentence. With the help of the Chomsky tree, the syntactic linkages seen in real language have been captured and turned into ROM relations in the existing rule-based system. Despite the rule-based ROMA software's efficacy, there is a drive to investigate other methods for producing ROM diagrams. This research determines if rule-based systems can be replaced or improved upon using machine learning techniques. It tries to alleviate the possible drawbacks of the rule-based approach by utilizing the capabilities of machine learning algorithms. The goal is to use machine learning methods to extract intricate connections and patterns in plain language, enabling more precise and effective development of ROM relationships. The project aims to examine the potential of machine learning-based systems in upgrading the ROM tool and its usefulness in Environment-based Design methodology by expanding

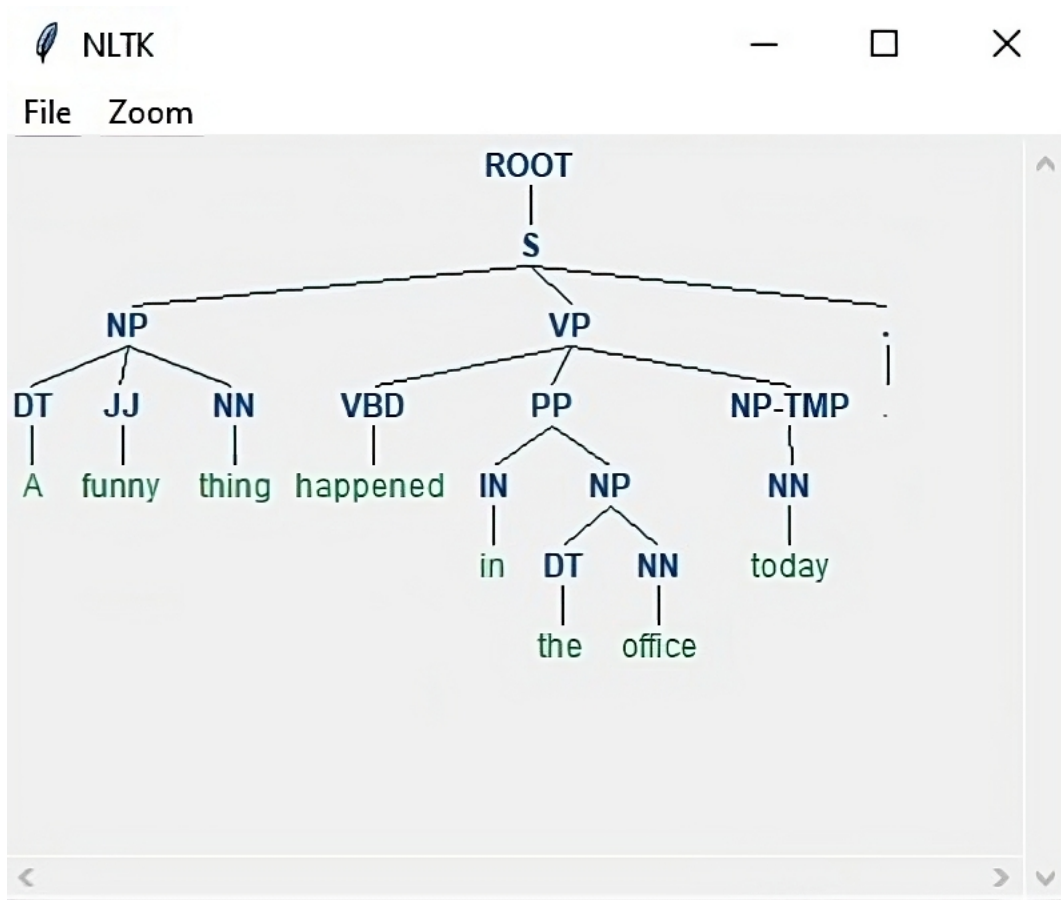


Figure 1.1: An example of Chomsky Tree for a sentence.

beyond the rule-based approach.

1.2 Problem Statement

A key aspect of this study involves exploring the application of machine learning algorithms as potential replacements for the existing rule-based method in generating ROM relations. If a ROM diagram can be produced, replacing the current rule-based system with the new system would be the next stage. As previously indicated, ROM relations are now generated using practices and knowledge that mainly rely on the rule-based software ROMA, which is based on the Chomsky tree linguistic model. Even though this method has shown promise in capturing syntactic relationships, the following difficulties and restrictions have to be addressed:

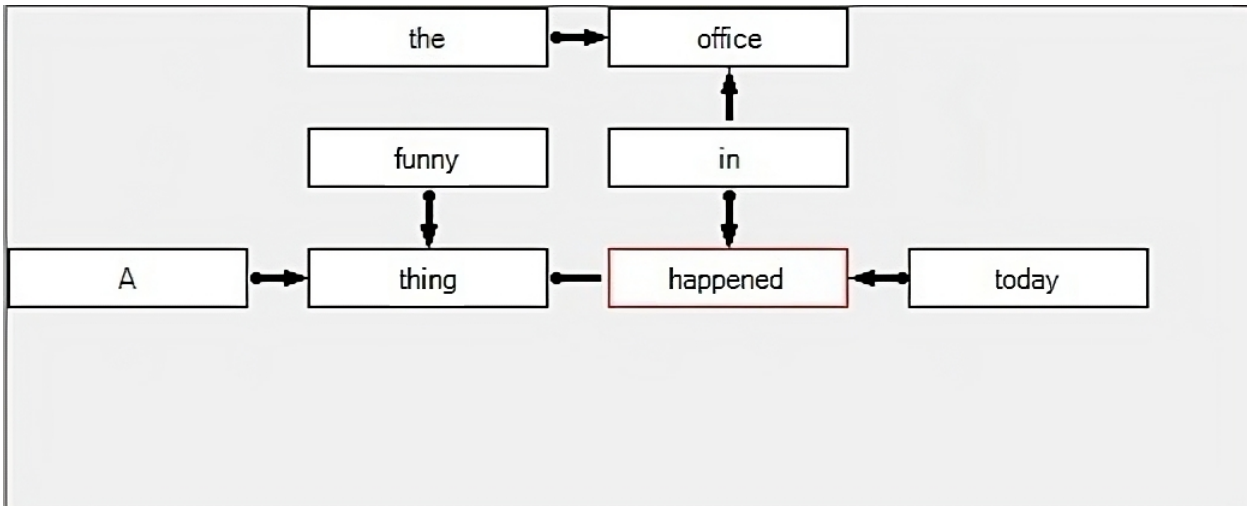


Figure 1.2: An example of a ROM diagram generated by the ROMA software.

- **Limited Adaptability:** Since rule-based systems rely on predetermined rules and might not be able to handle all linguistic patterns, they frequently struggle to adapt to new linguistic patterns and structures.
- **Lack of Contextual Understanding:** ROMA software may not have a thorough comprehension of the semantic and contextual aspects of natural language since it is built on the Chomsky tree which largely concentrates on syntactic interactions.
- **Knowledge Acquisition and Maintenance:** A rule-based system needs a lot of human effort and expertise to create and maintain. This can be a time-consuming and resource-intensive procedure that includes manually creating and changing rules based on linguistic knowledge. This is a serious issue since it is difficult to maintain the rule-based system as language changes and new linguistic patterns appear.
- **Difficulty in Handling Ambiguity:** Natural language frequently has ambiguous forms and multiple possible meanings. Such ambiguity may be difficult for rule-based systems to handle, which might result in inaccurate or inconsistent ROM results. Clarifying ambiguity necessitates more complex language processing abilities, which are frequently outside of the scope of rule-based methods.

- **Scalability and Generalization:** Scaling up and generalizing rule-based systems to various domains or languages may be challenging. It can be difficult to create complete rulesets that account for every possible language structure. This constraint limits the system's suitability and adaptation to various linguistic situations.
- **Manual Rule Construction:** Constructing rules for a rule-based system, such as ROMA, typically relies on linguistic expertise and manual effort. This manual process introduces subjectivity and the potential for errors or biases in rule creation, which can impact the accuracy and reliability of the generated ROM relations.

Although the rule-based method has been successful, this investigation aims to offer a new viewpoint and explore the possible advantages of using machine learning-based systems to improve the ROM tool. By incorporating machine learning algorithms, the proposed method offers adaptability and the ability to learn from data, resulting in a more accurate representation of natural language in the ROM diagrams.

1.3 Research Objective

The study intends to evaluate the capability of the new system in several predetermined language patterns, suggesting the feasibility of producing the ROM diagram by using the power of machine learning. It also aims to automate knowledge acquisition and maintenance for ROM creation. By minimizing the need for manual rule development, this objective seeks to increase efficiency and ensure that the system remains abreast of changing developments in language. The manual maintenance tasks associated with rule-based systems often prove time-consuming and resource-intensive, making automation an essential avenue for improving the efficiency of ROM creation.

In summary, this research strives to investigate the potential of machine learning algorithms as an alternative to rule-based systems for generating ROM diagrams. The research objectives encompass exploring the feasibility, evaluating the effectiveness, assessing scalability and adaptability, enhancing contextual comprehension, automating knowledge acquisition and maintenance, and mitigating the limitations of the rule-based system.

1.4 Outline

Chapter 2 offers a comprehensive understanding of the fundamental concepts that form the basis for developing the methodology used in the study. The chapter is divided into three sections, each focusing on distinct areas of knowledge essential for comprehending and effectively applying the methodology. The conceptual frameworks laid out in this chapter serve as the groundwork for the subsequent implementation and analysis.

Chapter 3 provides a comprehensive overview of the research approach and techniques employed in the study. It outlines the research design and methodology selection, explaining the rationale behind the chosen approach.

Chapter 4 presents a summary of the process employed to evaluate the performance of the developed model. It provides an overview of the performance evaluation measures used and describes the data or experiments utilized for validation purposes. The chapter showcases the results and analysis of the model's performance, shedding light on its effectiveness.

Chapter 5 serves as a concluding chapter, summarizing the findings and outcomes of the research. It discusses the implications and contributions of the study, highlighting its significance in the field. It also acknowledges the study's limitations and suggests potential directions for future research. The conclusion chapter concludes the thesis by offering final thoughts and insights.

Chapter 2

Literature Review

2.1 Introduction

Chapter 2 provides a brief summary of the fundamental concepts that form the basis for developing the methodology employed in this study. By providing a comprehensive understanding of these concepts, this chapter lays the groundwork for the subsequent implementation and analysis. The chapter is divided into three sections, each focusing on distinct areas of knowledge that are crucial for comprehending and applying the methodology effectively. The section 2.2 aims to establish a solid foundation by exploring the concepts of Environment-based Design (EBD) and the Recursive Object Model. Moving forward, section 2.3 focuses on understanding Natural Language Processing (NLP). It begins with an overview of NLP, highlighting its importance and relevance in various domains. The section then illustrates the common techniques employed in NLP such as named entity recognition, and Part-of-Speech tagging. Furthermore, the applications and limitations of NLP across different fields will be demonstrated. Finally, section 2.4 examines the integration of machine learning in NLP. An overview of machine learning is presented, providing an understanding of its underlying principles and techniques. The section then delves into the common machine learning algorithms, as mentioned in table 2.2 employed specifically in NLP tasks.

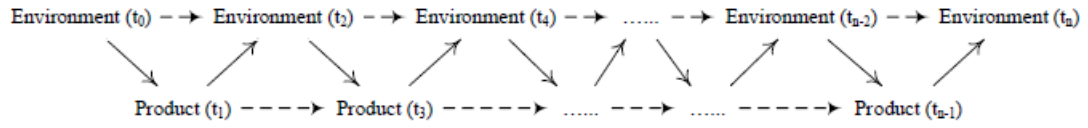


Figure 2.1: Design Process

2.2 Environment Based Design and Recursive Object Model

The Recursive Object Model (ROM), one of the key elements of this technique, and the core concept of Environment-Based Design will both be covered in this part. The section clarifies the relevance of taking into account the environment and its influence on the design process through an in-depth discussion of EBD concepts. The Recursive Object Model is also presented as a tool in the EBD methodology that makes it possible to represent and analyze natural language.

2.2.1 Environment-Based Design

Design can be considered as an intuitive human activity that has the purpose to change the existing environment to a desired environment by creating or designing a new solution that can enhance or even change the existing environment, as is shown in figure 2.1. To satisfy this purpose, there is a standard procedure called Environment-Based Design (EBD) Which can be used as a guideline to guide a designer through the life cycle of the design process.

As the name of this methodology, the environment is the key element, where everything is seen as the environment except the product in Environment-Based Design (Zeng, 2021). By following EBD, the structure of the environment will be addressed by questioning the structure and the lifecycles of the environment, understanding its importance and analyzing the environment (Zeng, 2021).

As mentioned above, EBD is an approach assisting designers to find their paths in the iteration of environmental changes and satisfying certain requirements (Zeng, 2015):

- (1) It helps designers in breaking out from the recursive loop that exists between design problems, design knowledge and design solution.
- (2) It leads to both routine and creative design.

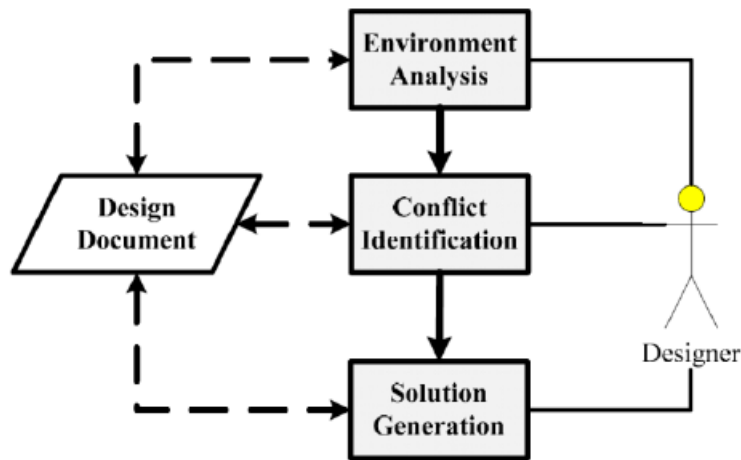


Figure 2.2: EBD Design Process [Zeng \(2015\)](#).

(3) It helps designers to maintain their mental stress at an optimal level during the design process.

There are three main activities in EBD: environment analysis, conflict identification, and solution generation. As described in figure 2.2, the current environment will be defined by asking and answering questions related to the problem in environmental analysis, and then undesired conflicts will be identified in the conflict Identification section. Finally, by resolving the disputes between interactions, the candidate solutions can be concluded.

2.2.1.1 Environment Analysis

In the design process, the identification of the product's environment plays a crucial role as it enables designers to gain a comprehensive understanding of the components within the environment and their interrelationships. This understanding can be facilitated through the formulation of two types of questions: generic questions and domain questions. Generic questions, as outlined by [\(Zeng, 2015\)](#), involve the generation of questions that aid designers in comprehending the design problem. By utilizing the Recursive Object Model (ROM) diagram, designers are able to navigate the design problem more effectively, thereby enhancing their understanding. The ROM diagram assists designers in determining the order in which questions should be posed, thereby emphasizing the significance of each component in addressing the design problem. On the other hand, domain

questions are aimed at gathering information that has a substantial impact on the specific problem at hand. In order to accomplish this, it is necessary to define all the components of the product's environment and their relationships, without taking into consideration any preconceived knowledge about design requirements or solutions. The identification of the product's environmental elements entails the initial identification of life cycle events, which are subsequently classified into three categories: natural, built, and human (Zeng, 2015). Moreover, within the Environment-Based Design (EBD) framework, the product's environment is further categorized into seven distinct classes, encompassing both the life cycle and events of the product (Chen & Zeng, 2006).

These methods provide designers with a methodical, thorough awareness of the context in which their products will be used. A strong basis for the design process is established by this understanding, which was obtained via the study of generic and domain questions as well as the categorization of environmental components.

2.2.1.2 Conflict Identification

A conflict arises when there is a scarcity of resources for an object to carry out a desired action on its environment or to accommodate the consequences of the object's action on its environment (Zeng, 2015). Within this conceptual framework, conflicts can be categorized into two types: active conflicts and reactive conflicts. Active conflicts pertain to the conflicts that are essential for initiating action, whereas reactive conflicts are critical for sustaining the ongoing action. The identification of conflicts requires a systematic approach. Firstly, verbs referred to as interactions in the Recursive Object Model (ROM) diagram are identified. Next, the relationships between these interactions are examined to determine which interactions are dependent on others. These dependencies are then utilized to construct an interaction dependence network, which serves as a graphical representation of the relationships between interactions. Finally, this network is employed as a means to identify and analyze conflicts (Zeng, 2015). This approach makes it possible to recognize and comprehend conflicts within the context of the conceptual framework. This method offers a well-organized and thorough investigation of the conflicts that could develop when items interact with the environment.

2.2.1.3 Solution Generation

Once conflicts have been identified within the Environmental-Based Design (EBD) methodology, a systematic procedure is followed to generate potential solutions. In this procedure (Zeng, 2015), when dealing with reactive conflicts, designers are required to explore possible approaches such as resource rearrangement, object optimization, action optimization, or the creation of new elements in order to resolve the reactive conflicts effectively. Conversely, in the case of active conflicts, designers distinguish between primitive conflicts and non-primitive conflicts. If the conflict is primitive, designers can directly propose a solution. However, if the conflict is non-primitive, designers engage in a process of decomposition and knowledge acquisition until the conflict is reduced to its primitive form. This iterative process continues until the desired product is achieved, and all conflicts are effectively addressed and resolved.

2.2.2 Recursive Object Model




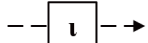
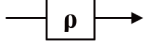
The design solution and design problem could be represented by Geometric Models, Sketches, Graphic language, Natural language, and Mathematical language in the design process. All these problems and solutions are expressed by human thinking and their ideas, which are represented by natural language. Therefore, the most crucial challenge for the industries is transforming all these texts into structured representations. In addition, this transformation process is highly time-consuming; therefore, it would be a game-changer to find a way to describe and transform natural language into a graphical representation automatically. Nowadays, natural language processing has become a critical component of computer-aided design systems since these systems would positively impact the quality of communications throughout the design process, facilitate understanding the customer's real intent, and elicit precise and complete product requirements (Zeng, 2008). The Recursive Object Model is a tool providing a graphic representation of natural language. Axiomatic theory, a logical tool for representing and reasoning about object structures, is used to create a ROM diagram (Zeng, 2002). In this theory, objects and their relations in the universe are primitive concepts, and two main axioms in this theory are (Zeng, 2008):

- Axiom 1: Everything in the universe is an object.

- Axiom 2: There are relations between objects

According to the first axiom, we can call everything in the universe an object, and therefore, the basic unit in the ROM is an object, and there is a compound object in the ROM that contains two or more objects. In addition, in terms of the second axiom, there are three kinds of relations in the ROM diagram which are constraint, connection, and predicate relations (Zeng, 2008). A constraint relation is a relation between two objects in which one of them changes the meaning of another, in other words, one of them limits, or particularizes the other. A connection relation is a relation between two objects; however, they do not change the meaning of each other. Lastly, a predicate relation is a relation that describes the act of one object on another. In summary, as is shown in Table 2.1, there are five basic symbols in the ROM diagram, which are object, compound object, constraint relation, connection relation, and predicate relation.

Table 2.1: Five basic symbols in the ROM (Zeng, 2008).

	Type	Graphic Representation	Description
Object	Object		Everything in the universe is an object
	Compound Object		It is an object that includes at least two objects in it
Relation	Constraint Relation		It is a descriptive, limiting, or particularizing relation of one object to another
	Connection		It is to connect two objects that do not constrain each other
	Predicate Relation		It describes an act of an object on another or that describes the states of an object

2.3 Natural Language Processing

2.3.1 Overview

A language is a standard group of rules and symbols used by humans to transfer and convey information (Khurana, Koli, Khatter, & Singh, 2023). In order to achieve human-like language

processing for a variety of activities or applications, natural language processing (NLP) is a theoretically grounded set of computer approaches for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis (Liddy, 2001). In this definition, there are multiple terms which need to be analyzed and reviewed:

- **Range of computational techniques** refers to the multiple methods or techniques from which to choose to accomplish a particular type of language analysis
- **Naturally occurring texts** refers to any language, mode, or genre, generated by humans for the communication purpose
- **Levels of linguistic analysis** refers to the various stages at which linguistic information can be processed and understood by machine learning models
- **Human-like Language Processing** refers to the fact that NLP is considered a discipline within Artificial Intelligence
- **For a range of tasks or applications** points out that NLP is not usually considered a goal itself, however, it is the means for accomplishing a particular task such as machine translation, question answering and etc

Natural Language Processing's primary objective is to process language that is similar to human speech and to understand natural language, as was noted in the list above. Understanding the language refers to a variety of activities, including paraphrasing an input text, translating the text into another language, responding to inquiries about the text's contents, and drawing conclusions from the text.

In general, Natural Language Processing is classified into Natural Language Understanding (NLU), and Natural Language Generation (NLG). NLU can be achieved through using some techniques such as Concept Extraction, Entity Detection, Emotion Analysis, Keyword Extraction, etc (Khurana et al., 2023).

There are several levels of NLU which are mentioned as follows (Liddy, 2001):

- (1) **Phonology**: At this level, it is feasible to carry out research on sound, particularly on how sound is arranged systematically and how it behaves and is organized in language.

- (2) **Morphology:** The study of morphemes, the smallest units of meaning in words, is known as morphology.
- (3) **Lexical:** Words are processed at the lexical level through part-of-speech tagging, semantic representation, and structural analysis.
- (4) **Syntactic:** words are grouped at this level to generate meaningful and correct structures called clauses and phrases.
- (5) **Semantic:** Semantic processing determines the feasible meaning of a sentence by processing its logical structure to recognize the most relevant words to understand the interactions among words or different concepts in the sentence.
- (6) **Discourse** Anaphora resolution and co-reference resolution, which aid in establishing the link between words and sentences, are part of the discourse level of NLP, which deals with analyzing the logical structure between sentences to maintain coherence.
- (7) **Pragmatic:** While semantic analysis concentrates on the literal meaning of words, pragmatic NLP analyses the context to create a meaningful representation of the text, deals with implicit meaning, and aids in comprehension of what is being discussed in the text.

In contrast, NLG is the process of producing meaningful phrases, sentences and paragraphs from an internal representation which has the following main sections ([Khurana et al., 2023](#)):

- (1) **Speaker and Generator:** The intentions will be transferred into a fluent phrase to explain the situation.
- (2) **Components and Level of Representation:** Selection of linguistic resources, organization of the textual material, and realization of the output as text or voice are all steps in the language creation process.
- (3) **Application or Speaker:** The text describes how the speaker maintains the model of the situation by structuring potentially relevant content and selecting a subset of propositions without participating in language generation.

2.3.2 Techniques in Natural Language Processing

It can be difficult for computer systems to decipher the intended meaning of written or spoken input due to the complexity of human language. Various essential NLP tasks, such as named entity recognition, part of speech tagging, word sense disambiguation, and sentiment analysis, are briefly introduced in this section. (Collobert et al., 2011).

2.3.2.1 Part-Of-Speech Tagging

Assigning each word a unique tag that indicates its grammatical function, such as noun, adjective, etc., is the goal of this task. Rule-based, stochastic, artificial neural networks and hybrid approaches are among the most well-known methods for identifying the POS tagging for each word. (Chiche & Yitagesu, 2022). A rule-based method of POS tagging provides labels to words in a sentence using unique rules derived from the linguistic characteristics of the language. This can be done by linguistic specialists or by machine learning on an annotated corpus (Brill, 1992).

2.3.2.2 Word Sense Disambiguation

For many years, natural language processing has employed the word sense disambiguation technique to identify the precise sense of the ambiguous word (Ide & Veronis, 1998). If the word "mouse" appears in the statement, for example, it might be interpreted as animal sense and computer sense. This approach makes it feasible to distinguish between the word's many meanings. While there are various methods that can be used to achieve this task, the three primary categories in NLP are knowledge-based, supervised, and unsupervised techniques. (Ranjan Pal & Saha, 2015).

Even though it may not be required for all NLP applications, this technique can nonetheless improve the performance of such systems. Many NLP applications, such as machine translation, information retrieval systems, voice processing, and grammatical analysis, employ this technique to enhance performance and get a deeper understanding of the natural language. (Ide & Veronis, 1998).

2.3.2.3 Named Entity Recognition

An important entity may be referred to as a person, company, or location in everyday speech. These essential components of natural language may be recognized using a technique called Named Entity Recognition, which belongs to the field of NLP. In the area of natural language processing, a number of approaches and algorithms are available for identifying significant textual parts, such as rule-based linear models, supervised learning techniques, and multitask models. (Roy, 2021).

This technique has been applied to several additional applications and industries. Biologists frequently employ this technique to identify DNA, drug names, and even disease names. Additionally, it may be used alongside subject identification to enhance search queries by recognizing important aspects. Furthermore, since some items must match in order for the translation to be successful, NER may be used to find these elements and improve translation performance (Roy, 2021).

2.3.2.4 Sentiment Analysis

Opinions are one of the most important elements of all human undertakings since they are the major influencer of behaviour for both our own actions and those of others. The same idea applies to companies, who seek out specific clients or members of the general public who share their opinions about their good or service(Liu, 2012). In the NLP, there is a technique called Sentiment Analysis, or also opinion mining, which analyses people's opinions, sentiments, and attitudes in the text.

Sentiment analysis has three different levels of analysis: document level, sentence level, and entity level. The document-level analysis determines if the entire text represents a positive or negative attitude, whereas the purpose of the sentence-level analysis is to identify those sentiments in the sentence. The two first levels' main flaw is that they cannot determine what people liked or didn't like, which is something that can be done at the entity level. (Liu, 2012).

2.3.3 Applications and Limitations

Natural Language Processing allows us to finish numerous time-consuming jobs more quickly and effectively while also drastically reducing our workload. The following are some of the most common NLP applications:

- **Information retrieval:** A well-known illustration of an information retrieval system is Google. The process of documenting, storing, and studying a collection of data in order to extract knowledge and find relevant outcomes that satisfy the user's needs is known as information retrieval. Many other industries, like digital libraries and media search engines, use this software for their purposes. ([Ibrihich, Oussous, Ibrihich, & Esghir, 2022](#)).
- **Question-Answering:** This application provides a list of potentially relevant documents in response to user queries. This technique has been used in many systems which are required to answer the user's questions automatically ([Liddy, 2001](#)).
- **Text Summarizing:** The higher level of natural language processing which reduces a larger text as a shorter text. This application represents a huge text into a more structured and narrative text ([Liddy, 2001](#)).
- **Machine Translation:** Although maintaining the meaning of phrases as well as syntax and tenses while doing machine translation is a challenging problem, improvements in artificial neural networks and deep learning have increased the quality of machine translation, and several techniques have been developed to assess their quality. ([Khurana et al., 2023](#)).
- **Dialogue System:** In real-world applications, dialogue systems are frequently used for support as well as taking action, with context awareness being essential for support systems. Earlier systems were only capable of tiny uses; however, current systems can make use of all linguistic levels and allow human-like conversation with machines. ([Khurana et al., 2023](#)).
- **Medicine:** NLP is used in the field of medicine for a variety of projects, including the National Library of Medicine's Specialist System for information extraction and the Linguistic String Language Processor, which enables doctors to extract and summarise information for identifying potential drug side effects. Other initiatives involve developing an NLP system called MEDLEE for recognizing clinical information in narrative reports and creating a vocabulary from medical dictionaries. ([Khurana et al., 2023](#)).

Besides all the useful applications, the field of NLP faces several challenges. These challenges include dealing with contextual words and phrases, synonyms, homonyms, sarcasm, ambiguity,

informal phrases, and culture-specific lingo. Misspelled or misused words can also pose a problem. Although NLP models have improved over time, there is still a need for models that are applicable to a broader range of people and languages.

2.4 Machine Learning in Natural Language Processing

A subfield of artificial intelligence and computer science called Machine Learning utilizes data and algorithms to simulate how people learn, progressively increasing the accuracy of its predictions. Artificial intelligence (AI) is the phrase used to describe the ability of a computer to think and reason, which is the goal of machine learning. The rapidly expanding field of data science includes machine learning as a key element. The term machine learning (ML) has been created to address more complex issues like speech recognition and picture categorization because AI does not incorporate a learning process. In the machine learning approach, algorithms are created based on the data provided, whereas in the coding approach, algorithms are created and written by programmers based on the intended outcome. Algorithms are taught to create classifications or predictions and to find significant insights in data mining projects via the use of statistical approaches. Following that, these insights inform business and application choices, perhaps having an influence on crucial growth indicators.

2.4.1 Overview of Supervised, Unsupervised, Semi-supervised and Reinforcement Learning

As mentioned in [2.3](#) There are four main methods and approaches for solving different tasks in machine learning: Supervised learning, Unsupervised learning, Semi-Supervised learning and Reinforcement learning which will be discussed in the next section.

2.4.1.1 Supervised Learning

In order to find patterns, or function f , relating inputs, called x , to outputs, commonly designated Y , this approach requires a training set of input-output examples. Supervised learning may be classified as either a regression problem or a classification problem depending on the output type.

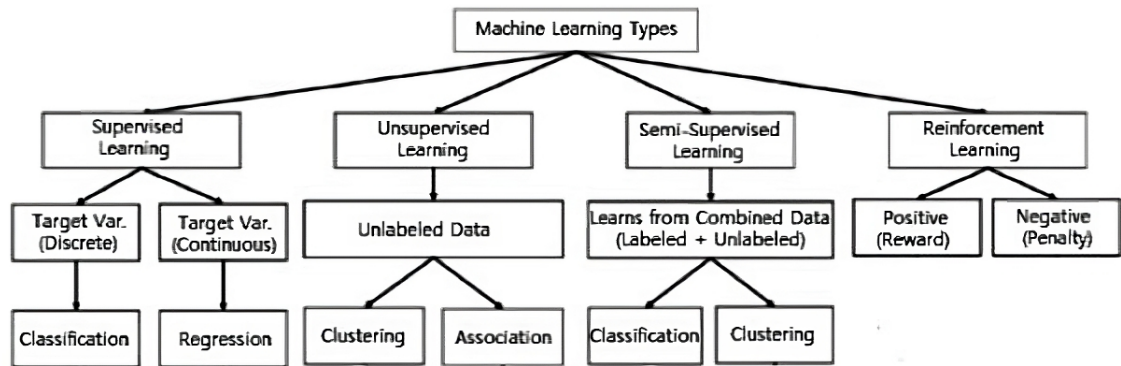


Figure 2.3: Machine Learning Techniques (Sarker, 2021)

The issue is referred to as a regression problem if the outputs are continuous (Simeone, 2018). The goal of both regression and classification is to derive from the training data set a predictor that generalizes the input-output mapping for the data that is not present in the training data. A data set is obtained and divided into separate training, validation, and test data sets. The training and validation data sets are then used to create a model that captures the relationship between features and target variables. Finally, the test data set is used to evaluate the model's performance and determine its predictive power (Choi, Coyner, Kalpathy-Cramer, Chiang, & Campbell, 2020).

2.4.1.2 Unsupervised Learning

Without explicit supervision or labelled examples, unsupervised learning uses algorithms to investigate and find hidden structures or patterns in the data. To understand the underlying patterns, these algorithms examine the natural connections and parallelisms between data points. They may categorize or combine related data points, apply labels based on traits, or locate clusters or relationships within the data set (Hinton, Sejnowski, Hinton, & Sejnowski, 1999). Unsupervised learning offers numerous inherent advantages. Firstly, it facilitates the identification of patterns and insights that may elude manual analysis, thereby unveiling latent trends, anomalies, or underlying structures that hold significant value for decision-making and subsequent analysis. Secondly, unsupervised learning algorithms exhibit efficiency in handling vast and intricate data sets, making them

particularly well-suited for tasks involving data exploration, dimensional reduction, and data pre-processing. Finally, unsupervised learning algorithms possess the capacity to generalize effectively to novel, unobserved data, as their focus lies in capturing the inherent distribution and structure of the data rather than relying solely on specific labelled examples. This characteristic empowers them to extract meaningful knowledge from new data instances without relying on explicit supervision, thus augmenting their applicability and adaptability in various domains ([Hinton et al., 1999](#)).

2.4.1.3 Semi-supervised Learning

A combination of supervised and unsupervised learning techniques, semi-supervised learning is a methodology that uses both labelled and unlabeled data. This method is especially beneficial when there are few training examples with labels available compared to a large amount of unlabeled data and producing labels requires a lot of resources or is expensive. Semi-supervised learning seeks to enhance overall learning performance and overcome the problems associated with data scarcity in supervised learning situations by utilizing unlabeled data in addition to the restricted amount of labelled data ([Chapelle, Schölkopf, & Zien, 2006](#)).

2.4.1.4 Reinforcement Learning

The interaction between an agent and its environment is at the centre of the Reinforcement Learning (RL) approach. The main goal of RL is to provide the agent with the ability to learn and decide intelligently by employing observations gained through interactions with the environment. These observations provide the agent with a foundation for taking action, and the reward system directs the agent in reducing risks and streamlining its decision-making process within the environment in which it is operating. ([Mohammed, Khan, & Bashier, 2016](#))

There are four main steps that an intelligent program (agent) needs to take ([Mohammed et al., 2016](#)):

- An input which represents the observations
- Based on the received input from the environment the decision needs to be taken by the agent
- A reward will be assigned to the agent's action based on the taken action

- The state of the action will be stored to use for future actions helping to minimize the penalty of the action

2.4.2 Machine Learning Algorithms in NLP

Natural language processing (NLP) is an area of study that includes machine learning approaches that use complex algorithms and methodologies to interpret and handle textual input. These techniques apply computational algorithms and statistical models to identify patterns and structures in natural language to extract and understand them. By benefiting from these techniques, sophisticated tools and applications including sentiment analysis and machine translation may be created, revolutionizing the way we communicate and comprehend human language. Computational algorithms, such as support vector machines, decision trees, and neural networks, are utilized to learn complex patterns and structures in the text. These algorithms can extract semantic meaning, syntactic relationships, and contextual information from the data. As a result, they enable the development of powerful NLP applications. In this section, the summary of these methods and techniques is investigated.

2.4.2.1 Text Representation in NLP

When trying to create an intelligent system that interacts and functions with natural language, we must employ an acceptable approach to translate the text into a collection of numbers that can accurately represent the text because computers can only comprehend numbers. Additionally, with the aid of these approaches, intelligent machines will be able to extract certain helpful features and data, such as the sentences and documents that are most crucial and vital. The next sections will go through a variety of techniques used in natural language processing for this purpose: Bag of words, Term Frequency, Inverse Document Frequency, and Word Embedding.

- **Bag of Words:** The Bag of Words (BOW) model serves multiple purposes, functioning as both a feature selection algorithm and a document classification tool. Furthermore, it represents a vector that captures the frequency of word occurrences within a document, often

referred to as a histogram (Qader, Ameen, & Ahmed, 2019). In general, this method is popular in feature engineering to convert textual information into a numerical representation, which has the following main steps:

- (1) **Tokenization:** Split the text into words
 - (2) **Counting:** Count the occurrences of each word in the document
 - (3) **Vector Representation:** represent the document as a vector where each element of the vector corresponds to the frequency of a specific word
- **Term Frequency - Inverse Document Frequency:** The technique leverages the number of times a word appears in a text as well as its frequency across all documents in a corpus to reflect a word's significance in a document. This technique is frequently used in the fields of text mining and information retrieval to assess the link between each word in a collection of documents. A valuable indicator of a word's relative relevance in a given context, the term frequency (TF) of TF-IDF measures how frequently a word appears in a document. This indicates that the words in papers with the highest TF values are the most significant. The inverse document frequency (IDF), in contrast, considers the word's frequency throughout the whole corpus. This suggests the frequency with which a certain term appears throughout the corpus of documents. High DF value words are not important because they are used often across all publications. Since IDF is the inverse of DF, it demonstrates that uncommon words in all manuscripts have a larger relevance (Kim & Gil, 2019). TF-IDF can be calculated using the following steps:
 - (1) **Term Frequency:** Calculating the number of times a word appears in the document, normalized by the total number of words in the document
 - (2) **Inverse Document Frequency:** Logarithm the total number of documents divided by the number of documents containing the word
 - (3) **TF-IDF:** Multiplying the TF and IDF value to measure the importance of each word
 - **Word Embedding:** Word embedding refers to a set of approaches used to quantify and represent the meanings of words. These approaches utilize vectors to represent the meaning of

a word within a vocabulary, taking into account both its frequency within a document and its frequency across a corpus. One approach, known as "one-hot" encoding, represents words as vectors with a "1" in a single position and zeroes elsewhere (Arseniev-Koehler & Foster, 2020). Since the created vector does not hold any semantic meaning, using other methods to represent the meaning and relationships of the words can be more useful and beneficial in the NLP tasks. Word embeddings possess the capability to capture human semantic representations by acquiring word meanings from extensive usage of language by humans, encompassing various sources such as news, books, web crawling, and audiovisual media like television and movie scripts. As a consequence, word embeddings serve as a highly valuable instrument for gaining insights into the intricacies of human language and enabling the exploration of sociolinguistic dimensions on a large scale (Hamilton, Leskovec, & Jurafsky, 2018).

There are two main approaches for creating embeddings which are count-based and Artificial Neural Networks based. A more recent count-based method today is the Global Vector. Moreover, ANN architecture learns word embedding from a given corpus such as word2vec and SkipGram methods (Arseniev-Koehler & Foster, 2020).

2.4.2.2 Multi-task Learning with Deep Neural Networks

Multi-task learning involves training machine learning models with data from multiple tasks at once, utilizing shared representations to capture common patterns among related tasks. These shared representations enhance data efficiency and may lead to quicker learning for connected or subsequent tasks, addressing challenges like the data volume and computational resources required by deep learning. Nevertheless, achieving these benefits has proven challenging and remains a current focus of research efforts (Crawshaw, 2020). The fundamental concept of multi-task learning lies in the belief that knowledge acquired from one task can be transferred to another, ultimately enhancing the overall learning process and generalization capabilities. By simultaneously training on multiple tasks, models can capture underlying patterns and relationships that might remain hidden when trained independently for each task. This approach not only enhances the predictive performance of individual tasks but also improves the efficiency of training, as the shared representations enable more effective utilization of available data.

Machine learning methods are fundamental to the field of natural language processing (NLP) as they play a pivotal role in facilitating the comprehension and efficient processing of human language by NLP models. Several prominent algorithms have emerged in the context of NLP, each offering unique characteristics and capabilities. These algorithms, which are listed below, are widely recognized and extensively utilized in various NLP applications and research endeavours.

2.4.2.3 Multi-layer Perceptron

The multi-layer perceptron (MLP) is a commonly used type of neural network where signals flow in one direction, from input to output, without loops. Hidden layers are not directly connected to the environment (Popescu, Balas, Perescu-Popescu, & Mastorakis, 2009). In MLP, input data is processed through each layer performing non-linear transformations on the data. The output of one layer is used as the input to the next layer, and the process continues until the final layer produces the network's output. However, due to architectural constraints, particularly in capturing long-range dependency in the data, the performance of this method is limited.

The main reason behind this limitation is its feed-forward architecture where the information flows only in one direction, from input to output and each layer only receives information from the previous layer (Popescu et al., 2009). Therefore, this algorithm does not support any loops or other forms of recurrence, which may be crucial for capturing long-range dependencies in the data such as natural text.

2.4.2.4 Recurrent Neural Network

Learning tasks involving sequential data are abundant in various domains, including image captioning, speech synthesis, music generation, and video game playing. Recurrent neural networks (RNNs), unlike feedforward networks, excel in capturing time dynamics and processing examples one by one, allowing them to retain a memory that encompasses a long context window. Recent advancements in network architectures, optimization techniques, and parallel computation have facilitated large-scale training of RNNs (Lipton, 2015). RNNs can retain memory with the help of recurrent edge which can introduce a notation of time to the model. As it is shown in figure 2.4, at each time step, nodes receive input and process it along with the hidden state from the previous

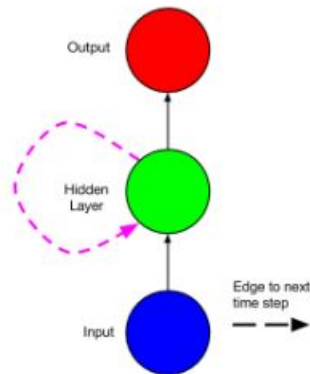


Figure 2.4: A simple RNN (Lipton, 2015)

time step. This hidden state serves as the memory of the model, capturing information about the past and updates based on the current input and the previous hidden state through a set of learnable parameters.

The architecture of Recurrent Neural Networks (RNNs) can be categorized into four main types, each serving specific purposes and applications:

- (1) **One-to-One:** This architecture involves a single input and a single output pair. It is commonly used in tasks such as image classification, where an image is provided as input, and the model generates a single output representing the class or category of the image.
- (2) **One-to-Many:** In one-to-many RNNs, there is one input and multiple outputs. An example of this architecture is image captioning, where the model takes an image as input and generates a sequence of words that describe the content of the image.
- (3) **Many-to-One:** The many-to-one architecture produces a single output based on multiple inputs. Sentiment analysis is an application that exemplifies this architecture, where the model receives a complete sentence as input and determines whether the sentiment expressed in the sentence is positive or negative.

- (4) **Many-to-Many:** The many-to-many architecture involves multiple inputs and produces multiple outputs. Machine translation is a primary application of this architecture, where the model takes a sequence of words in one language as input and generates a corresponding sequence of words in another language as output.

These different RNN architectures offer flexibility in handling various types of data and tasks, enabling the development of sophisticated models for tasks such as image processing, natural language understanding, and language translation.

RNNs have many features and advantages, however, they have some problems that make it difficult for researchers to choose them for their architectures. One of the main reasons is the difficulty in learning the long dependencies in the data. The primary challenge in capturing long-term dependencies within recurrent neural networks (RNNs) lies in the phenomenon known as the vanishing or exploding gradient problem. During the process of back-propagating errors over extended time intervals, the gradients encountered within the recurrent connections can exhibit exponential decay (vanishing gradients) or uncontrolled amplification (exploding gradients). As a consequence, RNNs encounter difficulties in accurately preserving and utilizing information across lengthy sequences, thereby constraining their capacity to effectively capture the inherent long-term dependencies within the data ([Hochreiter & Schmidhuber, 1997](#)).

2.4.2.5 Long Short-Term Memory

One solution that addresses the vanishing error problem is a method called long short-term memory (LSTM) where constant error carousels (CEC) are used which enforce a constant error flow within a special cell ([Hochreiter & Schmidhuber, 1997](#)).

Long Short-Term Memory (LSTM) is a specialized recurrent neural network (RNN) architecture that addresses the issue of capturing long-term dependencies in sequential data, a challenge that conventional RNNs often face. LSTM tackles this problem through the incorporation of memory cells and gating mechanisms. In LSTM there are four main gates which are mentioned as follows: ([Hochreiter & Schmidhuber, 1997](#))

- **Forget Gate:** The forget gate serves the purpose of selectively forgetting or preserving information stored in the memory cell. It is implemented as a sigmoidal unit that takes input from the concatenation of the previous hidden state and the current input. The forget gate applies a weighting to each component of the previous cell state, determining the extent to which it should be forgotten or retained. Through element-wise multiplication between the output of the forget gate and the previous cell state, the LSTM can dynamically control the retention of relevant information while discarding irrelevant or outdated information. This adaptive mechanism allows the LSTM to address the vanishing gradient problem by regulating the flow of information over time and maintaining the long-term dependencies necessary for effective sequential processing.
- **Input Gate:** The input gates, implemented as sigmoid threshold units with an activation function range of $[0, 1]$, assume the responsibility of regulating the transmission of signals from the network to the memory cell by appropriately scaling them. When the gate is closed, the activation tends to approach zero, thereby safeguarding the contents stored in the memory cell against disturbances stemming from irrelevant signals.
- **Output Gate:** The output gate plays a crucial role in determining the information that will be output from the memory cell. It is implemented as a sigmoidal unit that takes input from the concatenation of the previous hidden state and the current input. The output gate applies a weighting to the current cell state, controlling the extent to which it contributes to the final output. By element-wise multiplying the output of the output gate with the activation of the cell state, the LSTM selectively amplifies or attenuates the information that is passed on to the next hidden state and output. This gating mechanism allows the LSTM to regulate the flow of relevant information, enhancing the model's ability to capture and utilize essential features for accurate predictions or sequential processing tasks.
- **Cell State:** The cell state is a crucial component that enables the network to retain and carry information across long sequences. It acts as a memory storage unit that can selectively update and pass relevant information along the sequence. The cell state is modified through a combination of different gates, including the forget gate, input gate, and output gate. The cell

state is updated by first multiplying the previous cell state by the forget gate, which discards irrelevant information. Then, the input gate determines the new candidate values that could be added to the cell state. These candidate values are obtained by applying a hyperbolic tangent function to the input and the previous hidden state. The output gate, which regulates the amount of information to be exposed to subsequent layers or outputs, is then applied to the modified cell state.

2.4.2.6 Bidirectional LSTM

The Bidirectional Long Short-Term Memory (BLSTM) is proposed as a solution to overcome the limitations of a regular Recurrent Neural Network (RNN). The BLSTM incorporates both past and future input information in a specific time frame by splitting the state neurons into two parts: forward states and backward states. (Graves & Schmidhuber, 2005). In the structure of the BLSTM, forward states are responsible for processing input data in the positive time direction, while backward states handle input data in the negative time direction. The forward states and backward states are not connected to each other, resulting in a bidirectional structure. This structure allows the BLSTM to leverage information from both past and future time steps simultaneously, which is not possible in a regular unidirectional RNN (Schuster & Paliwal, 1997).

2.4.2.7 Gated Recurrent Units

Gated Recurrent Units (GRUs) are specialized memory elements used in recurrent neural networks (RNNs) for various applications. The Gated Recurrent Unit (GRU) is a simplification of LSTM that has gained popularity in computational neuroscience and machine learning communities due to its performance in various tasks, such as speech, music, video, and extracting nonlinear dynamics from neural data. However, empirical findings suggest that GRU networks may struggle with certain tasks like unbounded counting compared to LSTM networks (Jordan, Sokol, & Park, 2021). This neural network has the following features:

- GRUs are a simplified alternative to LSTMs that can achieve similar performance with fewer parameters

- GRUs use gating mechanisms to selectively update and reset the hidden state, allowing them to capture long-term dependencies in the input sequence
- GRUs have two gates (reset and update) and do not have a separate memory cell like LSTMs
- GRUs have a simpler architecture than LSTMs, which can make them faster to train and less prone to overfitting

2.4.2.8 Transformers

The "Attention Is All You Need" paper ([Vaswani et al., 2017](#)), a seminal work in natural language processing (NLP), introduced the Transformer model, which has significantly impacted the field. Motivated by the limitations of recurrent and convolutional neural networks in capturing long-range dependencies, the paper proposed the Transformer as an alternative architecture. The Transformer model revolutionized NLP by introducing the self-attention mechanism, which enables the model to focus on different parts of the input sequence. This attention mechanism facilitates efficient information integration and contextual relationship capturing. Unlike sequential models, Transformers allow for parallel computation, making them easier to train and faster in processing sequences. The advantages of Transformers over traditional models like recurrent neural networks (RNNs) include their ability to capture long-range dependencies, handle variable-length input sequences, and achieve state-of-the-art performance in various NLP tasks. Applications of Transformers span machine translation, sentiment analysis, question answering, and text generation, where they have demonstrated remarkable success. While Transformers have greatly impacted NLP, they also present challenges. Higher computational requirements and potential difficulties in capturing hierarchical structures are important considerations. Nonetheless, ongoing research continues to address these limitations and explore new directions for Transformers, such as investigating different attention mechanisms and architectural modifications for specific tasks or improved performance.

Table 2.2: Summary of Machine learning algorithms

Algorithms	Type	Description
Multi-layer Perceptron	Feedforward Neural Network	A basic type of artificial neural network, consisting of multiple layers of nodes.
Recurrent Neural Network	Sequential Model	Processes sequences of data, retain information through hidden states, suitable for sequential data.
Long Short-Term Memory	RNN Variant	A specialized RNN with memory cells, is useful for handling long-range dependencies in sequential data.
Bidirectional LSTM	RNN Variant	An LSTM variant that processes input data in both forward and backward directions for improved context.
Gated LSTM	RNN Variant	Utilizes gating mechanisms like GRU, providing similar advantages in terms of learning longer sequences.
Transformers	Attention Mechanism	Processes sequences as a whole using self-attention. excel in capturing global relationships in data

Chapter 3

Methodology

3.1 Introduction

Chapter 3 provides a comprehensive overview of the research approach and techniques employed in the study. The first part of the methodology focuses on data collection and data preprocessing procedures 3.2.1. This involves gathering relevant data from reliable sources and applying preprocessing techniques to ensure data quality and suitability for subsequent analysis. Subsequently, the methodology delves into the architecture of the model 3.2.2, providing a comprehensive overview of its structure and components. Finally, the methodology section concludes with the presentation of the results obtained from the training process. The section 3.2.3 will showcase the model's performance, accuracy metrics, and any other relevant evaluation criteria used to assess the model's effectiveness.

3.2 Research Methodology

The methodology section outlines the systematic approach employed in this study to investigate the mentioned research objectives by explaining the procedures, techniques, and tools utilized to gather, analyze and interpret the data. The methodology encompasses various aspects, including methods for data collection and processing, as well as a comprehensive explanation of the designed architecture.

3.2.1 Data Collection

The data collection section outlines the procedures and techniques employed to gather the necessary data for this study including the ROMA software, data collection, and any specific tools and procedures. By ensuring a robust and well-structured data collection process, this study aims to obtain high-quality and reliable data that aligns with the research objectives. The data collection phase involves planning, selection of appropriate sources, and implementation of standardized methods to gather the required data points.

3.2.1.1 ROMA

The ROMA software is a specialized tool developed for the purpose of generating ROM diagrams based on the given sentences. These diagrams provide visual representations of the structural components and relationships within the sentences, aiding in their analysis and understanding. The ROMA software utilizes advanced natural language processing algorithms and techniques to parse the sentences and identify key linguistic elements such as nouns, verbs, adjectives, and their respective dependencies. By leveraging this linguistic analysis, the software constructs comprehensive ROM diagrams that depict the syntactic structure of each sentence. The ROMA software offers an intuitive user interface, allowing users to input sentences and generate corresponding ROM diagrams with ease.

To ensure the reliability and accuracy of the ROMA software, extensive testing procedures were employed in this research, given its rule-based nature. The objective of these testing procedures was to evaluate the software's performance and enhance the precision of the generated ROM diagrams. As a result of these rigorous evaluations, the latest iteration of the ROMA software demonstrates an enhanced capacity to generate ROM diagrams for a wide range of sentence structures, including both the fundamental patterns elucidated in (Zeng, 2008) and more intricate linguistic constructions. Moreover, dedicated software has been developed to facilitate the storage and manipulation of ROM diagrams. This software empowers users to input textual data, generate visually appealing and user-intuitive ROM diagrams that correspond to the provided sentences, display the output in a tabular

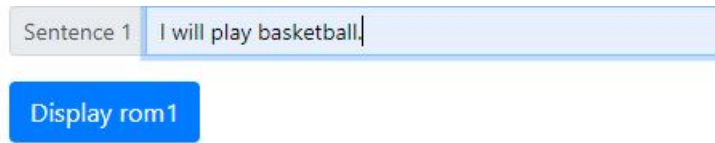


Figure 3.1: An illustration of inputting a sentence in the ROMWeb software.

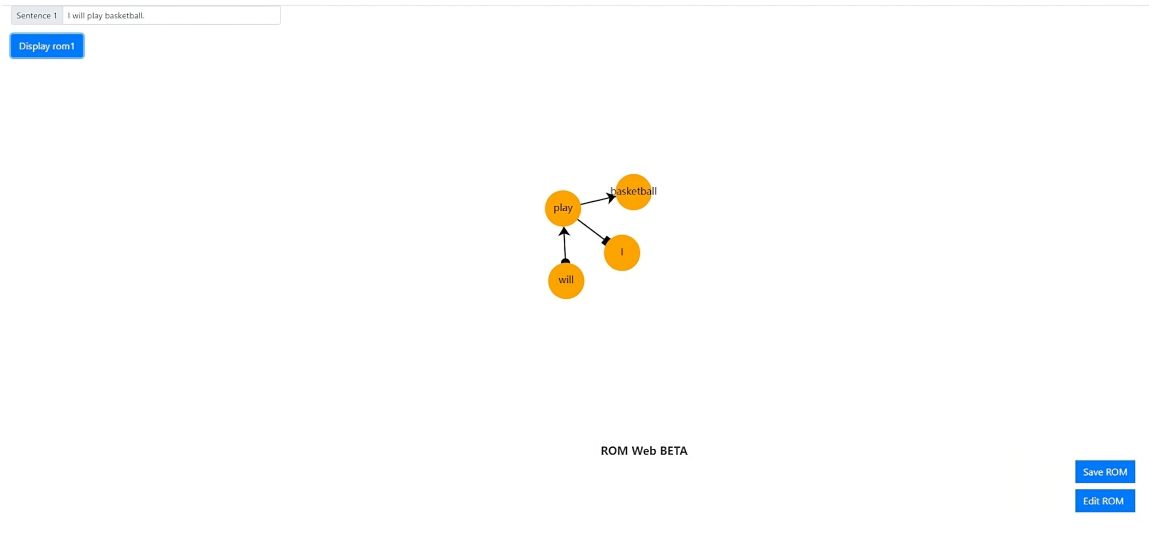


Figure 3.2: An illustration of result the ROM diagram in the ROMWeb software.

format, modify the diagrams by adding or removing relationships, and save the resulting ROM diagrams in a database. By offering these functionalities, the software enhances the usability and versatility of ROM diagram management for efficient analysis and manipulation of linguistic structures.

3.2.1.2 Data Collection Tool

The ROMWeb software has been developed as a tool for visualizing the results of ROM analysis and facilitating any necessary modifications to the generated output. This section serves to provide an overview of the software's functionality and outline the data collection procedures employed in this research endeavour. By presenting the capabilities of ROMWeb and detailing the methodologies employed for data acquisition, this section aims to enhance the understanding of the software's utility and its role in the research methodology.

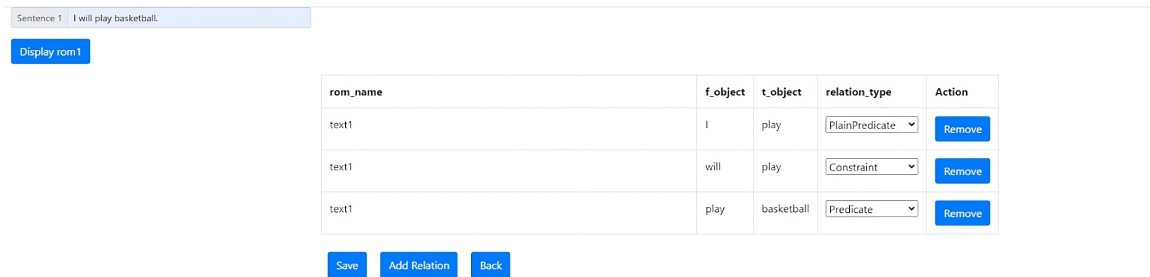


Figure 3.3: An illustration of Editing the result of ROM diagram in the ROMWeb software.

Software Description

First, the ROMWeb software facilitates the generation of ROM diagrams by accepting a sentence as input, as depicted in figure 3.1. The input functionality is located on the left side of the main layout. Subsequently, upon pressing the Display button, the software generates and presents the corresponding ROM diagram utilizing graphs and nodes, as illustrated in picture 3.2. Users have the option to save the generated result if the diagram is deemed satisfactory or initiate modifications by utilizing the Save or Edit buttons, respectively. This user-friendly interface empowers researchers to efficiently interact with and manipulate the generated ROM diagrams to meet their specific requirements. On the other hand, if the user would like to edit the result, the user can see the result in a table as mentioned in figure 3.3. In this situation, the user is able to take the following actions:

- Remove the relation by hitting the Remove Button
- Add a new Relation
- Change the content of the table by double-clicking on the cells
- Change the relation type from the drop-down menu in the "relation type" column
- Save the edited ROM result

All these functions are implemented using JavaScript and Python programming languages. JavaScript is a high-level, dynamic, and interpreted programming language that is primarily used for creating

Patterns	Examples
Subject + Verb + Object	I drink Coffee.
Subject + Verb + "a" + Object	They make a cake.
Subject + Verb + "the" + Object	She walks the dog.
"The" + Subject + Verb + Object	The truck carries goods.
"The" + Subject + Verb + "a or an" + Object	The cat catches a mouse.
"The" + Subject + Verb + "the" + Object	The teacher explains the concept.

Table 3.1: The Predefined patterns for collected sentences

dynamic and interactive web pages. In addition, Python is a high-level, interpreted programming language that is known for being easy to learn and readable. It may be used for a variety of purposes, including developing websites, analyzing data, developing artificial intelligence, and more.

Data Collection procedures

This section is a brief overview of the systematic steps and processes followed to gather data for this research study. Data for this research was primarily collected from ChatGPT, a powerful language model, that provided a diverse range of sentences that served as valuable input for the study.

The selection of data followed specific criteria based on sentence patterns. The initial focus was on basic sentence structures, such as "subject + verb + object". The whole list of patterns which was used in this study is shown in the table 3.1. These patterns formed the foundation for data collection and allowed for the systematic exploration of sentence variations.

As mentioned in the previous section, a web application called ROMWeb was developed to facilitate the data collection process. This tool enabled researchers to input sentences and generate corresponding ROM diagrams. The ROMWeb software played a crucial role in automating the generation of visual representations for the sentences, which facilitated data analysis and comprehension. In addition, the collected data needs to be stored and managed in a proper manner. MongoDB, a widely-used database system, was developed to provide a robust solution for storing and organizing the result of ROM diagrams and associated metadata.

3.2.1.3 Data Preparation

The data preparation phase plays a crucial role in machine learning projects, as it involves transforming raw data into a suitable format for analysis. In this section, we present the data preparation procedures undertaken for our project, focusing on loading and processing the data, generating word pairs, converting relationships to numerical labels, incorporating part-of-speech tags, calculating pair distances, and encoding categorical data. These steps ensure that the data set is appropriately prepared for subsequent analysis and model training.

Data Loading and Transformation

To initiate the crucial phase of data preparation, the raw data was acquired from a MongoDB database, where it was stored in JSON format. This initial step involved extracting the data and subsequently transforming it into a structured representation that is amenable to analysis. The Python programming language, in conjunction with the powerful Pandas library, was employed to convert the JSON data into a tabular data structure known as a data frame. This transformation not only facilitated seamless manipulation and exploration of the data set but also provided a foundation for subsequent pre-processing and feature engineering tasks. By leveraging the versatility and efficiency of the Pandas library, we were able to handle the data with ease and extract meaningful insights from it, thus setting the stage for the subsequent stages of data preparation and model development.

Word Pair Generation

To capture and uncover the inherent relationships embedded within sentences, a systematic and rule-based approach was employed in the generation of word pairs. Through a meticulous analysis of each sentence within the data set, pairs of words were identified and extracted based on a set of predefined rules. By way of illustration, consider the sentence "I play basketball." In this instance, the word pair "I" and "play" was extracted and recorded. By adopting this approach, we aimed to discern and capture potential pairs of words that might possess a discernible relationship with one another. This methodological framework facilitated the identification of candidate pairs, thereby

setting the stage for further analysis and subsequent utilization in the later stages of the research endeavour.

Converting Labels to Numbers

The relationships between word pairs, constituting the fundamental labels within our data set, necessitated a transformation from their original linguistic form to numerical representations. To accomplish this, a systematic mapping procedure was employed, wherein distinct numerical values were assigned to each relationship type. This encoding process served to translate the qualitative nature of the relationships into a quantitative format, thereby facilitating subsequent analysis and modeling tasks that often rely on numerical labels. By converting the relationships into numerical representations, we established a foundation for further computational analysis, enabling the application of various machine learning techniques and algorithms to effectively uncover patterns and dependencies within the data set.

Part-of-Speech Feature Incorporation

In order to enhance the linguistic information embedded within the data set, an essential step involved the calculation of part-of-speech (POS) tags for each individual word. POS tags serve as valuable linguistic indicators that reveal the grammatical attributes and syntactic functions of words, such as their roles as nouns, verbs, adjectives, or adverbs. By employing advanced natural language processing techniques, we systematically assigned appropriate POS tags to each word within the data set. This process bestowed an enriched representation of the data, augmenting its informational depth and facilitating subsequent analysis. By incorporating POS tags as additional features, we not only enhanced the quality of the data set but also provided valuable linguistic context, enabling a more comprehensive exploration of the interrelationships and patterns within the textual data.

Pair Distance Calculation

In order to capture the spatial arrangement and contextual cues within sentences, a calculation of the distance between word pairs was conducted. This distance metric served as a quantitative

measure, indicating the number of intervening words between each pair within a sentence. By quantifying the spatial separation or proximity of words, the pair distance information offered valuable insights into the structural organization and contextual coherence within the sentence. This additional feature facilitated a more comprehensive analysis by encompassing the relative positioning of words, potentially revealing meaningful patterns and capturing pertinent contextual information pertaining to the relationships between words.

Categorical Data Encoding

In order to accommodate the requirements of machine learning models, we undertook a crucial step of encoding categorical data into feature-based vectors. This transformation involved converting categorical variables, such as word types, into numerical representations. By mapping categorical data to numerical values, we facilitated the effective processing and interpretation of these variables by machine learning algorithms. This encoding process not only ensured the compatibility of categorical data with numerical-based models but also preserved the essential information contained within these variables, thereby enabling accurate and meaningful analysis and prediction tasks.

In this section, we have detailed the comprehensive data preparation procedures undertaken for our machine learning project. By loading and transforming the data, generating word pairs, converting relationships to numerical labels, incorporating part-of-speech features, calculating pair distances, and encoding categorical data, we ensured that the data set was suitably prepared for subsequent analysis and modelling. These steps provide a solid foundation for robust and accurate machine learning experimentation, paving the way for insightful findings and reliable model performance.

3.2.2 Proposed Model Architecture

The development of the proposed machine learning architecture is driven by the goal of predicting relationships between words within a given sentence. As highlighted in previous sections, the research narrows its focus to specific relationships, namely None, Constraint, Predicate, and Plain Predicate, in order to lay a strong foundation for the initial phase. The selection of these

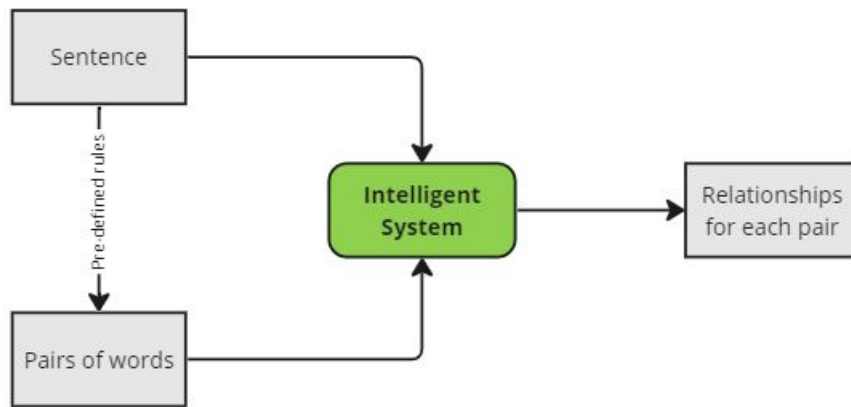


Figure 3.4: Overview of the system.

relationships is aligned with the objective of enhancing the representation of natural language and providing valuable insights to designers engaged in the Environment-Based Design process. By accurately identifying and generating these relationships, the machine learning model aims to equip designers with a comprehensive system that enables a deeper understanding and analysis of natural language, facilitating informed decision-making and fostering effective design practices.

3.2.2.1 Model Overview

The designed machine learning-based system has several components that work together to achieve its objective. This system operates by processing pairs within each sentence individually, taking into account both the specific pair and the sentence that contains it. This approach allows for a focused analysis of the relationships between words. The inputs to the model consist of a pair, representing two words within a sentence, along with the corresponding sentence itself as shown in the figure 3.4.

The proposed model, as shown in figure 3.5, leverages the strength of multiple models to improve overall performance. It incorporates several key components to handle the inputs effectively. Firstly, one of the models is responsible for predicting whether two words in a given pair have a relationship with each other. This component aims to identify and classify the presence or absence of a relationship between the pair. Additionally, another model is specifically designed to capture dependencies

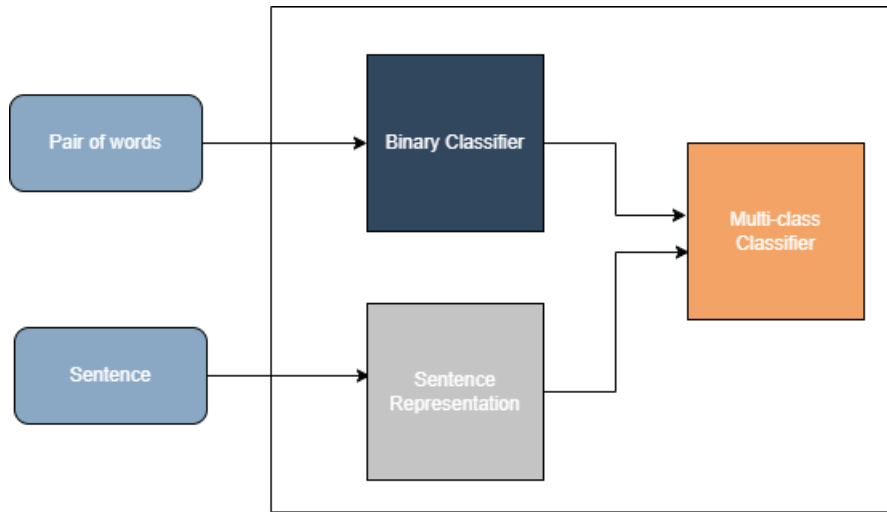


Figure 3.5: Overview of the architecture of the designed system.

within the sentence. This part focuses on tracking and understanding the intricate connections between words within the sentence structure and contributes valuable information about the overall context and dependencies within the sentence. To arrive at a final prediction, the results from these two models are concatenated, effectively, combining their outputs. This concatenated information is then used for the subsequent classification task, where the relationships between words are determined. By employing this approach, the model can leverage the complementary capabilities of the individual components, enhancing the accuracy and robustness of the predictions.

In summary, the proposed model employs the ensemble learning approach to predict relationships between words. It utilizes separate models to handle pair-level relationship classification and sentence-level dependency tracking.

3.2.2.2 Structure of Input and Output

This section will provide a detailed understanding of the inputs and outputs of the designed model. As mentioned in the previous section, the model's inputs consist of two main components: the pair, and the sentence. The pair component is enriched with additional features to provide a more comprehensive representation. Along with the words themselves, the input includes the Part-of-Speech (POS) tags corresponding to each word in the pair. These POS tags offer valuable information about the grammatical roles and syntactic categories of the words. Furthermore, the

distance between the two words in the pair is incorporated as an additional feature. By considering these additional aspects, the input for the binary classifier model encompasses the words, their corresponding POS tags, and the distance between them. This enriched input enables the model to capture better contextual information related to the relationship between the words in the pair. On the other hand, the sentence component serves as a contextual input for the model. At this stage, no additional information is added to the sentence input, and it is considered a standalone element. For the initial version of the model, the sentence is limited to a length of 10 words.

Upon analyzing the inputs, the output of the designed model is an array with four nodes, each corresponding to one of the four possible classes. These classes represent the different types of relationships that the model aims to predict. The output array provides a probability distribution over the four classes, indicating the model's confidence in assigning each relationship type to the given pair. This output schema allows for a clear and interpretable representation of the model's predictions, facilitating further analysis based on the predicted relationship classes.

In summary, the inputs to the designed model encompass the "pair" and "sentence" components. The pair input includes the words, their corresponding POS tags, and the distance between them, enhancing the representation of the relationship between the words. The sentence input serves as contextual information without any additional modifications at this stage. The model's output is an array with four nodes, representing the probabilities of the four possible relationship classes. This comprehensive input-output framework provides a foundation for the model's prediction capabilities and supports the analysis and interpretation of the predicted relationship types.

3.2.2.3 Model Components

The proposed machine learning model, as mentioned in figure 3.6, comprises multiple components carefully designed to process and analyze input data, thereby facilitating the understanding and interpretation of natural language. The following is a detailed explanation of these models' components and their respective functionalities:

MLP Model

The first model which is a simple multi-layer perceptron has sections as follows:

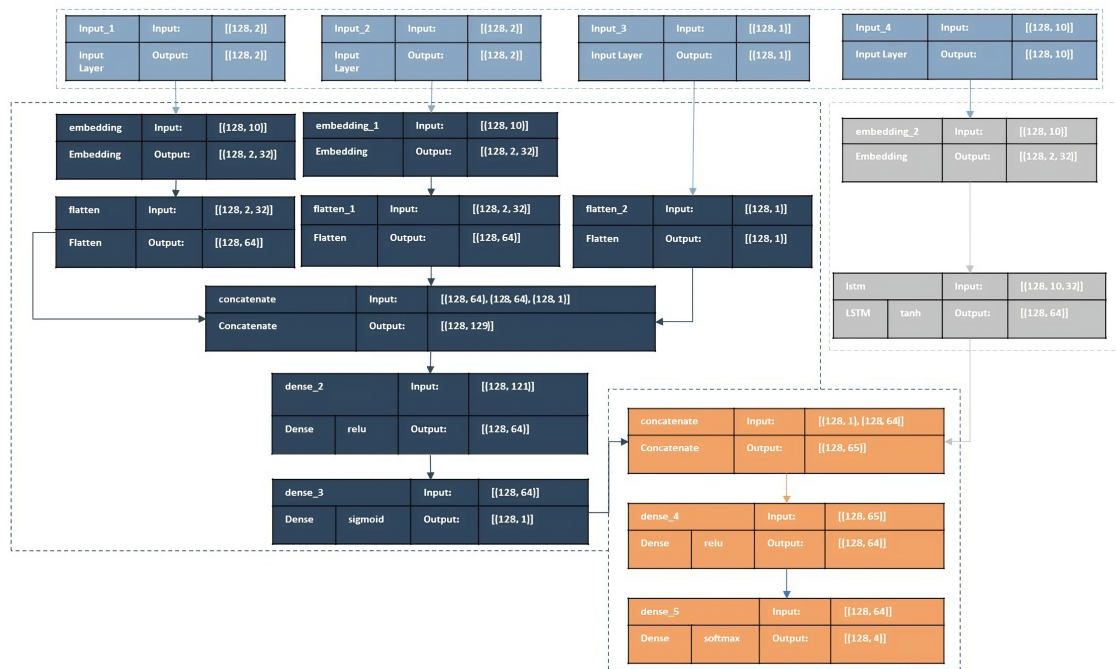


Figure 3.6: Overview of the designed model's components.

- **Inputs:** The binary classifier model, which is an MLP model, accepts three inputs, representing word pairs, part-of-speech (POS) tags associated with the words, and the distance between the words within a pair. These inputs refer to the data or information provided to the binary classification model for prediction.
- **Embedding Layers:** To convert the input data into dense numerical representations, the model employs embedding layers. In addition, separate embedding layers are used for word pairs and POS tags, while the distance input remains unchanged.
- **Flatten:** The embedding layer maps an input, such as words or tokens, to dense vector representations which result in a 3-dimensional tensor where each word in the input pair has an associated dense vector representation. Therefore, the embedded inputs are flattened to transform them into one-dimensional representations, facilitating the subsequent processing of data. In addition, this step is helpful in keeping the input shape constant which can be easily concatenated. Lastly, since MLP layers expect a 2-dimensional input layer, the flattened tensors preserve the individual feature values while arranging them in a way that aligns with

the MLP architecture, ensuring that the concatenated inputs can be processed by the MLP layers to learn complex patterns and relationships in the data.

- **Concatenation:** The flattened embeddings from word pairs, POS tags, and distance are concatenated into a single feature vector. This amalgamation of information enables the model to capture the intricacies and dependencies among these components.
- **MLP layer:** The concatenated embedding undergoes a fully connected layer, one hidden layer with ReLU activation function to introduce non-linearity and 64 neurons. Through these layers, the model learns complex relationships and patterns inherent in the data.
- **Output:** The output layer of the MLP model consists of a solitary node employing a sigmoid activation function. This configuration produces a binary output, indicating the presence or absence of a relationship between the words within a pair. The sigmoid function is commonly used in binary classification problems where the output is a probability between 0 and 1. If the output value is above a certain threshold, the model predicts the positive class; otherwise, it predicts the negative class.

LSTM Model

In the proposed architecture, Long-Short Term memory has been used to capture the dependencies and track information in the sentences. This model contains the following parts:

- **Input:** The LSTM model accepts a sequence of words as an input.
- **Embedding layer:** An embedding layer is employed to map each word in the input sequence to a dense vector representation. This step enables the model to capture and encode semantic information associated with individual words. In this study for the first version, both models have utilized the Embedding layer in Keras Library for implementing this purpose. The embedding layer in Keras inherently considers the contextual information of words within a sentence as well as taking into account the neighbouring words and their respective positions.
- **LSTM Layers:** The embedded sequence is then processed through an LSTM layer, which has 64 neurons, which leverages its recurrent nature to model temporal dependencies and

acquire contextual representations of the words. By considering the sequence as a whole, the LSTM layer captures relationships and dependencies among the words in the sentence.

Last MLP Model

For connecting the two mentioned models, the outputs of the models are concatenated with each other. This model has the following sections:

- **Input:** The model takes as inputs the outputs of both the MLP and LSTM models. These outputs encapsulate the learned representations and relationships from the respective components.
- **Concatenation:** The outputs of the MLP and LSTM models are concatenated to form a consolidated feature vector. This step integrates the complementary information captured by the individual models, enabling a more comprehensive analysis.
- **MLP Layers:** The concatenated features undergo further processing through additional MLP layers. This stage allows the model to extract higher-level relationships and patterns based on the combined information.
- **Output:** The final output layer of the model comprises multiple nodes, equal to the number of classes to be predicted. The activation function employed is softmax, which produces a probability distribution over the classes, facilitating multi-class predictions. The ensemble model is subsequently compiled using appropriate techniques such as the sparse categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric. This configuration ensures the effective training and evaluation of the model's performance.

In summary, the proposed model combines the strengths of the MLP and LSTM components to capture and analyze the relationships within word pairs and sentence sequences, respectively. By leveraging the abilities of these components, the model aims to provide a comprehensive representation of the intricate relationships between words in sentences. This holistic representation enables accurate predictions and paves the way for further analysis and exploration of natural language understanding tasks.

3.2.2.4 Training Process

The training procedure for the machine learning model involves the optimization of its parameters to minimize the selected loss function. In this study, the loss function employed is sparse categorical cross-entropy, which is widely used in multi-class classification scenarios where the class labels are integers. The objective of the training is to reduce the discrepancy between the predicted outputs and the actual labels, enabling the model to accurately classify input data into their respective classes.

To optimize the model parameters, the Adam optimizer is utilized, which is a popular choice for training deep neural networks. During the training process, the model learns and adopts the training data by iteratively updating its parameters to minimize the chosen loss function. The training data is divided into batches, and the model computes the loss between the predicted outputs and the ground truth labels. Subsequently, the gradients of the loss with respect to the model parameters are calculated through back-propagation, allowing for the determination of the appropriate direction to adjust the parameters. This iterative optimization process incrementally enhances the model's performance by iteratively updating the parameters in a manner that minimizes the loss. The proposed model, trained on training data, accepts a pair of words and sentences as inputs and produces the final output. The training process involves providing the training data to the model and continuously updating the model parameters using the Adam optimizer. In addition, the designed model has been trained based on the following conditions:

- The model has trained on a data set that has around 1500 sentences following the predefined structures which are demonstrated in table 3.1.
- Cross-validation, which is a widely used technique in machine learning for model evaluation and hyperparameter tuning, has been used in the training stage. The main purpose of this technique is to assess the model's performance and evaluate its generalization capabilities. It provides a more reliable estimate of how well the model will perform on unseen data by simulating the process of training and testing on multiple independent data sets. In this study, 10-fold cross-validation is employed where the data set is divided into 10 equal-sized subsets or folds.

- The training and evaluation process is repeated 10 times, with each iteration using 90 percent of the data as training and the remaining 10 percent as validation.
- In the training process, an epoch refers to a complete pass through the entire training data set during the optimization process which represents one iteration in which the model sees and processes all the training examples to update its parameters. In our case, a value of 30 epochs was chosen for training
- In the context of training a machine learning model, a batch size refers to the number of training examples utilized in each iteration of the optimization algorithm. It represents the number of samples processed together before the model's parameters are updated based on the computed gradients. In our case, a batch size of 128 samples was chosen for training. This value represents the number of training examples processed together in each iteration.
- In training a machine learning model, callbacks are objects or functions that are executed at specific points during the training process. In this study, ModelCheckpoint has been used allowing for the creation of checkpoints to capture the best-performing model based on a chosen metric. Early Stopping has been used in this training process. It is a technique used during model training to automatically stop the training process if the model's performance on a validation set fails to improve.

Throughout the training process, the model's performance is evaluated using the specified metrics, namely "accuracy" in this case. The accuracy metric measures the model's ability to correctly classify the training examples and serves as an indicator of its performance and progress.

3.2.3 Training Results

This section will present the evaluation and performance of the proposed model after undergoing the training process. For assessing the performance and effectiveness of the model, the model's accuracy, recall, precision, and F1 score have been analyzed on the validation data. These metrics provide valuable insights into the model's overall correctness.

Before looking at the explanations for the metrics, it is beneficial for some basic terms to be defined properly. These terms are described in table [3.2](#) and are mentioned as follows:

Table 3.2: Confusion Matrix

Confusion Matrix		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

- **True Positive (TP):** A true positive occurs when the model correctly predicts a positive instance as positive. This means the model correctly identifies a positive class.
- **True Negative (TN):** A true negative occurs when the model correctly predicts a negative instance as negative. This means the model correctly identifies a negative class.
- **False Positive (FP):** A false positive occurs when the model incorrectly predicts a negative instance as positive. This means the model falsely identifies a negative class as positive.
- **False Negative (FN):** A false negative occurs when the model incorrectly predicts a Negative instance. This means the model incorrectly identifies a positive class as negative.

3.2.3.1 Evaluation Metrics

After reviewing the basic terms, a brief explanation of accuracy, recall, precision, and F1-score will be explained.

- **Accuracy:** The overall efficacy of the model’s forecasts is measured by accuracy. It displays the percentage of the data set’s total number of instances that were properly identified including both true positives and true negatives. In terms of overall correctness, a model that performs better is one with a greater accuracy value.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \quad (1)$$

- **Recall:** Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify positive classes. It represents the proportion of true positives correctly identified by the model out of all actual positive instances in the data set. A higher

recall value indicates that the model is better at capturing positive instances.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2)$$

- **Precision:** Precision measures the accuracy of the positive predictions made by the model. It represents the proportion of true positives correctly identified by the model out of all positive predictions made by the model. Precision focuses on the quality of the positive predictions and is useful in situations where false positives are costly. A higher precision value indicates that the model is better at making accurate positive predictions.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3)$$

- **F1-score:** The F1-score is a balanced metric that combines both precision and recall. It is the harmonic mean of precision and recall and provides a single measure that balances both metrics. The F1-score is particularly useful when you want to find a balance between precision and recall, especially in imbalanced data sets where one class dominates over the other. A higher F1 score indicates a better balance between precision and recall.

$$F1 - score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad (4)$$

3.2.3.2 Training Results

In this section, we present the results obtained from training the proposed model and conducting a comprehensive validation process. The performance of the model is evaluated based on various metrics, including accuracy, loss, recall, precision, and F1-score. Additionally, we provide visualizations of the validation data, classification reports, and selected samples to gain insights into the model's performance. As discussed, for training the proposed model, the epochs and batch size are considered as 30, and 128 respectively. The statement "batch size 128 in 30 epochs," means that during the training process, the model will process 128 training examples together in each iteration.

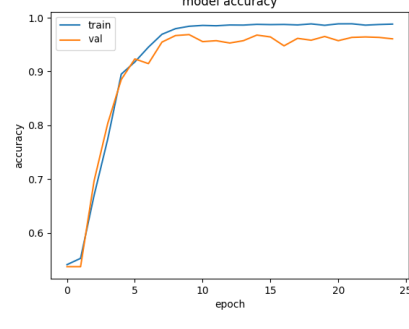
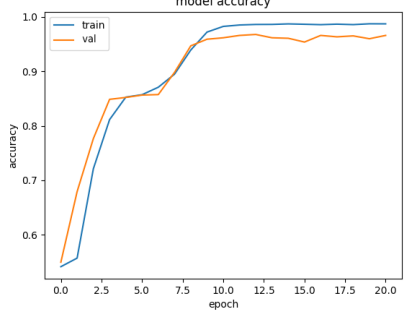
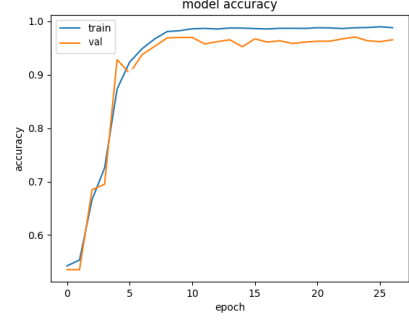
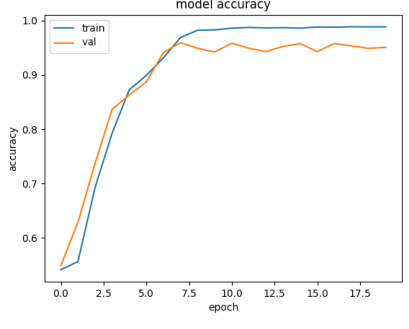
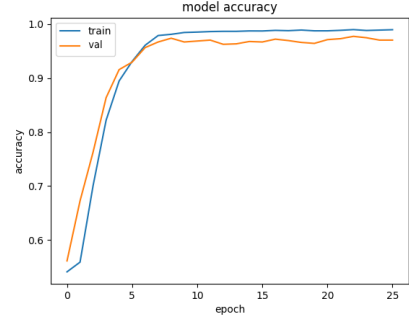
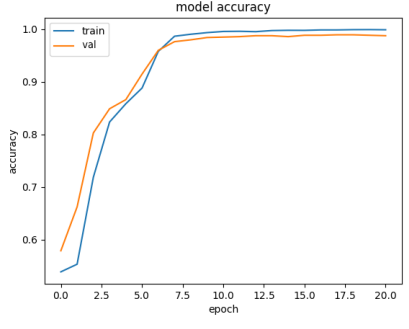
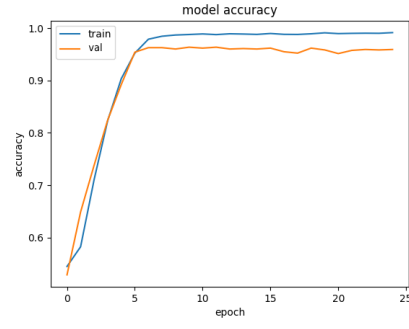
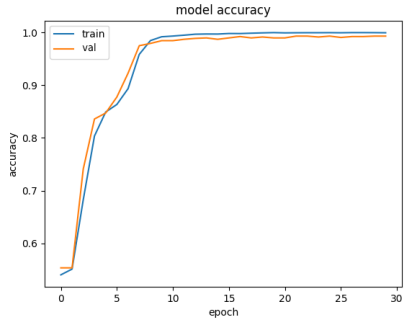
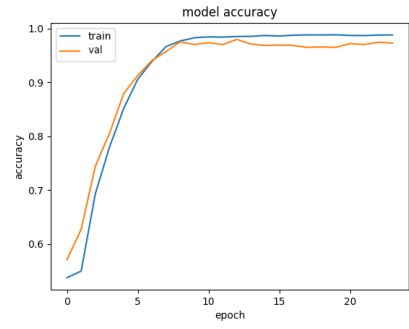
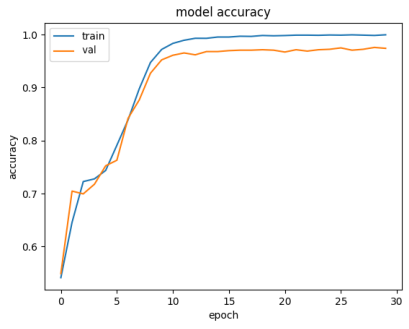


Figure 3.7: Accuracy of the proposed model in 10 folds

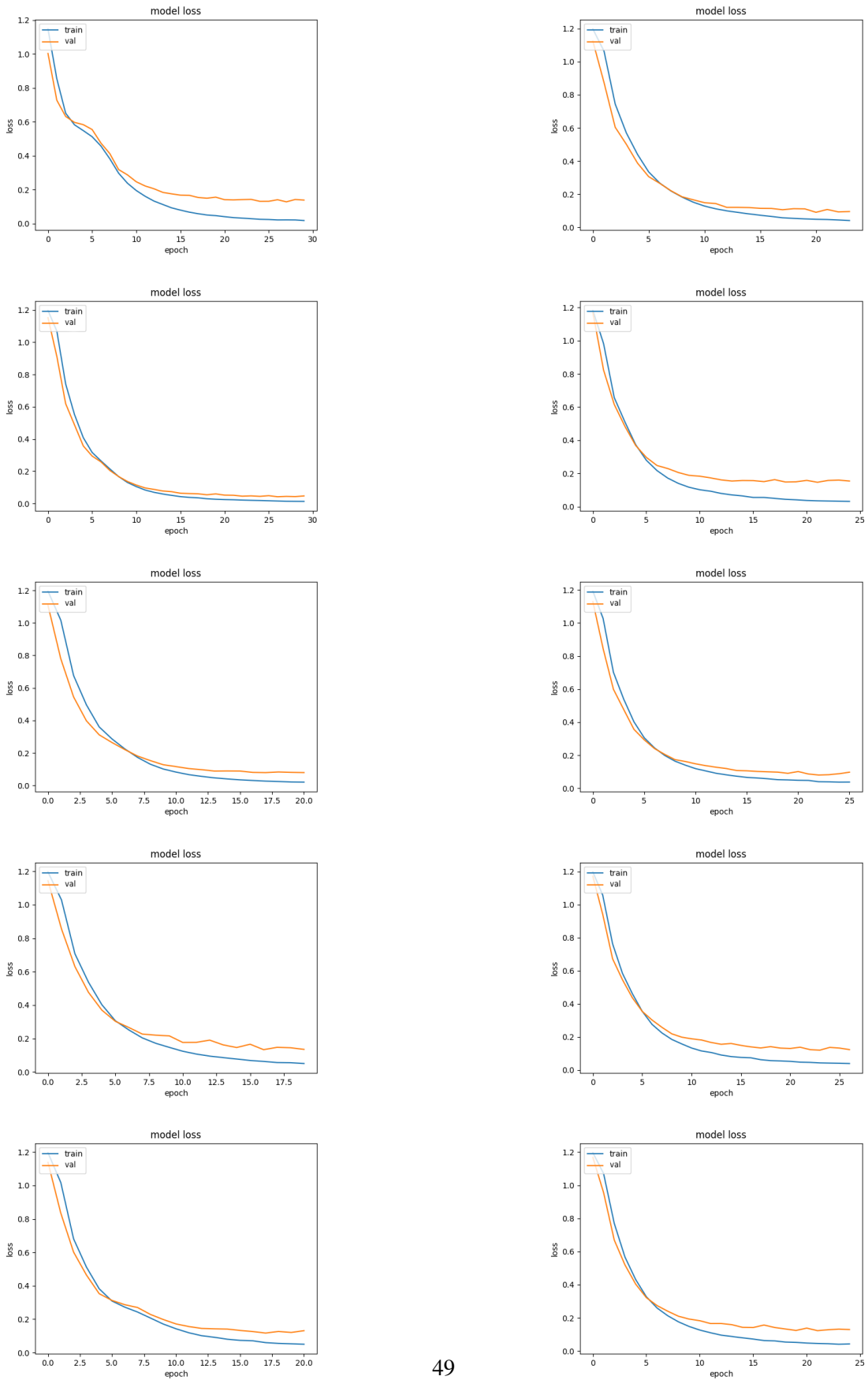


Figure 3.8: Loss of the proposed model in 10 folds

This process will be repeated for a total of 30 epochs. In each epoch, the training data set will be divided into batches of 128, and the model's weights will be updated based on the predictions and losses computed for each batch.

To validate the training in the proposed model, we employed 10-fold cross-validation, a widely-used technique for robust validation. The validation process involved assessing the accuracy and loss of the validation data for all the folds. The figures 3.7, 3.8 presented in this section illustrate the trends of accuracy and loss over time for each fold. Notably, the loss consistently decreases toward zero, indicating that the model successfully learns from the training data. Simultaneously, the accuracy steadily increases toward 1, demonstrating the model's ability to make accurate predictions.

The classification report provides a comprehensive analysis of the validation data, specifically for the best fold. It showcases metrics such as recall, precision, F1-score, and accuracy for each label predicted by the proposed model.

To further illustrate the performance of the proposed model, we randomly selected 30 samples from the validation data as it is shown in table 3.3. These samples serve as concrete examples of the model's predictions after training. By analyzing these instances, we can observe the model's ability to correctly classify various inputs and assess its overall performance.

In addition, the evaluation of the trained model on the validation data for the best fold is presented in the classification report, as indicated in table 3.4. The report provides insights into the model's performance by measuring various metrics. Notably, the achieved values for most of the metrics are high, indicating a satisfactory level of performance.

Overall, the results obtained from the proposed model demonstrate its effectiveness and strong predictive capabilities. The validation process, including the visualizations, classification reports, and selected samples, provides comprehensive insights into the model's performance. The model consistently achieves high accuracy, low loss, acceptable precision, recall, and F1-scores. These findings validate the robustness and reliability of the proposed model in capturing patterns and making accurate predictions.

Regularization techniques serve a critical role in the realm of machine learning, particularly when the performance of a training model demonstrates high accuracy. The need for regularization arises from the intention to mitigate the risks associated with overfitting, ensuring that the model's

Table 3.3: Comparing predicted and actual relationship on validation data after training the proposed model

No	Sentence	Pairs	Predicted relation	Actual relation
0	the politicians make a speech	make a	None relation	None relation
1	i grow a garden	i grow	PlainPredicate	PlainPredicate
2	we watch movie	we watch	PlainPredicate	PlainPredicate
3	the rain brings life	the rain	Constraint	Constraint
4	the teachers helped a student	the teachers	Constraint	Constraint
5	he studies biology in college	studies college	None relation	None relation
6	the scientist discovered a theory	the discovered	None relation	None relation
7	you walk the dog	you walk	PlainPredicate	PlainPredicate
8	the product met requirements	product met	PlainPredicate	PlainPredicate
9	the scientists discovered a cure	scientists a	None relation	None relation
10	the rivers supplied the water to the crops	supplied water	Predicate	Predicate
11	she plays video games	video games	Constraint	Constraint
12	we studied biology in college	in studied	Constraint	Constraint
13	the chef cooked a delicious meal	the a	None relation	None relation
14	the dancers performed the routine	performed the	None relation	None relation
15	the artist creates the sculpture	the sculpture	Constraint	Constraint
16	the bird sings a beautiful song	sings a	None relation	None relation
17	i take a class	i take	PlainPredicate	PlainPredicate
18	the musicians compose a piece	compose a	None relation	None relation
19	she follows the instructions	follows instructions	Predicate	Predicate
20	i swam laps	swam laps	Predicate	Predicate
21	he takes photos	takes photos	Predicate	Predicate
22	he catches a fish	he fish	None relation	None relation
23	the ships transported goods across the sea	goods the	None relation	None relation
24	i take a dance class	i take	PlainPredicate	PlainPredicate
25	she fixes the plumbing	she plumbing	None relation	None relation
26	the lake reflects the sky	lake sky	None relation	None relation
27	the doctor examines the patient	examines patient	Predicate	Predicate
28	the ships transport goods across the sea	goods sea	None relation	None relation
29	the engineer constructed the buildings	engineer buildings	None relation	None relation

Table 3.4: Classification report for the evaluation of the proposed methodology for validation data

	None relation	Constraint	Predicate	PlainPredicate	accuracy	macro avg	weighted avg
precision	0.99	0.97	0.99	1.0	0.99	0.99	0.99
recall	0.99	0.99	0.99	0.99	0.99	0.99	0.99
f1-score	0.99	0.98	0.99	0.99	0.99	0.99	0.99
support	637.0	191.0	150.0	173.0	1151.0	1151.0	1151.0

Table 3.5: Compare the accuracy of the proposed model without and with regularization

Fold No	Accuracy in proposed model	Accuracy in baseline model	
1	0.973	0.839	
2	0.973	0.823	
3	0.993	0.832	
4	0.959	0.839	
5	0.987	0.829	
6	0.970	0.853	
7	0.950	0.847	
8	0.965	0.858	
9	0.966	0.837	
10	0.960	0.815	
	0.970	0.837	Average

generalization capability remains intact. In instances where a model exhibits exceptional performance during training, there exists a concern that it might become overly specialized to the training data, thereby hampering its ability to accurately predict unseen or new data points. To address the potential issue of overfitting and to bolster the model’s capacity for generalization, a combination of L1L2 regularization and Dropout techniques was introduced into the training process. The primary goal was to strike a balance between the model to capture relevant patterns and features from the data while preventing it from becoming excessively dependent on the training data set. As is shown in table 3.5, the accuracy scores obtained after regularization showed a decrease compared to the initial training performance. This observation prompts the consideration of how and why the incorporation of regularization techniques led to such outcomes.

One plausible explanation for the decrease in accuracy lies in the inherent nature of regularization techniques. While these techniques are potent tools for averting overfitting, they introduce constraints on the model’s learning process. In some cases, such constraints might inadvertently lead to underfitting, a scenario where the model fails to capture the nuanced relationships present in the data due to excessive regularization. The interplay between L1L2 regularization and Dropout might have resulted in a collectively strong constraint on the model, which, in turn, affected its ability to accurately generalize to new data points. The outcome of this experiment highlights the need to strike a delicate equilibrium between allowing the model to learn from the data and controlling

its propensity for overfitting. Future experiments might involve fine-tuning the hyperparameters associated with L1L2 regularization and Dropout to identify a configuration that better aligns with the model's architecture and the dataset's characteristics. Additionally, exploring other regularization strategies or adjusting the strength of regularization could provide insights into achieving optimal generalization without compromising learning capacity.

Chapter 4

Validation of the proposed methodology

4.1 Introduction

Chapter 4 provides a brief summary of evaluating the performance of the developed model by subjecting it to a comprehensive testing phase using unseen data. The purpose of this chapter is to understand the ability of the designed model to generalize and make accurate predictions beyond the training data. Validation serves as a critical step in assessing the reliability, robustness, and applicability of the proposed algorithm. In addition, during the validation phase, the model will be evaluated based on various performance metrics, including accuracy, precision, recall, and F1-score. These metrics provide quantitative measures of the model's performance, enabling us to assess strengths and limitations. A comprehensive analysis of the model's performance will be conducted to gain insights into its behaviour, identify potential areas of improvement, and understand its performance across different classes. In the subsequent sections of this chapter, we will present the experimental setup, describe the testing data set, and the criteria for preparing the data as well as the report based on the mentioned metrics. Furthermore, the findings from this validation process will serve as a basis for drawing conclusions and discussion.

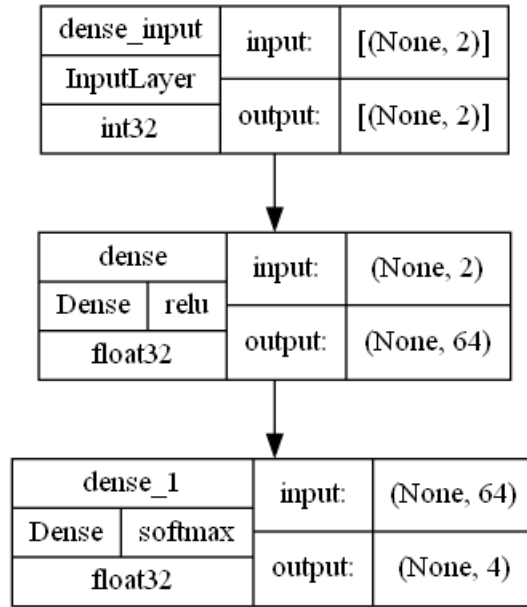


Figure 4.1: Baseline Model which is used for validating the proposed model.

4.2 Evaluation Methods

The objective of this section is to comprehensively explain the evaluation and validation procedures employed in this study. The first evaluation method involves creating a baseline model, which serves as a point of comparison for the proposed methodology. The baseline model chosen for this study is a Multi-layer Perceptron (MLP) model. As mentioned in figure 4.1, The MLP model comprises an input layer followed by a dense layer containing 64 nodes, and finally, an output layer consisting of four neurons. This choice of architecture for the baseline model was based on the proposed model and is the simplest version of the proposed model. In addition, The model is implemented as a sequential neural network that has the following information:

- The first layer is dense with 64 neurons and applies the Rectified Linear Unit (ReLU) activation function. This activation function introduces non-linearity into the model and helps in capturing complex relationships between the input features. The output of this layer is computed by taking a weighted sum of the inputs and applying the ReLU activation function to each neuron.
- The second layer is the output layer, which consists of four neurons. This layer uses the

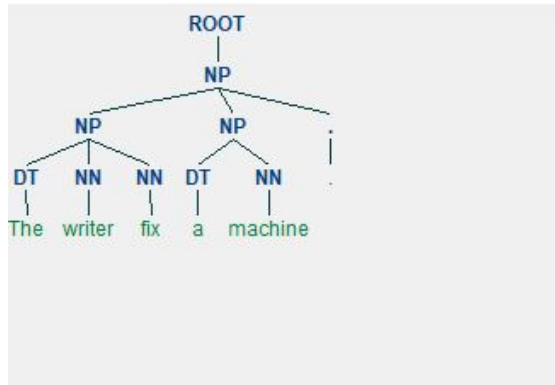


Figure 4.2: Demonstrating a problem in the Chomsky tree in detecting some verbs as nouns.

Patterns	Examples	Number of samples
Subject + Verb + Object	He takes photos.	100
Subject + Verb + "a" + Object	I take a class.	100
Subject + Verb + "the" + Object	You walk the dog.	100
"The" + Subject + Verb + Object	The truck carries goods.	100
"The" + Subject + Verb + "a or an" + Object	The cat catches a mouse.	100
"The" + Subject + Verb + "the" + Object	The engineer constructed the buildings.	100

Table 4.1: The Predefined patterns for collected sentences for testing the methodology.

softmax activation function, which is commonly used for multi-class classification problems. The softmax function outputs probabilities for each class, indicating the likelihood of the input belonging to each class. The sum of the probabilities across all classes is 1.

- The model is compiled using the sparse categorical cross-entropy loss function. This loss function is suitable for multi-class classification tasks where the target variable is represented as integers rather than one-hot encoded vectors. The optimizer used is Adam, which is an adaptive learning rate optimization algorithm. Adam adjusts the learning rate during training to accelerate convergence and improve performance.

Following the establishment of the baseline model, a comparative analysis was conducted to evaluate the performance of the proposed methodology against the baseline model. This analysis includes various metrics such as accuracy, precision, recall, and F1-score, which provide quantitative measures of the model's predictive capabilities and effectiveness in handling the given task. To further validate the proposed methodology, it is imperative to assess its performance on unseen or test data. This step ensures that the model's efficacy extends beyond the training data and is capable

of making accurate predictions on new and unseen samples.

One of the primary motivations for developing the proposed methodology is to address some specific limitations encountered by the rule-based system. One of the limitations, as depicted in the illustrations 4.2, highlights the particular situation in which the rule-based system fails to perform optimally. The Chomsky tree sometimes fails to detect some verbs because it considers those as nouns rather than verbs. Hence, one of the objectives of the evaluation is to examine whether the proposed model can effectively overcome this challenge and outperform the rule-based system in these scenarios. By incorporating these evaluation methods, this research aims to rigorously assess and validate the proposed methodology's performance and effectiveness. Through the comparison with the baseline model, the study aims to demonstrate improvements achieved by the proposed approach. Additionally, evaluating the methodology on unseen test data and addressing the limitations of the rule-based system contribute to the comprehensive evaluation of the proposed approach's practicality and potential for real-world applications.

4.3 Testing Data

4.3.1 Data Collection

This section is a brief overview of the systematic steps and processes followed to gather data for testing the proposed methodology. For accomplishing this objective, data is gathered from ChatGPT, a powerful language model, that provided a diverse range of sentences that served as valuable input for the validation of this study. As mentioned in chapter 3, the selection of data followed specific criteria based on sentence patterns. The initial focus was on basic sentence structures, such as "subject + verb + object", which is illustrated in table 4.1. In addition, The data collection process played a crucial role in obtaining the necessary samples for testing the machine learning-based approach for generating Recursive Object Model (ROM) diagrams. As discussed in the previous chapter, to streamline and automate this process, a web application named ROMWeb was developed. ROMWeb served as a user-friendly tool that enabled researchers to input sentences and generate corresponding ROM diagrams.

4.3.2 Data Prepration

As discussed in chapter 3, data preprocessing is a crucial phase in machine learning projects as it involves transforming raw data into a suitable format for analysis and model training. This section summarizes the data preprocessing procedures that were undertaken for both the training and testing data sets, ensuring consistency and reliability in the preprocessing steps. This section provides an overview of the essential data preparation procedures that have been done on testing data, including: data loading, word pair generation, label conversion, part-of-speech tag incorporation, pair distance calculation, and categorical data encoding. The procedures will be discussed as follows:

- **Data Loading and Transformation:** Raw data from a MongoDB database were extracted and transformed into a structured format using Python and Pandas, enabling seamless manipulation and exploration of the data set.
- **Word Pair Generation:** A systematic approach was employed to identify and extract word pairs from sentences, providing candidate pairs for further analysis and research stages.
- **Converting Labels to Numbers:** Relationships between word pairs were transformed from linguistic form to numerical representations, facilitating subsequent analysis and modelling tasks that rely on numerical labels.
- **Part-of-Speech Feature Incorporation:** Part-of-speech (POS) tags were calculated for individual words, enhancing the dataset's linguistic information and enabling comprehensive exploration of interrelationships and patterns.
- **Pair Distance Calculation:** The distance between word pairs within sentences were quantitatively measured, enriching the dataset with spatial information for a nuanced exploration of linguistic associations and dependencies.
- **Categorical Data Encoding:** Categorical variables, such as word types, were encoded into numerical representations, ensuring compatibility with machine learning models and preserving essential information for accurate analysis and prediction.

By performing these preprocessing steps consistently for both the testing data set, we ensured that the data sets were suitably prepared for subsequent analysis and evaluation.

4.4 Result and Evaluation

This section presents a comprehensive analysis of the outcomes obtained through a series of evaluation procedures. To accomplish these objectives, the results and evaluations encompass three main aspects. These illustrate the comparison result between the baseline model and the proposed model, then illustrate the evaluation of the proposed model based on the test samples, and finally determine whether the proposed model is able to overcome the mentioned limitations or not.

4.4.1 Baseline model vs Proposed model

This section presents a comparative analysis between the proposed model and a baseline model, aiming to evaluate the performance of the proposed approach. The baseline model was trained on the same data set as the proposed model, and the evaluation was conducted using 10-fold cross-validation. The accuracy and loss of the validation data for the best fold were examined, along with a classification report and selected samples. The results highlight the differences between the two models in terms of accuracy and loss, providing insights into their relative performance.

Figures 4.3, 4.4 illustrating the accuracy and loss trends of the validation data were generated. Additionally, a classification report was obtained, providing metrics such as recall, precision, F1-score, and accuracy for each label predicted by the baseline model as shown in the table 4.2. In addition, table 4.3 demonstrates the model's performance, by randomly selecting 30 validation samples.

A comparison of the proposed model and the baseline model reveals notable differences in their performance. The accuracy table presents the accuracy values for each fold, along with the average accuracy across all folds 4.4. The proposed model achieves an average accuracy of approximately 93 percent, outperforming the baseline model, which achieves an average accuracy of around 61 percent. This significant difference demonstrates the superior performance of the proposed model in capturing patterns within the data. Moreover, an analysis of the best-performing fold of the

Table 4.2: Classification report for the evaluation of the baseline model for validation data

	None relation	Constraint	Predicate	PlainPredicate	accuracy	macro avg	weighted avg
precision	0.79	0.57	0.64	0.51	0.66	0.63	0.69
recall	0.63	0.86	0.65	0.53	0.66	0.67	0.66
f1-score	0.70	0.69	0.64	0.52	0.66	0.64	0.66
support	554.0	197.0	155.0	147.0	1053.0	1053.0	1053.0

Table 4.3: The validation result for the baseline model for the best-performing fold.

No	pair	Predicted relation	actual relation
0	you the	None relation	None relation
1	the new	Constraint	None relation
2	cat caught	Predicate	PlainPredicate
3	we take	None relation	PlainPredicate
4	the competition	Constraint	Constraint
5	birds sing	Predicate	PlainPredicate
6	a class	None relation	Constraint
7	you a	None relation	None relation
8	the a	None relation	None relation
9	chop the	None relation	None relation

baseline model highlights its limitations. Although this fold exhibits relatively low accuracy, the corresponding loss is significantly higher compared to the proposed model. This suggests that the baseline model struggles to generalize and make accurate predictions beyond the training data. The classification report provides further insights into the baseline model’s performance. It demonstrates variations in recall, precision, F1-score, and accuracy across different labels, shedding light on the model’s ability to correctly classify instances for each class. These metrics further emphasize the disparity between the proposed and baseline models. Additionally, by examining selected samples from the validation data, it becomes apparent that the baseline model struggles to accurately predict certain instances. This observation highlights the limitations of the baseline model and its potential shortcomings in practical applications.

The comparative analysis between the proposed model and the baseline model underscores the superior performance of the proposed approach. With significantly higher accuracy and lower loss, the proposed model demonstrates its ability to capture patterns more effectively and make more reliable predictions. In contrast, the baseline model exhibits limitations in accuracy and struggles to generalize beyond the training data. These findings validate the efficacy of the proposed model and emphasize its potential for practical applications in the domain under study.

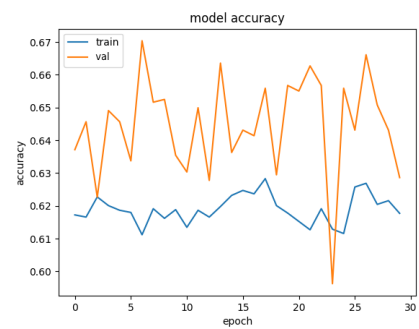
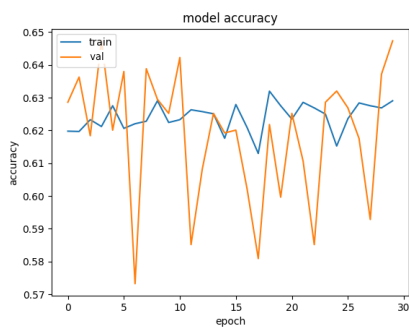
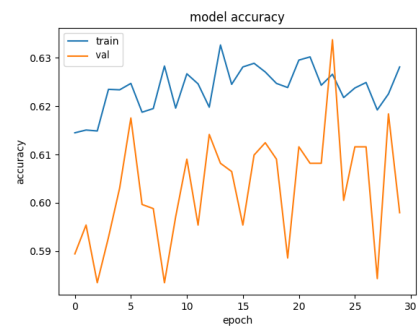
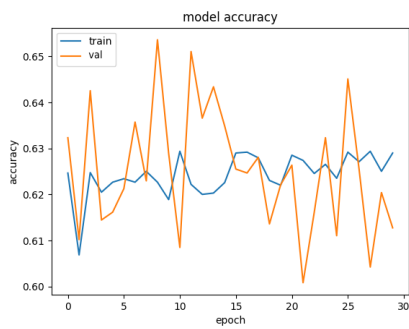
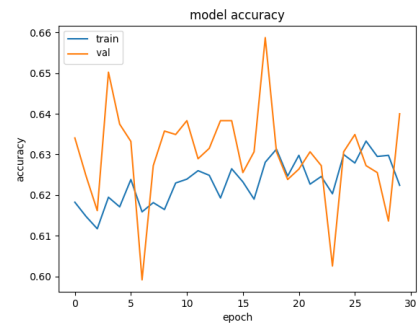
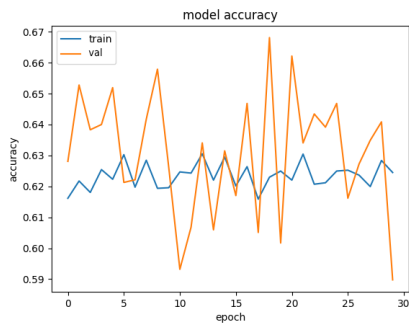
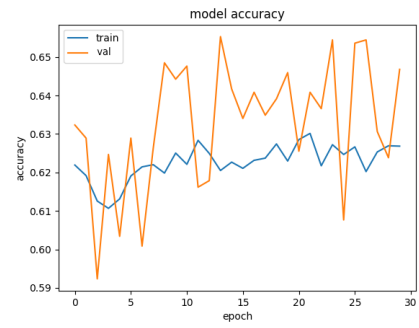
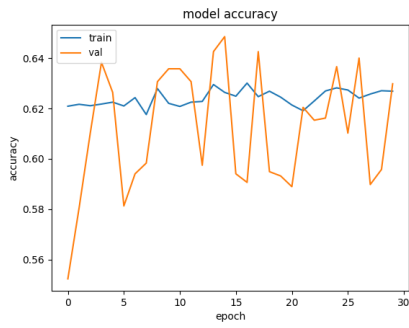
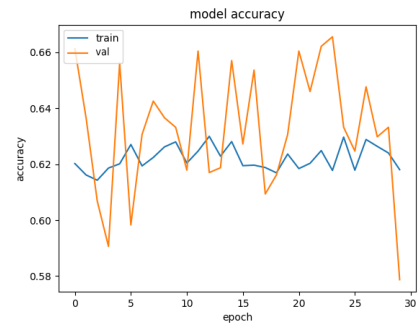
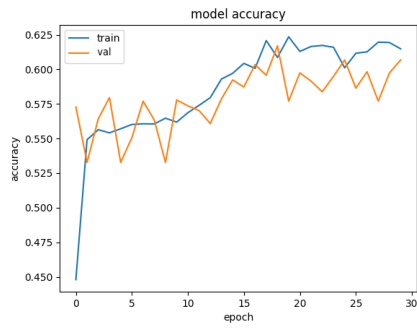


Figure 4.3: Accuracy of the baseline model in 10 folds

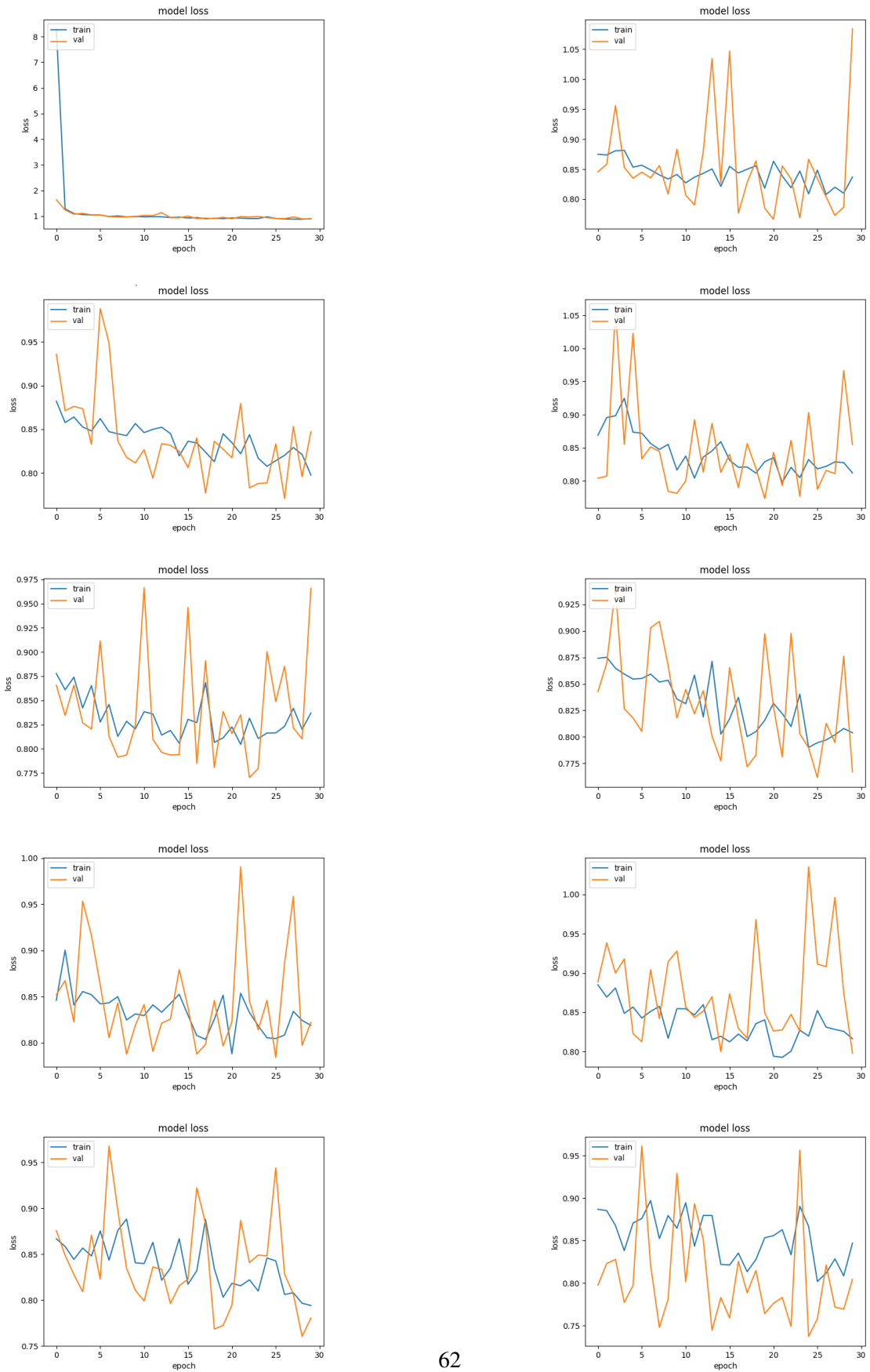


Figure 4.4: Loss of the baseline model in 10 folds

Table 4.4: The accuracy of the proposed model vs the baseline model

Fold No	Accuracy in proposed model	Accuracy in baseline model	
1	0.973	0.606	
2	0.973	0.578	
3	0.993	0.629	
4	0.959	0.646	
5	0.987	0.589	
6	0.970	0.639	
7	0.950	0.612	
8	0.965	0.597	
9	0.966	0.647	
10	0.960	0.628	
	0.970	0.617	Average

4.4.2 Evaluating Proposed Methodology

This section presents the evaluation of the proposed methodology using the 600 sentences that were not utilized in the training and validation process. After comparing the performance of the baseline model with the proposed model, the focus shifts to assessing the proposed model’s effectiveness in making predictions on unseen data. The best-performing fold from the proposed model, based on accuracy, is selected for generating a ROM diagram. Additionally, a classification report and a random selection of 10 sentences from the testing data are presented to provide further insights into the model’s performance.

The proposed model’s performance is evaluated using the testing data, employing the best-performing fold from the training phase. This fold, determined based on its high accuracy, is chosen to generate a ROM diagram, which provides insights into the model’s predictive performance. Additionally, two tables are presented: table 4.5 containing a classification report, offering metrics such as accuracy, precision, recall, and F1-score, and table 4.6 displaying a random selection of 10 sentences from the testing data.

The evaluation of the proposed methodology on the testing data reveals promising results. The ROM diagram, generated using the best-performing fold, showcases the model’s ability to achieve high true positive rates while maintaining low false positive rates. This indicates the model’s effectiveness in correctly classifying instances and minimizing erroneous predictions. The classification report provides comprehensive metrics, including accuracy, precision, recall, and F1-score, for

each label predicted by the proposed model. Notably, the proposed model demonstrates acceptable accuracy of 96 percent on the testing data. This high accuracy indicates the model's robustness and consistency in making accurate predictions on previously unseen instances. Furthermore, the performance on the testing data closely aligns with the performance observed during the training phase, indicating the model's generalization capability. The results showcase the model's ability to correctly classify various instances, reinforcing the overall accuracy observed in the classification report. Although the overall performance of the model is acceptable for the first version of the model, it is noteworthy that the precision of the constraint relationship label is comparatively lower than other labels. To be specific, out of 3890 samples in the testing data, 154 were incorrectly classified by the model. The majority of these misclassifications occurred when the two words in a sentence had a significant distance. Additionally, the model frequently predicted constraint relationships between determiners and verbs, which is not the correct association. This discrepancy may be attributed to two main factors. Firstly, the data used for training is imbalanced, meaning that there are significantly more instances of certain labels than others, leading to bias in the model's predictions. Secondly, the input features used to represent the data may not fully capture the complex patterns and dependencies required to accurately identify constraint relationships.

4.4.3 Proposed Model vs Rule-based system

This section aims to compare the performance of the proposed model with a rule-based system, with the objective of assessing whether the proposed model is capable of addressing the limitations inherent in the rule-based approach. To compare the proposed model with the rule-based system, sentences are gathered to test the proposed model representing the limitations. The output results from both approaches are then analyzed and compared to determine the extent to which the proposed model addresses the limitations of the rule-based system. The comparison between the proposed model and the rule-based system reveals insights into the capabilities of the proposed model. The results clearly indicate that the proposed model surpasses the limitations of the rule-based system, however, in a few cases, it has errors as well. Although there are still areas for improvement, such as the inclusion of additional training samples and patterns, the results highlight the capacity of the proposed model to alleviate the limitations of the rule-based system.

Table 4.5: Classification report for the evaluation of the proposed methodology for testing data

	None relation	Constraint	Predicate	PlainPredicate	accuracy	macro avg	weighted avg
precision	0.972	0.889	0.996	0.972	0.960	0.957	0.961
recall	0.975	0.928	0.908	0.998	0.960	0.952	0.960
f1-score	0.974	0.908	0.950	0.985	0.960	0.954	0.960
support	1992.0	699.0	600.0	599.0	3990.0	3990.0	3990.0

Table 4.6: The result of evaluating the trained model for the best-performing fold on the testing data

Sample No	Sentences	Pairs	Predicted relations	Actual Relations
1	he bakes a cake	a cake	Constraint	Constraint
	he bakes a cake	bakes cake	Predicate	Predicate
	he bakes a cake	he bakes	PlainPredicate	PlainPredicate
	he bakes a cake	he a	None relation	None relation
	he bakes a cake	bakes a	None relation	None relation
	he bakes a cake	he cake	None relation	None relation
2	he conducts the orchestra	he orchestra	None relation	None relation
	he conducts the orchestra	he the	None relation	None relation
	he conducts the orchestra	the orchestra	Constraint	Constraint
	he conducts the orchestra	conducts orchestra	Predicate	Predicate
	he conducts the orchestra	he conducts	PlainPredicate	PlainPredicate
	he conducts the orchestra	conducts the	None relation	None relation
3	i solve the puzzle	i puzzle	None relation	None relation
	i solve the puzzle	solve puzzle	Predicate	Predicate
	i solve the puzzle	solve the	None relation	None relation

Sample No	Sentences	Pairs	Predicted relations	Actual Relations
	i solve the puzzle	i the	None relation	None relation
	i solve the puzzle	i solve	PlainPredicate	PlainPredicate
	i solve the puzzle	the puzzle	Constraint	Constraint
4	i treat an illness	an illness	Constraint	Constraint
	i treat an illness	i treat	PlainPredicate	PlainPredicate
	i treat an illness	i an	None relation	None relation
	i treat an illness	treat an	None relation	None relation
	i treat an illness	treat illness	Predicate	Predicate
	i treat an illness	i illness	None relation	None relation
5	the architect designs the building	designs building	Predicate	Predicate
	the architect designs the building	the architect	Constraint	Constraint
	the architect designs the building	architect building	None relation	None relation
	the architect designs the building	designs the	None relation	None relation
	the architect designs the building	architect the	None relation	None relation
	the architect designs the building	the designs	None relation	None relation
	the architect designs the building	the building	None relation	Constraint
	the architect designs the building	architect designs	PlainPredicate	PlainPredicate
6	the architects con- struct the buildings	the architects	Constraint	Constraint

Sample No	Sentences	Pairs	Predicted relations	Actual Relations
	the architects construct the buildings	architects buildings	None relation	None relation
	the architects construct the buildings	construct the	None relation	None relation
	the architects construct the buildings	architects construct	PlainPredicate	PlainPredicate
	the architects construct the buildings	architects the	None relation	None relation
	the architects construct the buildings	the buildings	Constraint	Constraint
	the architects construct the buildings	the construct	None relation	None relation
	the architects construct the buildings	construct buildings	Predicate	Predicate
7	the teacher instructs the students	instructs students	Predicate	Predicate
	the teacher instructs the students	teacher the	None relation	None relation
	the teacher instructs the students	teacher students	None relation	None relation
	the teacher instructs the students	the instructs	None relation	None relation
	the teacher instructs the students	instructs the	None relation	None relation
	the teacher instructs the students	the students	Constraint	Constraint

Sample No	Sentences	Pairs	Predicted relations	Actual Relations
	the teacher instructs the students	teacher instructs	PlainPredicate	PlainPredicate
	the teacher instructs the students	the teacher	Constraint	Constraint
8	the teachers gave a les- son	the teachers	Constraint	Constraint
	the teachers gave a les- son	a lesson	Constraint	Constraint
	the teachers gave a les- son	the lesson	Constraint	None relation
	the teachers gave a les- son	teachers gave	PlainPredicate	PlainPredicate
	the teachers gave a les- son	the a	None relation	None relation
	the teachers gave a les- son	gave a	None relation	None relation
	the teachers gave a les- son	the gave	None relation	None relation
	the teachers gave a les- son	gave lesson	Predicate	Predicate
	the teachers gave a les- son	teachers lesson	None relation	None relation
	the teachers gave a les- son	teachers a	None relation	None relation
9	the teachers grade the papers	teachers the	None relation	None relation

Sample No	Sentences	Pairs	Predicted relations	Actual Relations
	the teachers grade the papers	grade papers	Predicate	Predicate
	the teachers grade the papers	grade the	None relation	None relation
	the teachers grade the papers	teachers grade	PlainPredicate	PlainPredicate
	the teachers grade the papers	the grade	None relation	None relation
	the teachers grade the papers	the teachers	Constraint	Constraint
	the teachers grade the papers	the papers	None relation	Constraint
	the teachers grade the papers	teachers papers	None relation	None relation
10	he conducts the re-search	the research	None relation	Constraint
	he conducts the re-search	he the	None relation	None relation
	he conducts the re-search	conducts research	Constraint	Predicate
	he conducts the re-search	conducts the	None relation	None relation
	he conducts the re-search	he conducts	PlainPredicate	PlainPredicate
	he conducts the re-search	he research	None relation	None relation

4.5 Discussion and Conclusion

The primary objective of chapter 4 was to assess the generalization ability and accuracy of predictions made by the designed model beyond the training data. Validation serves as a crucial step in determining the reliability, robustness, and applicability of the proposed algorithm. Throughout the validation phase, the model was assessed using various performance metrics, including accuracy, precision, recall, and F1-score, which provided quantitative measures of its performance and allowed for a thorough assessment of strengths and limitations.

The comparative analysis between the proposed model and the baseline model yielded significant insights into their performance disparities. The baseline model was trained on the same data set as the proposed model, and its evaluation was conducted using 10-fold cross-validation. The accuracy and loss trends of the validation data were examined, along with a comprehensive classification report and a selection of randomly chosen samples. These results provided valuable insights into the differences between the two models, particularly in terms of accuracy and loss, enabling a deeper understanding of their relative performance.

The accuracy table presented the accuracy values for each fold, along with the average accuracy across all folds. The proposed model consistently outperformed the baseline model, achieving an average accuracy of approximately 97 percent compared to the baseline model's average accuracy of around 61 percent. This substantial difference in accuracy highlighted the superior performance of the proposed model in capturing patterns within the training data. The evaluation of the proposed methodology using the testing data highlights the model's effectiveness and generalization capability. With a high accuracy of 96 percent on the testing data, the proposed model demonstrates its ability to make accurate predictions on unseen instances. The ROM diagram emphasizes the model's ability to achieve high true positive rates and low false positive rates, further substantiating its performance. While the overall performance of the model is satisfactory for its initial version, it is essential to acknowledge the limitations and areas for improvement, particularly concerning the precision of the constraint relationship label. These findings validate the proposed methodology's efficacy and suggest its potential for practical applications in the relevant domain. The comparison between the proposed model and the rule-based system emphasizes the superiority of the proposed

model in addressing the limitation inherent in the rule-based approach. Although there are still opportunities for enhancements, such as incorporating more training samples and patterns, the results underscore the potential of the proposed model in mitigating the limitations of the rule-based system.

Chapter 5

Conclusion and future works

Chapter 5 serves to summarize the findings and outcomes of the research and discuss potential directions for future exploration. In this section, we present the conclusions drawn from the investigation into employing intelligent systems and machine learning algorithms for generating Recursive Object Model (ROM) diagrams as an alternative to the rule-based method. We also outline potential future research and improvement areas to enhance the proposed approach further.

5.1 Conclusion

In conclusion, this research has explored the potential of machine learning algorithms as an alternative approach to rule-based systems for generating Recursive Object Model (ROM) diagrams. These limitations are limited adaptability, lack of contextual understanding, knowledge acquisition and maintenance issues, difficulty in handling ambiguity, generalization limitations, and manual rule construction. Furthermore, by adapting and learning from data, this research provides a framework that has the potential to capture linguistic patterns as well as semantic patterns to enhance the accuracy and efficiency of ROM diagram generation.

In addition, by comparing the proposed intelligent system and the rule-based system, while the proposed methodology has some minor errors in generating the ROM diagrams for those scenarios, the results underscore the potential of the proposed model in mitigating the limitations of the rule-based system.

Furthermore, the research has evaluated the adaptability of the machine learning-based approach in handling language patterns and structures. Since the proposed model can learn new patterns by providing new structures in the natural language, the process of knowledge acquisition and maintenance is automated, the study has addressed the challenges of manual rule construction, ensuring that the system stays up-to-date with evolving language patterns. Therefore, this led to improved efficiency and reduced human effort in maintaining the ROM generation process. Moreover, the research has enhanced the contextual understanding of ROM diagrams by incorporating the proposed model. By capturing semantic and contextual aspects of natural language, the proposed algorithm has improved the overall quality and comprehensiveness of the generated ROM relations. This advancement goes beyond the syntactic relationships captured by the rule-based system, providing a more holistic representation of natural language in generating the ROM diagrams.

In light of the quality enhancement of the ROM diagrams, which serve as the brain of the Environment-Based Design (EBD) methodology, designers following this approach can more accurately understand the subsequent steps in generating solutions, leading to greater creativity and improved outcomes. The successful application of the proposed model signifies the potential for integrating AI and ML-based systems into the design process, effectively facilitating and streamlining various aspects of the design workflow. Looking to the future, focusing on machine learning-based systems offers a promising path for continuously improving and upgrading the ROMA software. Rather than investing substantial time in understanding roles within the rule-based system and the underlying source code, the development and enhancement of ROMA can be facilitated by providing more samples to the machine learning model.

5.2 Limitations and Future works

While this research has made strides in leveraging machine learning algorithms for the creation of ROM diagrams, it is essential to acknowledge its limitations. Despite satisfactory performance within defined patterns, future versions should encompass a wider array of natural language patterns. Moreover, some minor inaccuracies have been noted in the ROM diagrams produced by the proposed model, especially in labelling constraint relationships, potentially due to the limited

features set for training data. In light of the aforementioned research findings and achievements, several promising avenues for future work emerge. These directions aim to enhance the proposed model's generalizability, integration, handling of complex sentences, addressing data imbalances, and leveraging advanced embedding techniques for capturing semantic relationships. The following areas warrant particular attention:

- (1) **Gather Comprehensive Patterns and Samples:** The current model's generalizability can be improved by assembling a more comprehensive dataset encompassing diverse patterns and samples. This entails gathering a larger corpus of sentences that exhibit a wide range of structures and relationships. By incorporating more extensive and diverse examples, the model can learn to handle various linguistic nuances, resulting in enhanced performance and robustness.
- (2) **Feature Engineering:** There is a need to improve the model's performance by incorporating additional features. These enhancements aim to increase the capability of the model to interpret complex sentence patterns.
- (3) **Integration with ROMA software:** After generalizing the model and confirming its performance, integrating it with the ROMA software to replace the Chomsky tree can be pursued. This integration would allow for seamless integration of the proposed model within the existing framework, enhancing the capabilities of ROMA and enabling more efficient analysis of sentence and decision-making processes in EBD methodology.
- (4) **Handling Complex Sentences:** If the proposed model struggles with complex sentences, the following solutions can be explored:
 - **Improving Word Pair Generation Algorithm:** Enhancing the algorithm for generating word pairs by considering additional conditions beyond distance and part-of-speech tags can be beneficial.
 - **Exploring Different RNN Architectures:** If necessary, alternative recurrent neural network architectures other than LSTM can be investigated to enhance the model's performance. Models like Gated Recurrent Units or advanced transformer architectures

can handle longer sentences and more intricate sentence structures.

- **Addressing Imbalanced Data:** Given the presence of imbalanced data arising from the existence of the "None" relation, alternative algorithms such as XGBClassifier should be considered instead of Multilayer Perceptron (MLP). These algorithms are specifically designed to handle imbalanced datasets, mitigating the challenges associated with the underrepresentation of certain relationship types. This adaptation would improve the model's ability to recognize and classify different relationship categories accurately.

- (5) **Utilizing Advanced Embedding Techniques:** In pursuit of capturing semantic relationships more effectively, incorporating state-of-the-art embedding techniques holds considerable promise. These techniques, rooted in cutting-edge natural language processing advancements, can represent natural language in numerical form, enabling the model to capture subtle semantic nuances and achieve a deeper understanding of sentence relationships. By leveraging these advanced embedding techniques, the model's performance in capturing complex semantic associations can be significantly enhanced.

By addressing these future works, the proposed model can be further refined, resulting in improved generalization, seamless integration with the ROMA software, enhanced handling of complex sentences, mitigation of data imbalances, and more effective capturing of semantic relationships. These advancements hold the potential to yield valuable contributions to the field of sentence relationship analysis and its applications in the field of design methodology.

References

- Arseniev-Koehler, A., & Foster, J. G. (2020, August). *Sociolinguistic Properties of Word Embeddings* (preprint). SocArXiv. Retrieved 2023-05-19, from <https://osf.io/b8kud> doi: 10.31235/osf.io/b8kud
- Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. *Proceedings of the Third Conference on Applied Natural Language Processing - Association for Computational Linguistics*(1), p. 152. Retrieved from <https://doi.org/10.3115/974499.974526>. doi: 10.3115/974499.974526
- Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, Mass: MIT Press. (OCLC: ocm64898359)
- Chen, Z. Y., & Zeng, Y. (2006, September). Classification of Product Requirements Based on Product Environment. *Concurrent Engineering*, 14(3), 219–230. Retrieved 2023-06-30, from <http://journals.sagepub.com/doi/10.1177/1063293X06068389> doi: 10.1177/1063293X06068389
- Chiche, A., & Yitagesu, B. (2022, January). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1), 10. Retrieved 2023-05-10, from <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00561-y> doi: 10.1186/s40537-022-00561-y
- Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020, February). Introduction to Machine Learning, Neural Networks, and Deep Learning. *Neural Networks*.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011, March). *Natural Language Processing (almost) from Scratch*. arXiv. Retrieved 2023-05-08, from

- <http://arxiv.org/abs/1103.0398> (arXiv:1103.0398 [cs])
- Crawshaw, M. (2020, September). *Multi-Task Learning with Deep Neural Networks: A Survey*. arXiv. Retrieved 2023-08-08, from <http://arxiv.org/abs/2009.09796> (arXiv:2009.09796 [cs, stat])
- Graves, A., & Schmidhuber, J. (2005, July). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6), 602–610. Retrieved 2023-06-30, from <https://linkinghub.elsevier.com/retrieve/pii/S0893608005001206> doi: 10.1016/j.neunet.2005.06.042
- Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2018, October). *Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change*. arXiv. Retrieved 2023-07-02, from <http://arxiv.org/abs/1605.09096> (arXiv:1605.09096 [cs])
- Hinton, G. E., Sejnowski, T. J., Hinton, G. E. E., & Sejnowski, T. J. E. (1999). *Unsupervised Learning: Foundations of Neural Computation*. MIT Press.
- Hochreiter, S., & Schmidhuber, J. (1997, November). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. Retrieved 2023-05-27, from <https://direct.mit.edu/neco/article/9/8/1735-1780/6109> doi: 10.1162/neco.1997.9.8.1735
- Ibrihich, S., Oussous, A., Ibrihich, O., & Esghir, M. (2022). A Review on recent research in information retrieval. *Procedia Computer Science*, 201, 777–782. Retrieved 2023-05-14, from <https://linkinghub.elsevier.com/retrieve/pii/S1877050922005191> doi: 10.1016/j.procs.2022.03.106
- Ide, N., & Veronis, J. (1998). Word sense disambiguation. *Computational Linguistics*.
- Jordan, I. D., Sokol, P. A., & Park, I. M. (2021, July). Gated recurrent units viewed through the lens of continuous time dynamical systems. *Frontiers in Computational Neuroscience*, 15, 678158. Retrieved 2023-06-30, from <http://arxiv.org/abs/1906.01005> (arXiv:1906.01005 [cs, stat]) doi: 10.3389/fncom.2021.678158
- Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023, January). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3), 3713–3744. Retrieved 2023-05-08, from <https://link.springer.com/10.1007/s11042-022-13428-4> doi: 10.1007/s11042-022-13428-4

- Kim, S.-W., & Gil, J.-M. (2019, December). Research paper classification systems based on TF-IDF and LDA schemes. *Human-centric Computing and Information Sciences*, 9(1), 30. Retrieved 2023-05-19, from <https://link.springer.com/10.1186/s13673-019-0192-7> doi: 10.1186/s13673-019-0192-7
- Liddy, E. D. (2001). Natural Language Processing. In *Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.*. Retrieved from <https://surface.syr.edu/istpub/63/>
- Lipton, Z. C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning.
- Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Springer International Publishing*. Retrieved from <https://doi.org/10.1007/978-3-031-02145-9> doi: 10.1007/978-3-031-02145-9
- Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). *Machine Learning* (??th ed.). CRC Press. Retrieved 2023-05-18, from <https://www.taylorfrancis.com/books/9781498705394> doi: 10.1201/9781315371658
- Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer Perceptron and Neural Networks. , 8(7).
- Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019, June). An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges. In *2019 International Engineering Conference (IEC)* (pp. 200–204). Erbil, Iraq: IEEE. Retrieved 2023-05-18, from <https://ieeexplore.ieee.org/document/8950616/> doi: 10.1109/IEC47844.2019.8950616
- Ranjan Pal, A., & Saha, D. (2015, July). Word Sense Disambiguation: A Survey. *International Journal of Control Theory and Computer Modeling*, 5(3), 1–16. Retrieved 2023-05-10, from <http://www.airccse.org/journal/ijctcm/papers/5315ijctcm01.pdf> doi: 10.5121/ijctcm.2015.5301
- Roy, A. (2021, January). *Recent Trends in Named Entity Recognition (NER)*. arXiv. Retrieved 2023-05-10, from <http://arxiv.org/abs/2101.11420> (arXiv:2101.11420 [cs])
- Sarker, I. H. (2021, May). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. Retrieved 2023-05-18, from <https://>

link.springer.com/10.1007/s42979-021-00592-x doi: 10.1007/s42979-021-00592-x

- Schuster, M., & Paliwal, K. (1997, November). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. Retrieved 2023-06-30, from <http://ieeexplore.ieee.org/document/650093/> doi: 10.1109/78.650093
- Simeone, O. (2018, November). *A Very Brief Introduction to Machine Learning With Applications to Communication Systems*. arXiv. Retrieved 2023-05-16, from <http://arxiv.org/abs/1808.02342> (arXiv:1808.02342 [cs, math])
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017, December). *Attention Is All You Need*. arXiv. Retrieved 2023-05-26, from <http://arxiv.org/abs/1706.03762> (arXiv:1706.03762 [cs])
- Zeng, Y. (2002). AXIOMATIC THEORY OF DESIGN MODELING. *Journal of Integrated Design and Process Science*.
- Zeng, Y. (2008, August). Recursive object model (ROM)—Modelling of linguistic information in engineering design. *Computers in Industry*, 59(6), 612–625. Retrieved 2023-05-02, from <https://linkinghub.elsevier.com/retrieve/pii/S0166361508000249> doi: 10.1016/j.compind.2008.03.002
- Zeng, Y. (2015, June). Environment-Based Design (EBD): a Methodology for Transdisciplinary Design+. *Journal of Integrated Design and Process Science*, 19(1), 5–24. Retrieved 2023-05-03, from <https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/jid-2015-0004> doi: 10.3233/jid-2015-0004
- Zeng, Y. (2021, July). Environment: The First Thing to Look at in Conceptual Design. *Journal of Integrated Design and Process Science*, 24(1), 45–66. Retrieved 2023-05-02, from <https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/JID200005> doi: 10.3233/JID200005