ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

EXTRACTING GRAPH-STRUCTURED INFORMATION FROM SIMPLE TEXT

Διπλωματική Εργασία

της

Δέσποινα Ιωακειμίδου

Θεσσαλονίκη, Φεβρουάριος/2020

ΕΞΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΑΣ ΔΟΜΗΜΕΝΗΣ ΩΣ ΓΡΑΦΗΜΑΤΑ ΑΠΟ ΑΠΛΟ ΚΕΙΜΕΝΟ

Δέσποινα Ιωακειμίδου

Μηχανικών Παραγωγής και Διοίκησης, ΔΠΘ, 2010

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπουσα Καθηγήτρια
Κολωνιάρη Γεωργία

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26/02/2020

Κολωνιάρη Γεωργία	Σακελλαρίου Ηλίας	Κεραμόπουλος Ευκλείδης
.....	.....	.....

Δέσποινα Ιωακειμίδου

.....

Περίληψη

Η επεξεργασία φυσικής γλώσσας αποτελεί έναν κλάδο της επιστήμης της πληροφορικής που αναπτύσσεται συνεχώς τα τελευταία 70 χρόνια και έχει αλλάξει τον τρόπο που οι άνθρωποι αλληλεπιδρούν με τους υπολογιστές. Με την ανάπτυξη του διαδικτύου τα τελευταία χρόνια καθημερινά δημιουργείται ένας τεράστιος όγκος δεδομένων κειμένου και η σωστής επεξεργασίας τους είναι πλέον αναγκαία. Η διαχείριση και η αποθήκευση των δεδομένων κειμένου έχει προσελκύσει το ενδιαφέρον των ερευνητών με διάφορετικους τρόπουυς ενώ σημαντική είναι επίσης και η ανάλυση των δεδομένων για εμπορικούς λόγους. Η ανάλυση μεγάλου όγκου δεδομένων, που αποτελει ενα κομμάτι της επεξεργασίας φυσικής γλώσσας, αναπτύσσεται πλεον ταχύτατα. Δεδομένου οτι κάθε κείμενο αποτελείται απο μια αλληλουχία λέξεων-δεδομένων, οι οποίες είναι συνδεδεμένες μεταξύ τους συντακτικά και οτι οι βάσεις δεδομένων με γράφους ειναι αρκετα αποδοτικες οταν τα δεδομένα είναι συσχετισμένα, η επεξεργασία φυσικής γλώσσας και η Neo4j αποτελουν εναν αποτελεσματικο συνδυασμό. Η παρούσα εργασία εστιάζει στην επεξεργασία αδόμητου κειμένου και στη δημιουργία γράφου, ο οποίος θα αντιπροσωπεύει τις ιδιότητες των οντοτήτων του κειμένου αλλα και τις συσχετισεις που υπάρχουν. Θεωρείται οτι η προεπεξεργασία του κειμένου είναι αρκετά σημαντική για την επιτυχια του μοντέλου αυτου, καθώς το κείμενο που εισάγεται μπορεί να είναι αρκετά πολύπλοκης μορφής. Με την προεπεξεργασία το αδόμητο κείμενο μετατρέπεται σε μια μορφή διαχειρίσιμη για περαιτέρω ανάλυση, χωρίς όμως να χάνεται πολύτιμη πληροφορία. Τα αποτελέσματα δείχνουν οτι με κάποιες προϋποθέσεις οσον αφορα τη σύνταξη της πρότασης, τα ήδη συσχετισμένα δεδομένα κειμένου οπτικοποιούνται αποδοτικά σε γράφο.

Λέξεις Κλειδιά : επεξεργασία φυσικής γλώσσας, γράφος, βάσεις δεδομένων με γράφους

Abstract

Natural language processing is an industry of computer science that has been continually evolving for the last 70 years, and it has changed the way people interact with computers. With the development of the internet in the last few years, various amounts of text data are created every day, and their proper processing is now necessary. Managing and storing text data has attracted researchers' interest in several ways, and data analysis for commercial purposes is also essential. The analysis of large volumes of data, which is a part of natural language processing, is rapidly expanding. Given that each text consists of a sequence of data-words that are syntactically correlated and graph databases are quite efficient when the data is correlated, natural language processing and Neo4j are a compelling combination. This work focuses on constructing the unstructured text and creating a graph, which will represent the properties of the entities of the text and the relationships between the words. It is considered that preprocessing of the text is essential for this model's success, as the text being imported can be quite complicated. Preprocessing the unstructured text is converted into a manageable format for further analysis without losing valuable information. The results show that the syntax form of the sentence significantly affects the performance of the application. With a specific syntax form, the application can perform exceptionally well.

Keywords:Natural language processing, Neo4j, graph databases.

Acknowledgments

I would like to thank my supervisor Georgia Koloniari for her support and guidance during the completion of my master-thesis. Also I would like to thank my family for their support.

Contents

List of Images

List of Tables

1 Introduction

1.1 Problem Statement

Along with the growth of the web and social media, there is an increase in online data. As the number of data increases, the mechanisms to process data and extract valuable information become challenging. Natural language processing (NLP) is a sub-area of artificial intelligence that deals with human interaction with the machine [1]. Natural language processing (NLP) has been noticed very often in computer science in the past few decades. Enormous attempts have also been made in words field of NLP to convert the natural language processing text into computer programs [2].

Natural language processing is not a new science. Though, it is a complicated science due to the nature of human language. While humans can quickly master a language, the ambiguity and imprecise characteristics of the natural languages make NLP difficult for machines to implement. However, thanks to the increased interest in human-to-machine communications and the significant amount of available data, NLP's technology is rapidly advancing.

NLP's most popular applications are speech recognition, data extraction, text summarization, question-answering, speech combination, and Natural Language Interfaces to Database. The NLP techniques are also beneficial for sentiment analysis. All these applications are already being used in companies. NLP for business is an increasingly popular topic. Most companies try to automate customer support, analyze feedback, make better marketing strategies, understand business trends, or collect user data insights.

Most of the time, the data produced by an NLP application are unstructured but interconnected, and they need further transformation, which makes the use of an information system necessary. The goal of any information system is to transform data into information. A database is designed to take corresponding data and provide tools for aggregation and analysis. A graph database is a data management system with a graph structure consisting of a set of vertices and a set of edges. It is a technology solution to already associated data, like the text in our case, and it is instrumental in understanding big datasets and process connected data. This is also the reason for being used in supply chain management and e-commerce recommendations. Therefore, a visual representation of our data helps us to draw easier and faster conclusions.

This work aims to represent data from the text as an optimal graph to analyze them fur-

ther. This work's primary objective is to properly process the sentences, select preprocessing methods that will extract useless information from the text, and create a useful graph.

We used Neo4j, a graph database management system developed by Neo4j, Inc. Neo4j is described by its developers as an ACID-compliant transactional database with native graph storage and processing. Python programming language was used to develop the appropriate algorithm.

1.2 Structure of the Thesis

The rest of the thesis is structured as follows. A literature overview is presented in Section 2, giving the current thesis's research areas' status. Natural Language Processing and a historical overview is carried out in 2.1 and 2.2. Sections 2.3 and 2.4 refer to NLP's tasks and the approaches to solve them, and in the next subsections, there is an overview in Graphs and Databases. Section 3 consists of the technologies that are used for the development of the application. Section 4 presents the proposed methodology starting from text preprocessing and creating the tag patterns and finally, the graph that every rule creates. The evaluation of the application is presented in Section 5. The evaluation is carried out between three texts. The results and the final graphs that are created are the essential elements to verify the application. Finally, Section 6 contains the current thesis's conclusions and some plans regarding how to improve the current methodology

2 Literature Overview

2.1 Overview of Natural Language Processing

All In-apps messages (Whatsapp, telegram, etc.), social media, forums, blogs, google searches, and many other apps are constantly generating a large amount of text data every second. NLP can handle data of this size. When Alan Turing published an article called " Computing Machinery and Intelligence," many applications have been developed to store and process data from the start of NLP. NLP nowadays has changed the way people interact with computers. Some everyday examples are virtual assistants like Alexa or Siri. Natural Language Processing (NLP) is a vast area of Computer Science concerned with the interaction between Computers and Human Language. The primary goal of NLP is an automated understanding of the semi-structured language that humans use. The idea of natural language processing is to design and develop a computer system that can analyze, understand, and synthesize natural human languages. Natural language falls within the domain of artificial intelligence

2.2 History of NLP

2.2.1 1940-1960

The work of the first phase concentrated on machine translation (MT). In 1952 the first international conference on MT (the first artificial intelligence conference). The first phase's high point was the Teddington International Conference on Machine Translation of Languages and Applied Language Analysis held in 1961. At that point, there were not suitable higher-level languages, and programming was all in assembler. Access to machines was often limited; they had insignificant storage and were remarkably slow. In the '60s, with the most advanced algorithms and on the best available machines, processing time for a long sentence was 7 minutes | the greatest of the NPL research produced in this period. The research was focused on syntax, notably because syntactic processing was manifestly important.

2.2.2 Late 1960-late 1970

The second NLP phase operates as artificial intelligence (AI)-flavored, emphasizing word knowledge and its use in constructing and manipulating meaning representations. The shared spirit of those working in the field and the widespread view that progress could be and was being

made were apparent in the ARPA Speech Understanding Research (SUR). Unfortunately, some of the SUR projects were ambitious attempts to build genuine systems combining the top-down with the bottom-up processing; the results were not efficient. The above led to the development of modular architectures, toolkits, and general-purpose formalisms developed to supply specialized lexicon, semantics, and domain and database models on top of standard syntax sublanguage approach, which had been established for text processing and sometimes providing in specific syntax thoroughly.

2.2.3 Late 1970-late 1980

In the early 1980s, it was apparent that it was much harder to develop well-founded NLP systems even for heavily restricted applications than had been assumed. Those more challenging applications in processing tasks or claims a possible strategy for utilitarian MT are given enough effort. As the second phase of NLP is characterized as AI-flavoured and semantics-oriented, in a general sense of" semantic," the third phase can be described as the grammatical-logical phase. Computational grammar theory became a very active research area linked with logics for meaning and knowledge representation to achieve the user's expectations and intentions and with speech features and functions like theme and emphasis. Serious attempts were made to utilize popular dictionaries in machine-readable form. This led to performance using test corpora to validate or enhance existing lexical data and projects to define formalisms, e.g. exploiting feature structures, that can capture rich lexical information and permit its direct use, including inferential use formal processing context.

2.2.4 The 1990s

The lexicalist approach to grammar that appeared in the 1980s has become increasingly important. The lexicon has taken over much of what was formerly assigned to the syntactic component leaving the latter with few general rules. For instance, parsing with head phrase structure constitutes a compositional operation on large sections of constituent feature data, with analogous semantic rules addressing logical forms. Lexical tree adjoining grammars illustrate the same trend. Hidden Markov Modeling helped to improve the results in speech recognition. This concert with evaluation has been associated with a conspicuous growth of interest in the design and provision of linguistic resources, for example, the British National Corpus and WordNet and test tools like the Penn Treebank, which has annotations allowing parser performance assessment. Public domain processing tools, e.g., taggers, are also now

available, allowing rapid prototype system assembly using modular architectures.

2.2.5 The present

Most progress has been made in syntax, where we have sufficient means of grammar characterization and useful techniques like chart parsing. More generally, workers in the field now have a stock of conceptual tools, like typed feature structures or case and domain frames and enough experience of using them to put together a system or interface subsystem for many experimental or developmental purposes and even, for suitably restricted tasks or limited output expectations, for regular operational production. The main goal of any NLP application is to generate a parse tree for a sentence belonging to the set of that language. The major problem in NLP is ambiguity. Almost every phrase contains words that can be translated in different ways. Some words in the process of POS tagging can be classified as a verb or as a noun. For example, the word "can." The problem solved as the researchers create a rule-based software that examined the sentence structure and compared the sentence with a sentences prototypies[3]. As technology progresses, NLP will continue improving in combination with Machine Learning. The future of NLP is the Natural Language Generation (NLG). NLG is the technology that transforms data into natural language.

2.3 Natural Language Processing Tasks

The human language is everywhere as text and as speech but is also problematic; there are hundreds of languages and dialects. NLP applications have been created for two reasons: to understand and to generate human language. The most common tasks are:

Automatic summarization: The goal of automatic summarization is to take an information source, extract from it and present the actual content to the user in a condensed form and appropriate to the user's or application's needs.[4]

Machine Translation: Machine translation is one of the oldest subfields of artificial intelligence, and it automatically converts one language into another, preserving the meaning along the process. The recent empirical techniques have led to significant improvements in quality. [5].

Named entity recognition: The task of identifying named entities like a person, location, organization, drug, time, clinical procedure, biological protein. Named Entity Recognition (NER) is a crucial task in NLP systems. For decades NER systems have been studied, and now those systems provide question answering application and relation extraction.[6]

Relationship extraction: The task requires the detection and classification of semantic relationship mentions within a set of artifacts, typically from text or XML documents. The task is very similar to that of information extraction (IE), but IE additionally requires the removal of repeated relations (disambiguation) and generally refers to the extraction of many different relationships [7].

Sentiment analysis: refers to the computational study of people's opinions. It is focused on emotions and attitudes toward topics, issues, or products. In recent years with the explosive growth of social media, the task has become technically challenging and practically very useful. Businesses always search to find public or consumer opinions about their products and services. Besides, the opinions of existing users' are useful for potential customers. Sentiment analysis has a significant impact on decision-making for businesses and users.[8]

Speech recognition: is a subfield that enables the recognition and translation of spoken language into text. It is also known as speech to text, computer speech recognition, and speech recognition. It consists of methodologies and technologies from computer science and linguistics.

Topic segmentation: refers to the process of dividing the written text into meaningful units, such as words, sentences, or topics. Some languages have explicit word boundary markers, but other languages contain signals and shapes that are not present in all written languages. [6].

2.4 Approaches to solve NLP tasks

NLP's growing importance is unquestionably evident, there are billions of text data being generated every day, and at the same time, the wide range of applications have created different ways to approach NLP tasks. There are three main approaches to solve an NLP task based on how the data are processed:

1. Rule-based Rule-based approaches are the oldest approaches to NLP. Those approaches are applied to text and offer much insight, usually, they have good performance. Rule-based approaches:

    - usually are focused on pattern-matching or parsing
    - they use "fill in the blanks" methods
    - have are low precision, high recall (they have high performance in specific use cases) but often have low-performance whet they generalized

2. "Traditional" Machine Learning approaches include probabilistic modeling and linear classifiers. Traditional machine learning approaches are defined by:

    - training data
    - feature engineering
    - training a model
    - inference (applying model to test data)
    - "semantic slot filling".

3. Neural Networks They are comparable to "traditional" machine learning, but with some remarkable differences:

    - Networks are being trained so feature engineering is generally skipped.
    - Neural Networks are fed from streams of raw parameters (actually vector representations of words) without engineered features
    - vast training corpus [6].

2.5 Text Preprocessing

NLP problems usually come from internet data such as blogs and social media. Those data are not in formats that can yield valuable results if they are not processed. Preprocessing is the process by which text is converted into something an algorithm can digest. There are several preprocessing methods and based on the original form of the text and the result desired, the appropriate methods are selected each time. Preprocessing is an essential step in NLP. There are four main stages:

- Cleaning consists of discarding a text's less useful parts through stopword removal, dealing with capitalization and characters, and other details.
- Annotation refers to the application of a scheme to text. Annotations sometimes contain structural markup and part-of-speech tagging.
- Normalization consists of Stemming, Lemmatization, and other forms of standardization that translate terms in the scheme or carries out linguistic reductions.
- The analysis consists of statistically probing, manipulating, and generalizing from the dataset for feature analysis.

The following are some important approaches to preprocessing:

- Tokenization

  Tokenization is the method of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens. The aim of the tokenization is the exploration of the words in a sentence. The list of tokens becomes the input for further processing, such as parsing or text mining. Tokenization is useful in linguistics (where it is a form of text segmentation) and in computer science, where it produces part of lexical analysis. Textual data is only a block of characters at the beginning. All processes in information retrieval require the words of the data set. Hence, the requirement for a parser is the tokenization of documents. The primary use of tokenization is identifying meaningful keywords.

- Lower Casing

  A common approach is reducing all letters to lower case. Usually, this approach performs satisfactorily, mostly when the capital letter is used at the beginning of the sentence. On the other hand, the preservation of uppercase letters is necessary, as they refer to names, companies, or organizations and the case folding strategy creates confusion.

- Stemming

  Stemming is the process of reducing inflection in words (e.g. troubled, troubles) to their root form (e.g. trouble). The "root" in this case may not be a real root word, but just a canonical form of the original word.

- Lemmatization

  Lemmatization on the surface is significantly related to stemming, but it transforms words to the actual root. For example, the word" better" would map to" good."

- Stop-words removal Many words in texts repeat very frequently but are meaningless as they are used to join words together in sentences. It is widely accepted that these kinds of words, stop words, do not contribute to textual documents' context or content. The intuition behind using stop words is that, by removing common information words from a text, we can focus on the essential words instead.

- Normalization

  Text normalization transforms the words of the text into a canonical form. For example, the word" gooood" and" gud" can be modified to" good".

- Noise Removal Noise removal refers to removing characters, digits, and text pieces that can interfere with text analysis. Noise removal is one of the essential text preprocessing steps. It is also highly domain-dependent.

- Text Enrichment/Augment Text enrichment involves augmenting original text data with information that it is not already contained in the text. Text enrichment provides more semantics to the original text, thereby improving its predictive power and the depth of analysis that can perform on data, like Part of Speech Tagging. [7][8].

2.5.1 The significance of preprocessing

Preprocessing is an essential procedure in NLP as the right methods can improve the final results. Preprocessing reduces the text document's indexing size by stopwords and stemming and improves the IR system's efficiency and effectiveness. There are many preprocessing methods, so it is challenging to export the right information in each specific case. The choice of the right methods and techniques can increase the value of the final results.

2.6 Graphs

A graph is a pictorial representation of a set of objects where links connect some pairs of objects. The connected objects are represented by points termed as vertices, and the links that connect the vertices are called edges. Formally, a graph is a pair of sets (V, E), where V is the set of vertices and E is the set of edges, connecting the pairs of vertices [9].

2.6.1 Root of graph Analytics

The solve of the " Seven Bridges of Königsberg" problem by Leonhard Euler in 1736 was Graph analytics's birth. The problem questioned whether it was possible to visit four areas of a city, connected by seven bridges, while only crossing each bridge only once. Such a thing was not feasible, Euler with the insight that only the connections themselves were relevant, set the groundwork for graph theory and its mathematics [9]. The first graph textbook was published in 1936, two hundred years later. In the late 1960s and 1970s, network science and applied

graph analytics began to emerge. In the last few years, there has been an explosion of interest in and usage of graph technologies.

Graph-powered recommendation engines support firms personalize products or services by contextualizing many connections. Relevant real-time recommendations require the connecting product, customer, historical decisions, supplier, logistics, and even social sentiment data. Furthermore, a real-time recommendation engine requires the ability to instantly capture any new interests shown from the customer's current visit. As businesses have become more customer-centric, it is urgent to use data connections to make well-timed decisions. Those decisions require technology to unite data from different sources such as customers, suppliers, and logistics information. Graphs are supporting a wide variety of artificial intelligence (AI) cases. This analysis surfaces relationships and presents a richer and more in-depth context for prescriptive analytics and AI uses like TextRank. Besides Graph analysis underpinning natural language processing (NLP) and natural language understanding (NLU) technologies. A knowledge graph representation helps an application connect words in the context in which the words are used.

2.6.2 Graph Properties

There are some fundamental properties of graphs referred to the graph traversal, and they are affected by the algorithm that creates them. The figure 1 present different types of graphs based on the interconnectivity and their overall structure.

- Undirected vs. Directed. The term undirected graph refers to a graph that there are no directions to the relationships between nodes. n the contrary directed graph relationships have one specific direction. The direction is not always crucial for constructing a graph or the analysis that will be performed. The direction is relevant to Community Detection algorithms, mostly Weakly and Strongly Connected Components.
- Cyclic vs. Acyclic. In graph theory, cycles are paths through relationships and nodes where there is a movement from one node to a particular node. There are many types of cycles within graphs. Cycles require attention when using algorithms that may create infinite loops. A common type of connected and acyclic (and undirected) graph is a tree structure.
- Weighted vs. Unweighted. Weighted graphs indicate values (weights) to either the

nodes or their relationships. Pathfinding algorithms usually use weighted graphs . [10].

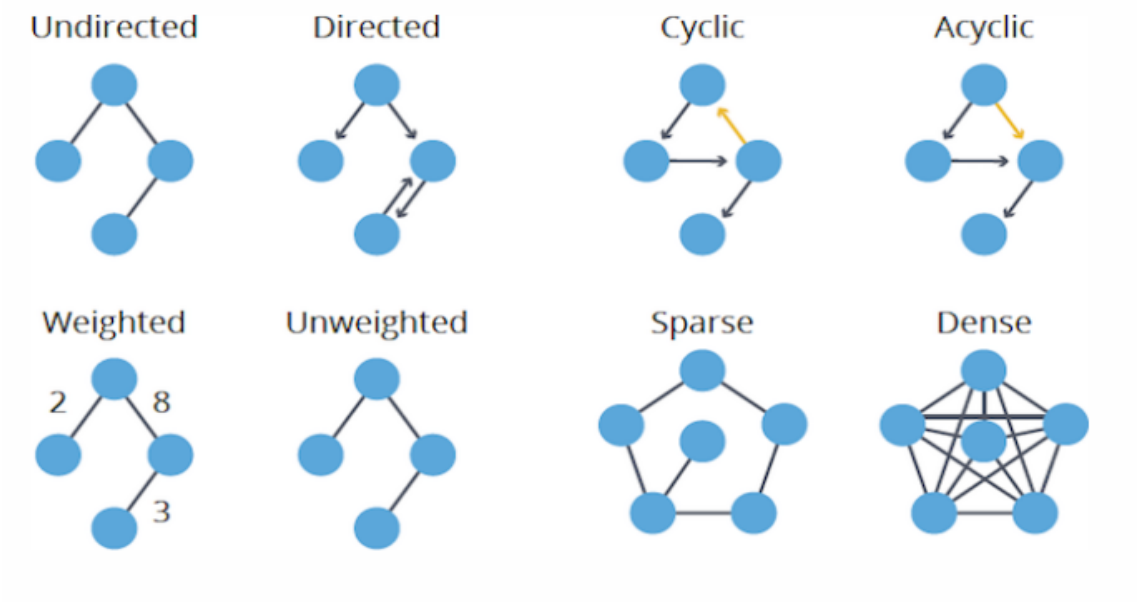

Figure 1: Types of Graphs. source: Neo4j

Graphs arise in many fields, including sociology, biology, computer science, and engineering. Graphs also play an essential role in linguistics and are handy for representing semantic knowledge. Moreover, creating a graph that models human language as a graph of interacting words has shed light on the nature of language, particularly on how words may be organized in the brain and how language evolves [11]. Graphs and Natural Language Processing (NLP) are linked. It is obvious as the words in a sentence are connected through the syntactic relations and semantically connected. ut when the techniques for Language Processing are involved, the connection between graphs and those techniques is hard to find. Graphs are used for many NLP applications, such as correcting typos and machine translation. Graphs are a powerful representation of connections and solve linguist issues. [21].

2.7 Databases

A data [base] model (db-model) is a concept that describes a collection of conceptual tools for representing real-world entities to be modeled and the relationships among these entities. Data an be store, organized and manipulated in differents logical struc-

tures the type of database depending upon the usage requirements. There are 11 types Centralised,Distributed.,Personal, End-user database, Commercial, NoSQL, Operational, Relational, Cloud, Object-oriented, Graph.Popular database models and their management systems include:

- Relational database management system (RDBMS) – adaptable to most use cases, but RDBMS Tier-1 products can be quite expensive.
- NoSQL DBMS – well-suited for loosely defined data structures that may evolve over time.
- In-memory database management system (IMDBMS) – provides faster response times and better performance.
- Columnar database management system (CDBMS) – well-suited for data warehouses that have a large number of similar data items.
- Cloud-based database management system – the cloud service provider is responsible for providing and maintaining the DBMS [12].

2.8 Graph Databases

Graph database models can be defined as those in which data structures for the schema and instances are modeled as graphs or generalizations of them, and data manipulation is expressed by graph-oriented operations and type constructors. These models took off in the eighties and early nineties alongside object oriented models [13].

As mentioned before, the graph is a pictorial illustration of a set of objects where links connect pairs of objects. Points are termed as vertices, and edges are the links that connect them. Graph databases are handling complex and flexible data models. Besides, that kind of database process highly connected data and while at the same time improve complex queries' performance by traversing the graph. The simplicity of its model is another characteristic of its quality.[9].

The fact that data and relations among them are at the same level drive simplicity of the model and ease of understanding. There are several advantages as a modeling tool for this type of data. Through graphs, users can see the relationship between nodes and allow better handling of application data. Another advantage of the graphs is that they

can keep information about a node and provide more information about the relationships connected to nodes.

In addition, queries can refer right to the graph structure. Finding shortest paths, determining certain subgraphs, and so forth is associated with the graphs in specific language algebra operations. Detailed graphs and graph operations enable users to express a query at a high level of abstraction. On the other hand, in deductive databases, usually, complex rule programs need to extract the information needed. It is not essential to require full awareness of the structure to formulate meaningful queries. Lastly, for browsing, it may be beneficial to ignore the schema. As far as implementation is involved, graph databases may present efficient algorithms for achieving specific operations and significant structures for the storage of data[13].

3 Technologies

3.1 NLTK

Natural Language Toolkit (NLTK) is a suite of open-source programs. NLTK affords ready-to-use computational linguistics courseware and also provides tutorials, modules, and problem sets. Also, this suite includes symbolic and statistical natural language processing and is interfaced with annotated corpora.

The toolkit is displayed as a set of independent modules, defining a specific data structure or task. A set of core modules defines the primary data types and processing systems used throughout the toolkit. The token module provides basic classes for processing individual elements of text, such as words or sentences. The tree module defines data structures for representing tree structures over text, such as syntax trees and morphological trees. The probability module implements classes that encode frequency distributions and probability distributions, including various statistical techniques. In NLTK, there are:

- Parsing Modules The parser module defines a high-level interface for producing trees that represent the structures of texts. The chunkparser module defines a sub-interface for parsers that identify nonoverlapping linguistic groups (such as base noun phrases) in unrestricted text.
- Tagging Modules The tagger module represents a standard interface for augmenting each token of a text with additional information, such as part of speech.
- Finite State Automata The FSA module represents a data type for encoding finite state automata; and an interface for performing automata regular expressions.
- Type Checking Debugging time is an essential factor in the NLTK's ease of use. All data types and processing classes use a type-checking module. It is necessary to achieve higher performance.
- Visualization Visualization modules represent graphical interfaces for viewing and managing data structures and graphical tools for studying NLP tasks. The visu-

alization modules implement interaction and experimentation; NLP is not directly performed to data tasks.

- Text Classification The classifier module illustrates a standard interface for classifying texts [22].

3.2 Neo4j

Neo4j is a property graph model. Various approaches construct a graph database's key components, as happens with most technologies. An approach is the property graph model. Data is designed as nodes, relationships, and properties (data stored on the nodes or relationships). As shown in 2, both relationships and nodes have properties, and the graph presents all the valuable information.

- Nodes are the entities in the graph. They contain several attributes (key-value pairs) called properties. Nodes can be marked with labels, describing their several purposes in any domain.
- Relationships present directed, named, semantically-relevant connections between two nodes. A relationship regularly is between a start node and an end node, has a direction and a type. Like nodes, Relationships, just like nodes, can have properties. Neo4j is a property graph model. Various approaches construct a graph database's key components, as with most technologies. In the majority of cases, relationships have quantitative properties. Costs, distances, ratings, weights, time intervals, or strengths can be used as properties. The efficient way that relationships are stored in a graph database, nodes can share several relationships without reducing performance. Relationships are stored in a specific direction; however, they can always be navigated efficiently in either direction.
- Properties are name-value pairs that are used to add qualities to nodes and relationships.

Neo4j is an open-source graph database management system. In big data, valence is the tendency of individual data to connect and the databases' overall connectedness. A graph platform like Neo4j offers an efficient means for data scientists and solutions teams to move through the discovery and design stages. Neo4j is a directed acyclic graph (DAG) data structure. Neo4j is a native graph database because it efficiently

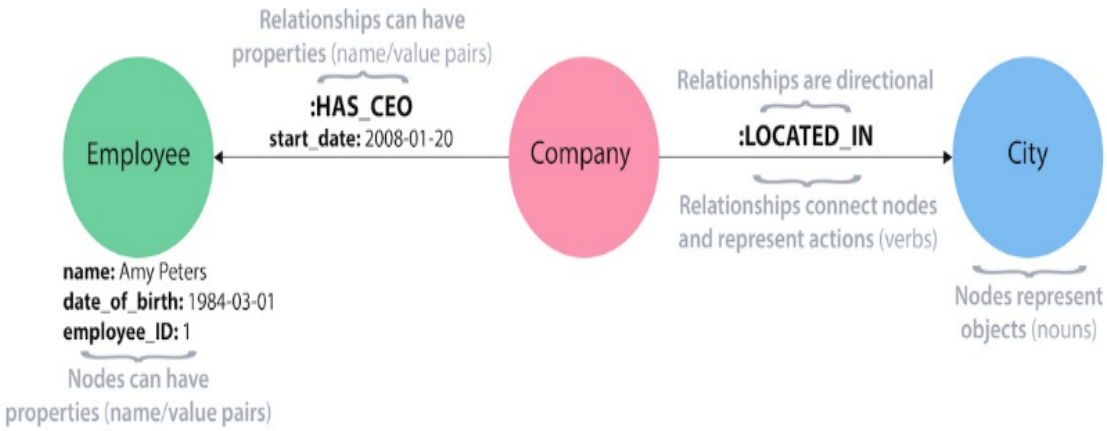

Figure 2: A graph model in Neo4j. source:neo4j.com

implements the property graph model down to the storage level. The database uses pointers to navigate and traverse into the graph. Unlike graph processing or in-memory libraries, Neo4j provides full database features, including ACID transaction compliance, cluster support, and runtime failover { securing that graphs can analyze data in product scenarios. The most common use examples powered by Neo4j are Fraud Detection and Knowledge Graph, Network and Database Infrastructure Monitoring IT Operations. Also is being used as Recommendation Engine and Product Recommendation System, in Social Media and Social Network Graphs, Master Data Management Privacy, Identity, and Access Management, Risk and Compliance and as is evident in Analytic [14].

3.2.1 Cypher

Cypher is a language provided by Neo4j. In particular, Cypher is a declarative graph query language inspired by SQL that attempts to avoid the need to write traversals in code. Cypher is still under intensive development, with support recently being added for graph manipulation instead of just query ability. Cypher attempts to use more of a keyword system, like SQL. Cypher obtained some of the list semantic from languages such as Haskell and Python. This similarity is a weakness compared to the more mature RDBMS SQL. There is a need for consistency that requires all implementations before understanding what approach is most suitable to the problem [15].

Cypher is a very human-readable language, and it is accessible not just for developers. Everyone can quickly learn and use it. Also, a declarative language allows users to state

what actions they want to be performed (such as match, insert, update or delete) upon their graph data without requiring them to describe (or program) exactly how to do it. Cyphers' expressions are similar to SQL like WHERE, ORDER BY, and a language used by a graph database has expressions to represent graph data patterns [16].

Examples in Cypher:

CREATE

- NODE is used to create node or relationship between 2 nodes
- RELATIONSHIP first create clause create two nodes and then creates the relationship between them.

e.g. Create (n)

MERGE

The MERGE keyword ensures that the supplied pattern exists in the graph, either by reusing existing nodes and relationships that match the supplied predicates, or by creating new nodes and relationships.As shown in figure 3 MERGE create two nodes.

e.g.

Merge (n:Person name: 'We')

Merge (n:Person name: 'application', title: 'faster')

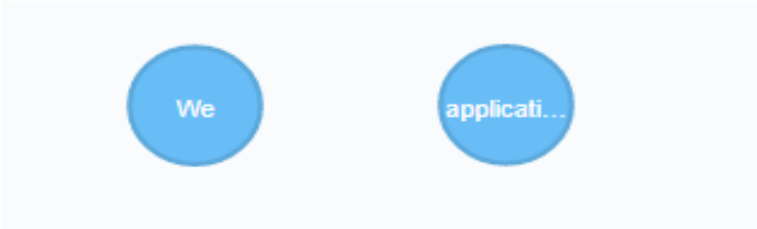

Figure 3: Example MERGE

MATCH

The MATCH keyword in Cypher searches and finds an existing node, relationship, label, property, or pattern in the database. MATCH works like SELECT in SQL. Figure 4 shows the relationship that MATCH created between the two nodes.

e.g. MATCH (a:Person),(b:Person) WHERE a.name = 'We'AND b.name = 'application 'CREATE (a)-[r:developed]->(b)

Figure 4: Example MATCH

RETURN

The RETURN keyword in Cypher specifies what values or results you might want to return from a Cypher query. With the RETURN Cypher returns nodes, relationships, nodes and relationship properties, or patterns from the results. RETURN is needed for reads.The node and relationship variables become important when using RETURN. In order to bring back nodes, relationships, properties, or patterns, they need to have variables specified with MATCH clause to return.

e.g. MATCH (a:Person),(b:Person) WHERE a.name = 'We'AND b.name = 'application 'CREATE (a)-[r:developed]->(b) RETURN type(r)

3.3 Py2neo

Py2neo is a client library and toolkit for working with Neo4j from within Python applications and from the command line. To create the Neo4j database, we developed a python code. This code uses the py2neo library to access the Neo4j database, and it reads our data (external source) to create nodes, relationships, properties, and indexes [17].

3.4 Python

Python is an interpreted, high-level, general-purpose programming language. The first release was in 1991 from the creator Guido van Rossum. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach strive to help users write exact, logical code for small or large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Due to its comprehensive standard library, Python is often described as a "batteries included" language. Python was formulated in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, included list comprehensions and a garbage collection system to collect reference cycles. The next version Python 3.0, in 2008, was a significant revision of the language that is not completely backward-compatible. [7].

3.4.1 Pycharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. The Czech company JetBrains developed this environment, and it provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development and Data Science with Anaconda cite7.

4 Methodology

This proposed methodology aims to export a semi-structured form from unstructured text. This purpose is to make explicit the relationships specifically, that is, the interactions between the entities of a sentence and the properties that describe them. Graphs are a way of representing correlated data and represent complex relationships as they exist in a sentence. To be converted into a graph, syntax rules were created. The rules try to match the basic elements, such as an object, subject, and the verb, to the graph's corresponding elements. Before applying them, it is necessary to preprocess the text to eliminate unnecessary information, following the rule matching, generating the Cypher code, linking the application to the neo4j database, and ultimately creating the graph. The methods of preprocessing that will be used are the Tokenize method, Stop word removal, and Part-of-speech-tagging. Figure 5 represents the proposed methodology step by step.

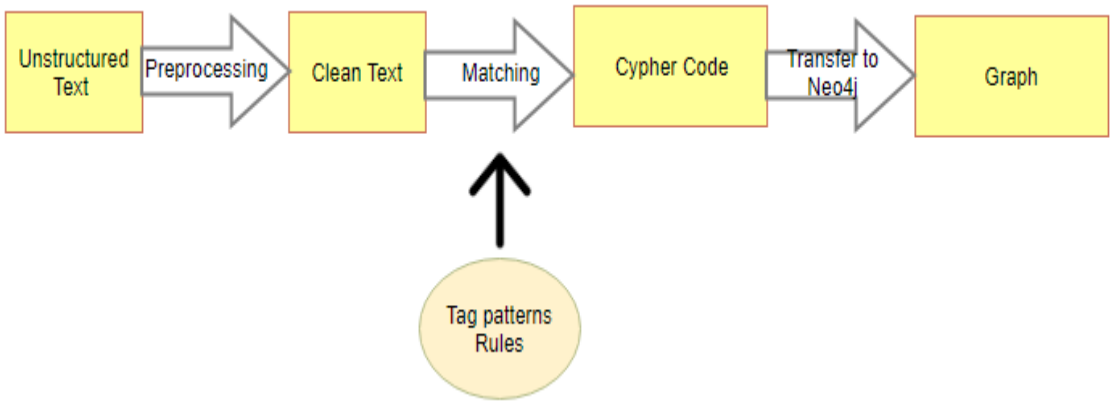

Figure 5: Methodology flow chart

4.1 Text Preprocessing

Text Preprocessing is crucial in any text analysis, as it directly impacts the project's success rate. In this project, based on rules, the text preprocessing is critical, so the valuable properties utilized from the text.he combination of preprocessing methods were selected based on the final aim. Each of these methods is important for the proper efficiency of the application.

4.1.1 Tokenization

Tokenization is an integral part of virtually every NLP task. It is the process of splitting a string into a list of pieces or tokens; a token is a piece of a whole. In a sentence, the token is a word, and in a paragraph sentence is a token. The primary use of tokenization is to identify meaningful keywords. The disadvantage of tokenization is difficult to tokenize the document without any whitespace, special characters, or other marks. Tokenizing at the word level is perhaps the most common and widely used tokenization.

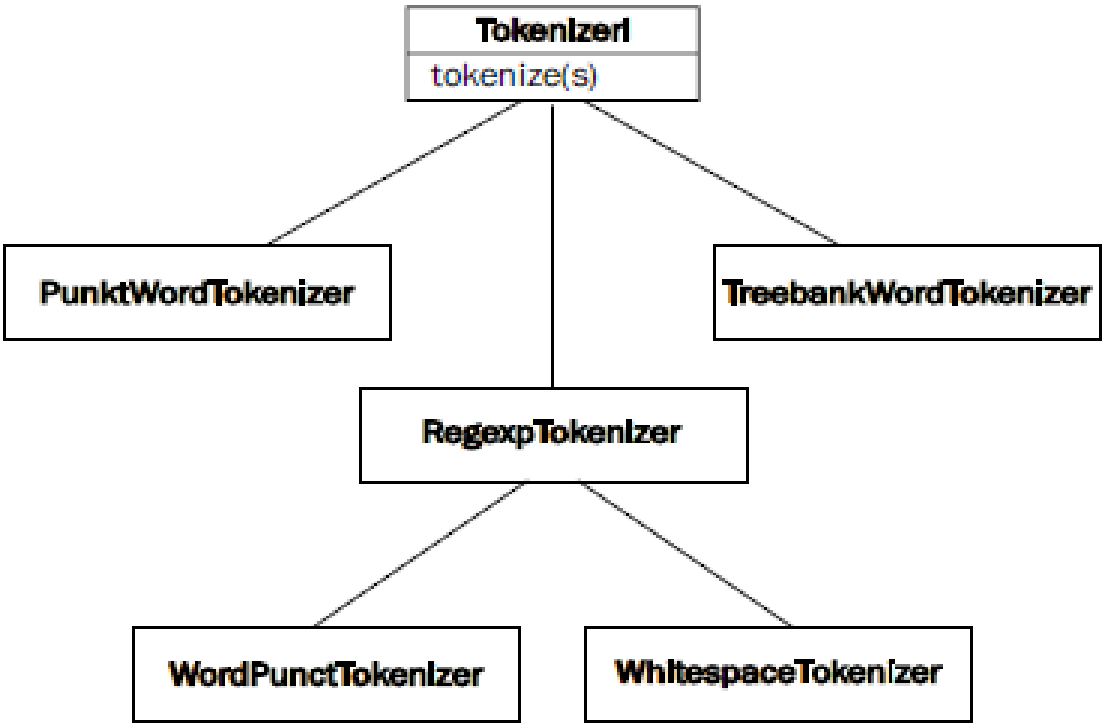

Figure 6: The inheritance tree of Tokenizer1, source:Python 3 Text processing with NLTK 3 Cookbook.

The various tokenizers can be seen in figure 6.In particular, this approach uses PunktWordTokenizer. PunktWordTokenizer it is an alternative word tokenizer. It splits on punctuation but keeps it with the word instead of creating separate tokens. The PunktSentenceTokenizer class uses an unsupervised learning algorithm. The list of tokens becomes the input for the next processing, stop words removal.

4.1.2 Stop Word Removal

Stop word removal is an essential preprocessing step for some NLP projects. Stopwords are common words that generally do not contribute to a sentence's meaning, at least for information retrieval and natural language processing. These are words such as 'the" and" a". Most search engines will filter out stopwords from search queries and documents to save space in their index [18].

Stop words accounts for 20-30% total words in a particular text document, and stopwords removal reducing the data set size witch is a way to increasing performance. In our case, the most common words have no further information for the final model. In some cases, stop word removal is ineffective for the final result like sentiment analysis. The table below shows the stop words used in this application.

Table 1: Stop Words

i,me,myself	they,them,their, theirs,themselves	do,does, did,doing
our,ours,ourselves	what,which,who,whom	a,an,the
you,yours, yourself, yourselves	this,that,these,those	and,but,if,or,because
he,his,himself	am,is,are,was,were	as,until,while
she,her,hers	be,been,being	of,at,by,with,about
it,its,itself	have,has,had,having	against,between,into, through
during,before,after	above,below,up, down,in,out, to,from,on,off, over,under	against,further
then,once	here,there	when,where,why,how
all,any,both,each, few,more	other,some,such	no,nor,not
only	so,than,too,very	own,same
can,will,just, do,should	now	

It is important to note that those words are being removed from the text when they are not the first word of the sentence. Example of stopwords removal:

input text(15 words):

People will disagree with the idea that a kind action is good for the soul.

after stopwords(7 words):

People disagree idea kind action good soul.

4.1.3 Part of Speech Tagging

Part-of-speech tagging is the process of specifying a part-of-speech marker to each part-of-speech tagging word of an input text. The input to a tagging algorithm is a sequence of (tokenized) words and a tagset, and the output is a sequence of tags, one per word. There are many parts-of-speech lists; most advanced language processing in English uses the 45-tag Penn Treebank tagset. Part-of-speech is valuable because of the vast amount of information it provides about a word and its neighbors. Knowing whether a word is a noun or a verb shows a lot about its neighboring words (nouns are preceded by determiners and adjectives, verbs by nouns) and about the syntactic structure around the world (nouns are generally part of noun phrases), which makes part-of-speech tagging a critical component of syntactic parsing [19]. Example of Part of Speech Tagging:

Input text: 'People will disagree with the idea that a kind action is good for the soul.'

Post text : [('People', 'NNS'), ('disagree', 'VBP'), ('idea', 'NN'), ('kind', 'NN'), ('action', 'NN'), ('good', 'JJ'), ('soul', 'NN'), ('.', '.')

It is apparent from figure 7 that Penn tree bank have many tags, as it divides each category into subcategories even at the punctuation marks. This gives as much information as possible about the part of speech of a word.

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	$
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	#
JJS	superlative adj	*wildest*	SYM	symbol	+,%, &	"	left quote	' or "
LS	list item marker	*1, 2, One*	TO	"to"	*to*	"	right quote	' or "
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure 7: Part Of Speech Tagging Penn tree bank, source: www.researchgate.net

4.2 Tag Patterns

4.2.1 Preprocessing of the Tag Patterns

At the final step of preprocessing of text, it is necessary to separate the part of the sentence we are interested in and store it. Chunking is the method that is being used. Chunking is the process of extracting phrases from unstructured text. The nine tag patterns that have been created by nltk.RegexpParser.

A grammar-based chunk parser, "chunk.RegexpParser," uses a set of regular expression patterns to specify the parser's behavior. Chunking of the text is encoded using a ChunkString, and each rule acts by modifying the chunking in the ChunkString. The rules are all implemented using regular expression matching and substitution. The rules that created use tag patterns to describe sequences of tagged words [20].

Chunk structures can be represented using either tags or trees. A parse tree or parsing tree or derivation tree, or concrete syntax tree is an ordered, rooted tree representing the syntactic structure according to some context-free grammar. The term parse tree itself

is used primarily in computational linguistics; in theoretical syntax, the term syntax tree is more common [7].

4.2.2 Tag Patterns

A tag pattern is a string consisting of angle-bracket delimited tag names. An example of a tag pattern is:

<DT><JJ><NN><VBD><DT><NN>

Rules are defined in terms of regular expression patterns over "tag strings." Most rules are defined using a special kind of regular expression pattern, called a tag pattern. Tag patterns are identical to regular expression patterns in most respects; however, there are a few differences, which are intended to simplify their use with REChunkParsers:

- In tag patterns, "<" and ">" act like grouping parentheses; so the tag pattern "<NN>+" matches one or more repetitions of "<NN>"; and "<NN|JJ>" matches "<NN>" or "<JJ>."
- Whitespace in tag patterns is ignored; so "<DT> | <NN>" is equivalent to "<DT>|<NN>". This allows you to make your tag patterns easier to read, by inserting whitespace in appropriate places.
- In tag patterns "." is equivalent to "[<>]"; so "<NN.*>" matches any single tag starting with "NN."

The rules that make up a chunk of grammar use tag patterns to describe sequences of tagged words. A tag pattern is a sequence of part-of-speech tags delimited using angle brackets, e.g. <DT>?<JJ.*>*<NN.*>+. This given example will chunk any sequence of tokens beginning with an optional determiner (?), followed by zero or more adjectives of any type (including relative adjectives like earlier/JJR), and this is stated with the (*) in the angle bracket of JJ, followed by one or more (+) nouns of any type [25]. Also, when there is no symbol between angle brackets, the sequence of the part-of-speech tags is strict. e.g. <NN.*><VB.*><NN.*><.> This example will chunk any sequence of tokens with one Noun of any kind (NN.*), one Verb of any kind (VB.*), again, one Noun of any

kind sentence ends there. Table 2 shows the punctuation marks used to create the tag pattern in the current application.

Table 2: Symbols that used in Tag Patterns of this application.

Symbols	Example of use	interpretation
(?)	<DT>?<NN>	optional (DT) before (NN)
(*)	<JJ>*<NN>	zero or more (JJ) before (NN)
(+)	<NN>+<NN>	one or more (NN) before (NN)
	<NN><NN>	one and only (NN) before (NN)

Table 3: Tags that used in tag patterns of this application.

Tag	Description
JJ,JJR,JJS	Adjective,Adjective comparative, Adjective superlative
NN,NNS,NNP,NNPS	Noun, Noun plural, proper noun sing. and plural
VB,VBD,VBG,VBN, VBP,VBZ	verb base form,verb past tense,verb gerund,verb past participle,verb non-3sg pres
PRP	Personal pronoun
RB,RBR,RBS,RBS	Adverb,adverb comparative,adverb superlative
(.)	sentence final punc

The Tables 2,3 above represent the symbols and the tags of part of speech that used in the development of this application those elements create the tag patterns that used as rules.

4.2.3 Tag patterns as Rules

Syntactic Rules and Tag patterns

The syntax is the grammatical structure of words and phrases to create coherent sentences. There are some different types of syntax in English sentences. The simple sentence that follows the subject-verb format, the compound sentence with more than one subject or verb, the complex sentence containing a subordinating clause, a complex sentence containing a subordinating clause, and a compound-complex sentence containing two independent clauses and more dependent clauses. The rules have been created to respond in most of the simple independent clauses. he critical elements of the sentence are the verb, subject, and object, while more elements define them, such as Adjectives. A canonical clause consists of a subject followed by a predicate. A verb phrase realizes the predicate; the subject is mostly realized by a noun phrase.he rules were created based on the sequence of parts of speech used in these sentences. ccording to this, the first rules created correspond to the most simple sentences that consist of a verb, a subject, and an object, so the tag pattern of this sentence is <NN.?><VB.?><NN.?>. n the next step, the rule was enriched with adjectives identifying either the subject or the object. Wherefore adjectives are essential information and are easily recognized in the text; there are rules for the adjective and noun sequence only. Finally, pronouns and adverbs are used in rules. Personal pronouns are pronouns used to represent people or things. In a sentence, personal pronouns can be used as an object or as a subject. An adverb is a word or an expression that modifies a verb, adjective, determiner, clause, preposition, or sentence. In this case, personal pronouns are used only as a subject, and adverb modifies the verb. The rules that have been created referred to the text after the stop word removal. Besides, some rules represent part of the sentence and not the entire sentence. Based on which rule the sequence belongs, the part of speech, and the interactions between them, every word of the tag pattern allotted to a node, a relationship, or a graph's property.

4.3 Matching Text to Rules

Matching is the procedure by which the sequence of part of the speech tagging of a sentence corresponds to each of the nine tag patterns-rules been created for this application.

NP1

Tag pattern 1 : "NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>"

This tag pattern is a sequence of one or more adverbs and a noun.It is referred to a sentence or part of the sentence.

input_text : This is a big, beautiful, fluffy cat.

This is a simple independent clause that consists of one verb and a subject. The verb is a linking verb that connects the subject with words that gives information about the subject. Linking verbs do not show any action; they simply link the subject with the rest of the sentence. Those verbs are removed by the stop word preprocessing method. So, in particular," cat" is the subject (NN.?) and" big," beautiful," and" fluffy" are the properties (JJ.?). The aim is to create one node for the subject" cat" with properties "big, beautiful, fluffy." As shown in 8 NP1 is the root of the tree, and the words that correspond to the rule are the leaves.

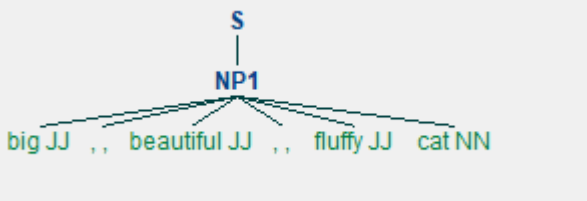

Figure 8: NP1 Tree.

Python Code of NP1 rule.

```
for subtree in x.subtrees(filter=lambda t: t.label().endswith("NP1")):

        cypher = graph.cypher
        list= []
        for index, entity in enumerate(subtree):
                if entity[1] == 'NNS' or entity[1] == "NN":
                    str1 = "Merge (n:Person {name:'"
                    str1 += entity[0]
                    print(entity[1])
                else:
                    list.append(entity[0])

        str1 +="',title :'"
        if len(list) >= 2:
         for k in list :
            str1 += k
            str1 += ","
        else:
            for k in list :
             str1 += k
        str1 += "'})"
        cypher = graph.cypher
        print(str1)
        print("NP1", str1)
        cypher.execute(str1)
```

NP2

Tag pattern 2 : <NN.?><NN.?>

This tag pattern is a sequence of a noun and a noun. It is referred to a sentence or part of the sentence. This tag pattern is about the noun phrases that are very common in the English language. Most of the time, that rule is used in a part of a sentence.

input_text : "The chocolate cake is here. "

This is a simple independent clause that consists of one verb and a subject. The subject in this sentence is the noun phrase; it is a linking verb that connects the subject with a word that gives information about the subject. Linking verbs do not show any action; they simply link the subject with the rest of the sentence. Those verbs are removed by the stop word preprocessing method. In this sentence," chocolate cake" is the subject, which is a sequence of two nouns. The aim is to create one node for the subject" cake" (NN.?) with the property" chocolate" (NN.?). Figure 9 shows the tree that NP2 creates.

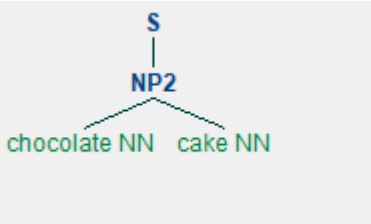

Figure 9: NP2 Tree

NP3

Tag Pattern 3: <NN.?><VB.?><NN.?><.>

This tag pattern is a sequence of a noun, a verb and a noun.This is the most common syntax in the English language and it is referred to a sentence or part of the sentence.

input_text: "Tom owns a car. "

This is a simple independent clause that consists of one verb, a subject, and an object. The subject in this sentence is the first Noun, the verb shows the action, so it is the main verb, and the object is the second Noun. The tag patterns that contain verb <VB.?> are the one that creates a relationship in the final graph. So in this sentence," Tom" is the subject," owns" is the verb, and" car" is the object. The aim is to create one node for the subject" Tom" (NN.?), one node for the object "car," and a relationship with the type" owns" that connects the two nodes. Figure 10 represent the NP3 Tree.

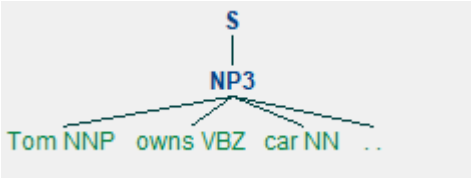

Figure 10: NP3 Tree

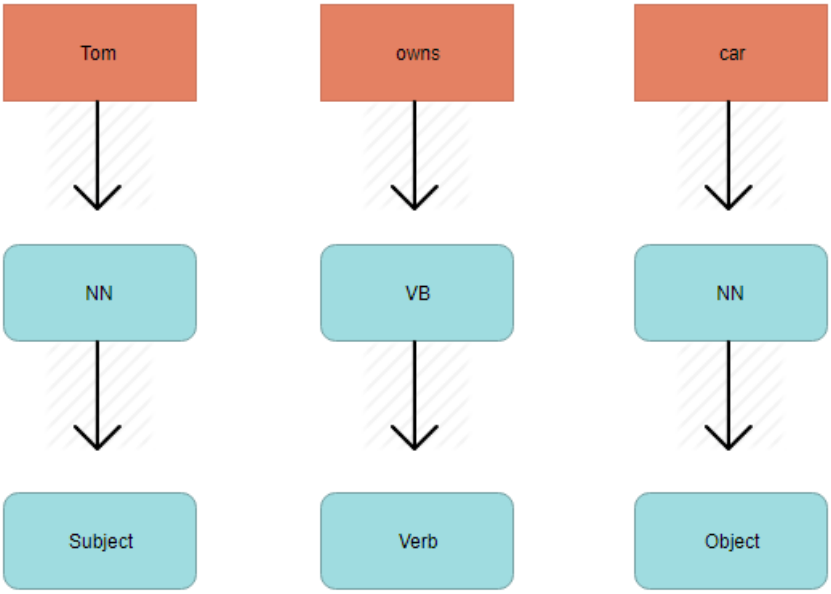

Figure 11: Matching of the sentence with NP3.

Figure 11 indicates the matching of the words to the part of speech tag and the correspondence to the syntactic structure.

NP4

Tag pattern 4 : <NN.?><VB.?><JJ.?><NN.?><.>

This tag pattern is a sequence of a noun, a verb, an adjective, and a noun. It is similar to the previous rule, but there is the adjective <JJ.?> as a <NN.?>. Further, it is obvious that this tag pattern contains another previous tag pattern <JJ.?><NN.?>.

input_text :" Shara bakes delicious cookies."

This is a simple independent clause that consists of one verb, a subject, an adverb, and an object. The subject in this sentence is the first Noun; the verb shows the action, so it is the main verb, and the object is the second Noun. The tag patterns that contain verb <VB.?> are the one that creates a relationship and also the <JJ.?> gives property to a noun-node in the final graph. In this sentence," Shara" is the subject (NN.?)," bakes" is the verb (VB.?)," delicious" (JJ.?) is the property of the object" cookies" (NN.?). The aim is to create one node for the subject" Shara", one node for the object" cookies" with property" delicious," and the verb" bakes" as the relationship type that connect the two nodes. Figure 12 represent the two subtrees that the sentence contains, the NP1 and the NP4.

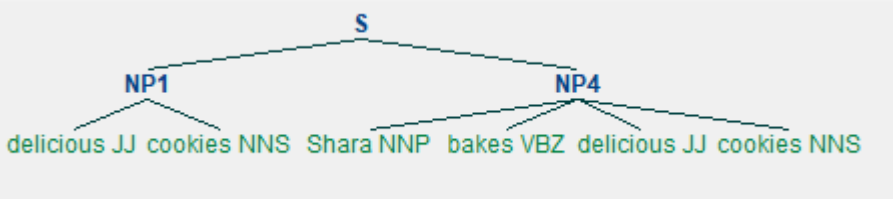

Figure 12: NP4 Tree.

Figure 13 indicates the matching of the words to the part Of speech tag and the correspondence to the syntactic structure.

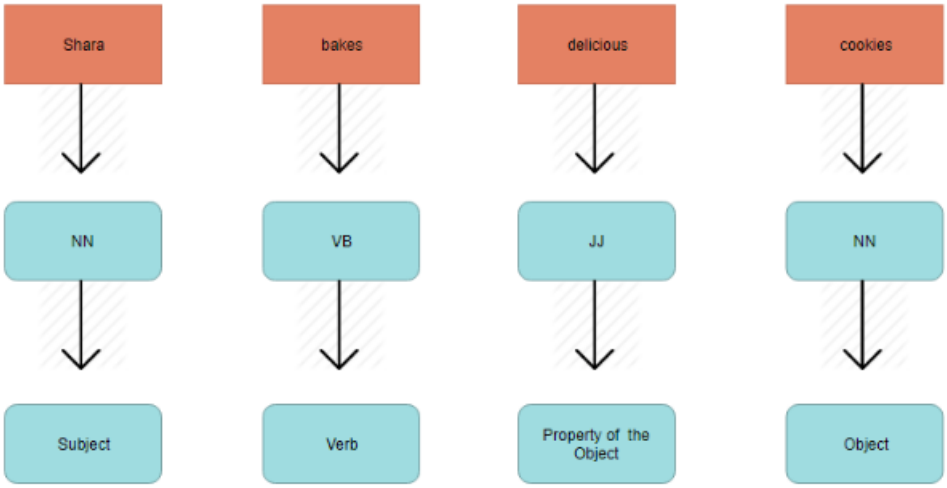

Figure 13: Matching of the sentence with NP4.

NP5

Tag pattern 5: <PRP><VB.?><RB.?><NN.?><.>

This tag pattern is a personal pronoun, a verb, an adjective, and a noun. A personal pronoun takes the place of a noun, and like nouns, pronouns function as subjects or objects in sentences. Pronouns are divided into five categories there is the subject pronouns (I), the object pronouns (Me), the possessive pronouns (My), possessive adjectives (Mine), reflexive pronouns (Myself); in these rules, only the subject pronouns are used.

input_text : I walk quickly to the town.

This is an independent clause that consists of a verb, a subject, an adverb, and an object. The subject in this sentence is the personal pronoun; the verb shows the action, so it is the main verb; the adverb gives property to the verb, and the Noun is the object. So in particular," I" is the subject (PRP), 'walk" is the verb (VB.?)," quickly" (RB.?) is a description of the verb" walk" and" town" (NN.?) is the object. The aim is to create one node for the subject" I," one node for the object" town," and the verb" walk" as the relationship type with property" quickly" that connect the two nodes.Figure 14 represent the NP5 tree. The figure 15 indicates the matching of the words to the part Of speech tag and the correspondence to the syntactic structure.

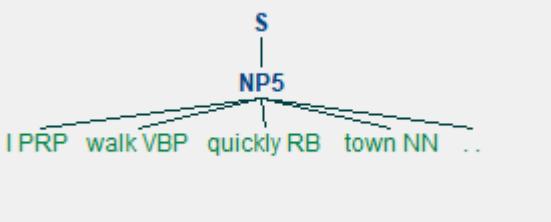

Figure 14: Tree for NP5.

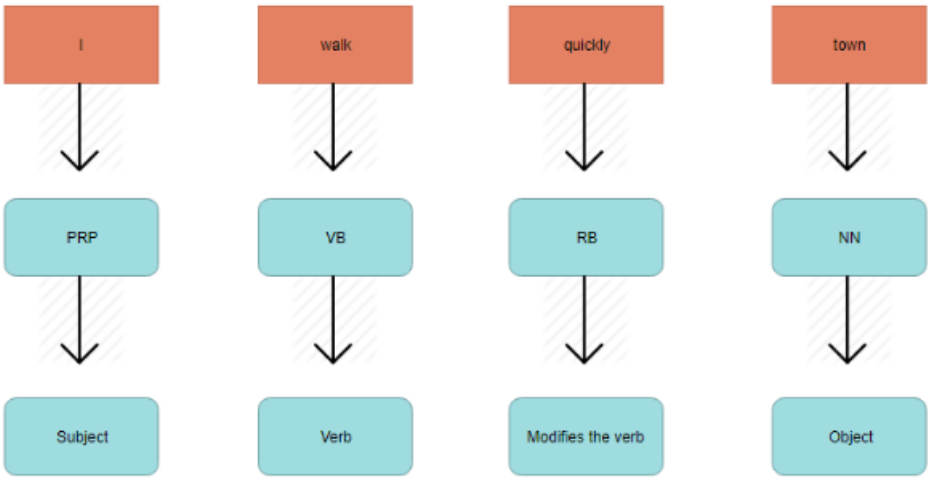

Figure 15: Matching of the sentence with NP5.

Python Code of NP5 rule. The positions from the tree are used to create the final graph that correspond to the sentence.

```
for subtree in x.subtrees(filter=lambda t: t.label().endswith("NP5")):
        #print(subtree)
        cypher = graph.cypher

        str1 = "Merge (n:Person { name:'"
        str1 += subtree[0][0]
        str1 += "'})"
        print(str1)
        cypher.execute(str1)

        str1 = "Merge (n:Person { name:'"
        str1 += subtree[3][0]
        str1 += "'})"
        print(str1)
        cypher.execute(str1)

        cypher = graph.cypher

        str1 = "MATCH (a:Person),(b:Person) WHERE a.name = '"
        str1 += subtree[0][0]
        str1 += "'AND b.name ='"
        str1 += subtree[3][0]
        str1 += "'CREATE (a)-[r:"
        str1 += subtree[1][0]
        str1 += "{ title:'"
        str1 += subtree[2][0]
        str1 += "'}]->(b) RETURN type(r)"
        print(str1)
        print("NP5", str1)
        cypher.execute(str1)
```

NP6

Tag pattern 6 : <PRP><VB.?><NN.?><.>

This tag pattern is a sequence of a personal pronoun, a verb, and a noun. It is similar to the <NN.?><VB.?><NN.?> tag pattern as the personal pronoun takes the place of the Noun.

input_text :" She called her sister."

This is an independent clause that consists of a verb, a subject, and an object. The subject in this sentence is the personal pronoun; the verb shows the action, so it is the main verb, and the Noun is the object. In this sentence," She" is the subject (PRP)," called" is the verb (VB.?)," sister" is the object. The aim is to create one node for the subject" She", one node for the object" sister" and the verb" called" as the relationship type that connects the two nodes.Figure 16 represent the NP6 tree.

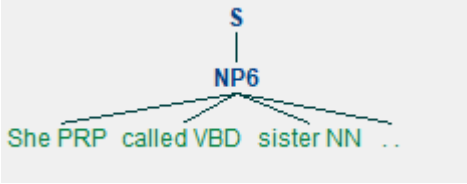

Figure 16: NP6 Tree.

The figure 17 indicates the matching of the words to the part of speech tag and the correspondence to the syntactic structure. NP7

Tag pattern 7: <PRP><VB.?><JJ.?><NN.?><.>

This tag pattern is a sequence of a personal pronoun, a verb, an adjective and a noun. It is similar with the previous tag pattern but the <JJ.?>. have been added.

input_text : " They have schedule a beautiful trip."

This is an independent clause that consists of a verb, a subject and an object. The subject in this sentence is the personal pronoun, the verb shows the action, so it is the

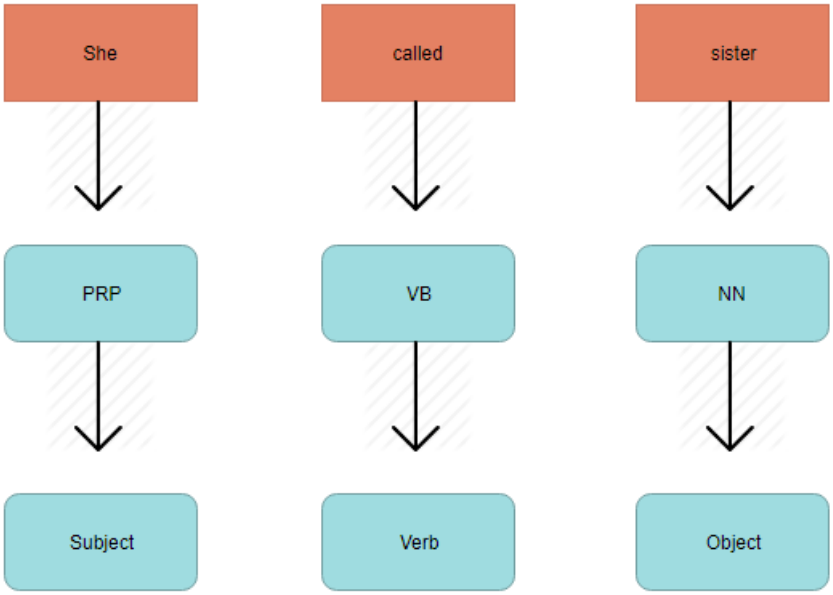

Figure 17: Matching of the sentence with NP6.

main verb, the noun is the object, and the adjective is the object's property. In this sentence," They" is the subject (PRP)," schedule" is the verb (VB.?)," trip" is the object with the property, and" beautiful" (JJ) is the property. The aim is to create one node for the subject" They," one node for the object" trip" with the property" beautiful," and the verb" schedule" as the relationship type that connects the two nodes.Figure 18 represent the NP7 tree.

Figure 18: NP7 Tree.

The figure 19 indicates the matching of the words to the part of speech tag and the correspondence to the syntactic structure. NP8

Tag pattern 8 : <PRP><VB.?><RB.?><.>

This tag pattern is a sequence of a personal pronoun, a verb and an adverb.

input_text :"You walk quickly."

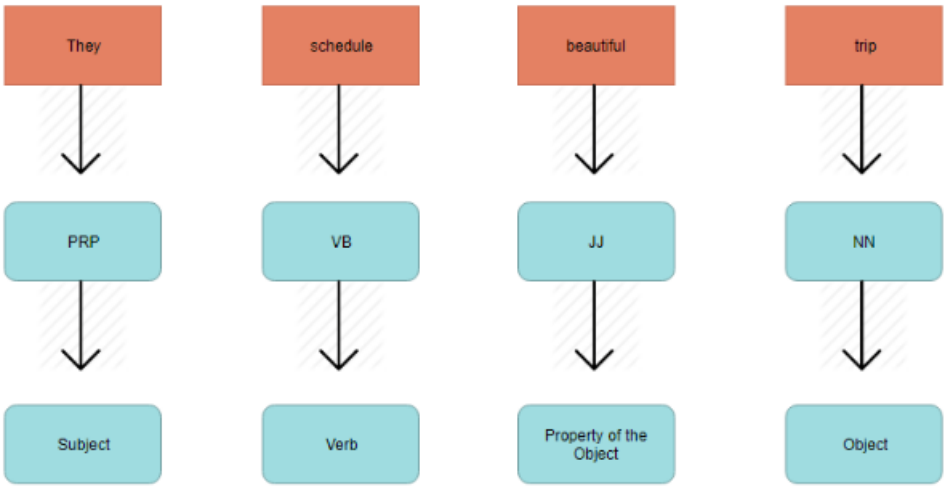

Figure 19: Matching of the sentence with NP7.

This is an independent clause that consists of a verb and a subject. The subject in this sentence is the personal pronoun; the verb shows the action, so it is the main verb. Some verbs can be used without an object. When these verbs have an object, the subject does the action. When they have no object, the action or event happens to the subject. In this sentence," You" is the subject (PRP)," walk" is the verb (VB.?)," quickly" is the adverb. The aim is to create one node for the subject" You" in this sentence. There is not an object, so a node" temp" is created, and the verb" walk" connects the two nodes." Quickly" is the property of the relationship" walk."Figure 20 represent the NP8 tree.

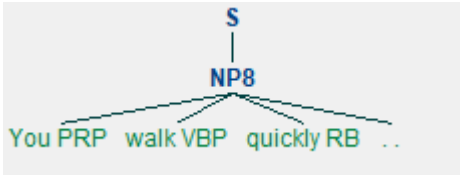

Figure 20: NP8 Tree.

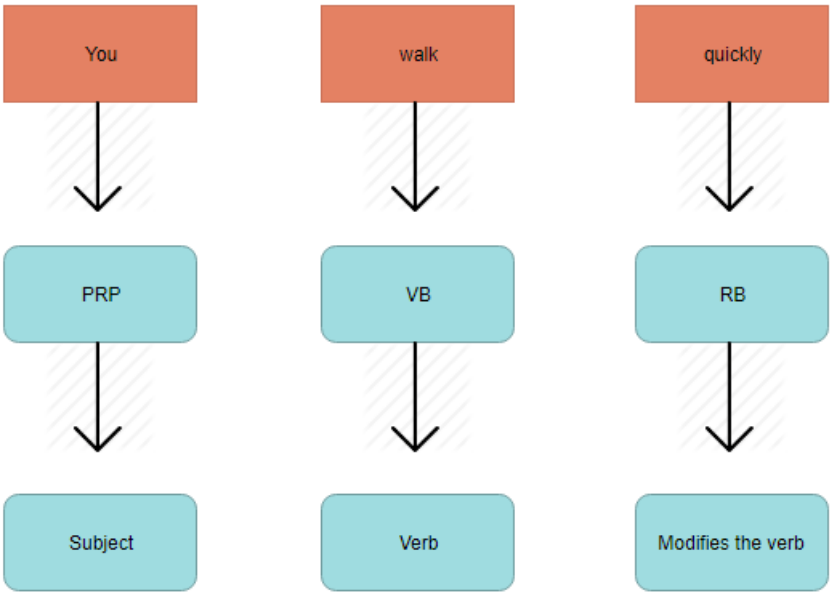

Figure 21: Matching of the sentence with NP8.

The figure 21 indicates the matching of the words to the part of speech tag and the correspondence to the syntactic structure.

NP9

Tag pattern 9: <PRP><VB.?><NN.?><RB.?><.>

This tag pattern is a sequence of a personal pronoun, a verb,a noun and an adverb.

input_text :" He drives the car carefully."

This is an independent clause that consists of a verb, a subject, and an object. The subject in this sentence is the personal pronoun, the verb shows the action, so it is the main verb, the Noun is the object, and the adverb is the verb's property. An adverb is a word that modifies a verb, an adjective, another adverb, or even a whole sentence. This rule is about the adverb that modifies the verb. So, in particular, in this sentence," He" is the subject (PRP)," drives" is the verb (VB.?)," carefully" (RB.?) is the adverb, and" car" (NN.?) is the object. The aim is to create one node for the subject" He," one node for the subject" car," and a relationship to connect the two nodes with the title" drives" and property" carefully."Figure 1 represent the NP9 tree.

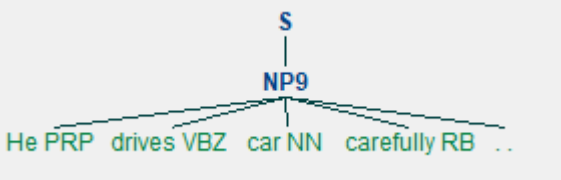

Figure 22: NP9 Tree.

The figure 23 indicates the matching of the words to the part of speech tag and the correspondence to the syntactic structure.

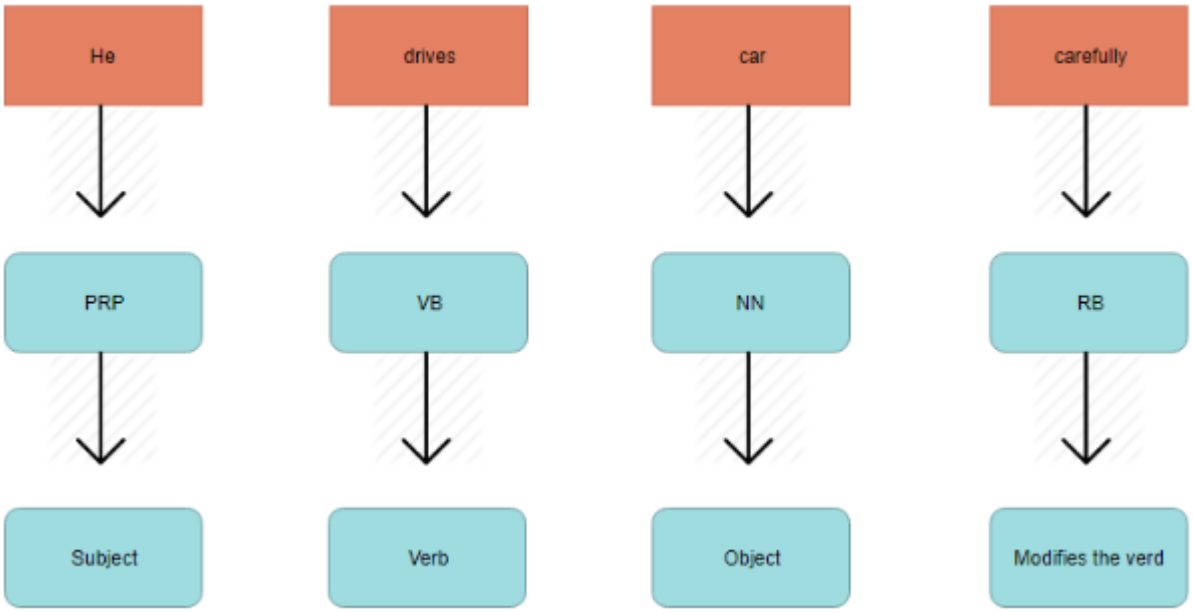

Figure 23: Matching of the sentence in NP9.

4.4 Transfer between editor and Neo4j.

Initially the connection between editor and Neo4j is made with the use of graph class. The attributes of the class Graph are about the host, the password, the user and the supported URI scheme.

]

graph = Graph("http://neo4j:1234@127.0.0.1:7474/db/data/"), uri scheme is http, user is neo4j, password is 1234, localhost is 127.0.0.1:7474.

]

The unstructured text is inserted into the editor as text. According to the tag pattern that the sentence match, a Cypher code is being created to produce the final graph. At the end of every tag pattern-rule process, a string (str1) produced in Cypher, and the method to execute the Cypher code is sent and executed in the graph database server. hrough the graph class, the Cypher code is executed in the browser of Neo4j and creates the final graph for every tag pattern.

cypher.execute (str1)

5 Evaluation

The evaluation for the application will be done in two parts. The first part will compare the graph automatically generated by the Neo4j and the graph created by the application. The second part will measure the performance of the application in a text that is not in the proper syntax format. ext after some changes in the syntax of the sentences, the performance will be measured again. The performance will be measured by the percentage of words from the text used in the final graph, the relationships created in every case, and the percentage of the verbs used. The percentage of the verbs indicates if the application recognizes correct the relationship between the words. If the system correct creates the relationship between the nodes. The initial evaluation is a comparison between the approach created and the approach of Neo4j.

5.1 Evaluation between Neo4j and the application.

Code in Cypher WITH split(tolower("Our generation is facing the greatest threat. Human activity has increased the average temperature of the Earth. Climate change is affecting wildlife, many animals face risk. Planet Earth needs our help. We have to act efficiently. We have to change quickly our lifestyle. Planet Earth is a unique and beautiful home. Some easy changes can make a big difference. We should all of us use reusable cups. Single-use plastic are made from fossil-fuel. We should think our transportations very carefully. The unnecessary use of cars is a major cause of global warming. We have to work for the environment now. The environment improves the quality of life.")," ") As text Unwind Range(0,size(text)-2)As i MERGE (w1:Wordname:text[i]) MERGE(w2:Wordname:text[i+1]) MERGE(w1)-[:NEXT]->(w2) RETURN w1,w2

Table 4: Results in Neo4j.

Words	107
Nodes	76
relationships	98

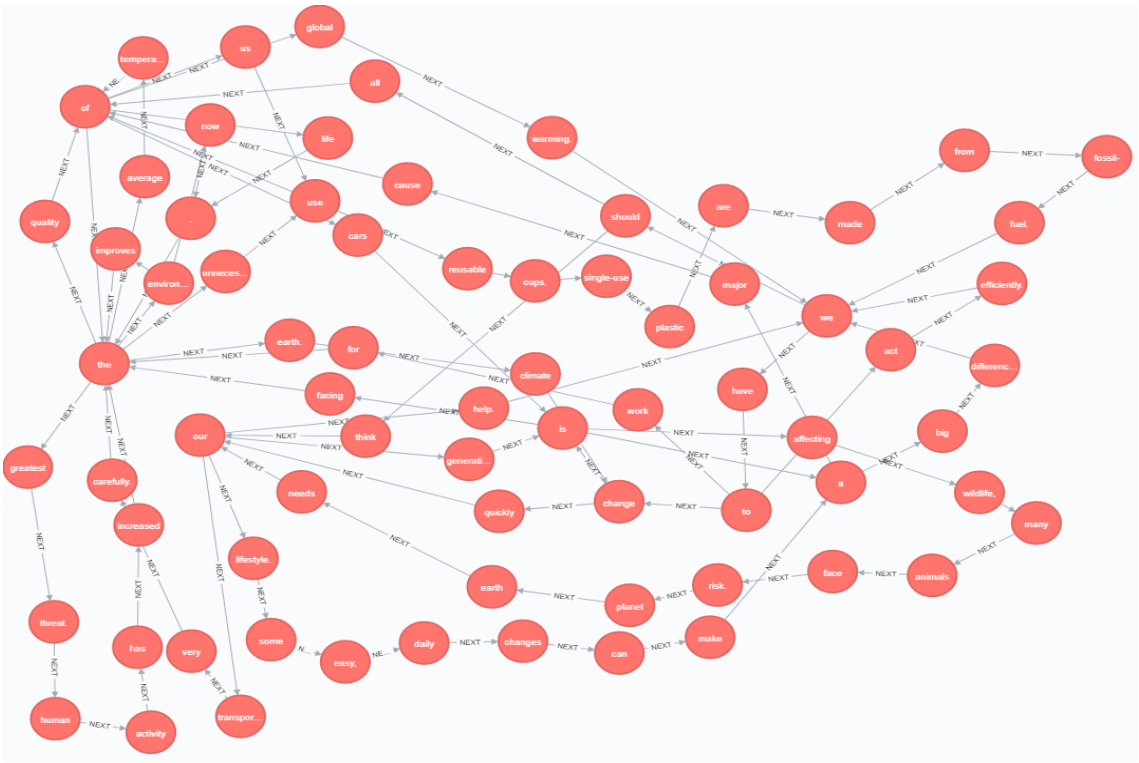

Figure 24: Graph in Neo4j.

There are some significant differences in Neo4j. Table 4 highlights a large number of nodes and relationships the Neo4j creates. Fundamental, Neo4j creates an acyclic graph that connects the word with the next one along a relationship named" NEXT." Also, without the text processing, all the words from the sentence are being used in the graph as nodes. The graph created by the Neo4j is too complicated. It has many nodes and too many edges. Merge is applied in creating the nodes, so there are no multiple nodes, but the nodes' number is much larger than the one of the applications. The most crucial difference is that Neo4j does not use any preprocessing methods to the text before creating the final graph. This is the reason that all words are being used and converted to nodes. The stop-words-removal method removes common words that do not provide further information in the text while at the same time significantly reducing its size. The part of speech tagging would characterize the words and separate their use at the graph.

So the graph that is shown in figure 24 is chaotic and non-functional. Although edges indicate the sequence of words, it is not easy to follow the sentence's path in reality. There are nodes associated with many nodes, so multiple edges start from one node, and there no separation between the sentences. The nodes with more edges are used most

in the text, but those are generally the most common. Those are the words" of," the," we," and" our" in this tex. Usually, the most common words would provide information on the subject of the text. This graph is a visualization of text that shows only the word sequence in a sentence.

5.2 text in the application.

The same text in this application use all the rules that have been created at least one time. Text composed of 107 words, after stop word removal is 67, i.e., the text is reduced by 37,3%. The most used rules are NP1 and NP4, which attribute properties to a noun. More precisely, below, there are the sentences and the rule that corresponds to them.

5.2.1 Rules in the text.

Sentence: *Our generation is facing the greatest threat.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(greatest,threat)

NP4:<NN.?><VB.?><JJ.?><NN.?><.>(generation,facing,greatest,threat)

The tree that created by the rules is presented in figure 25.

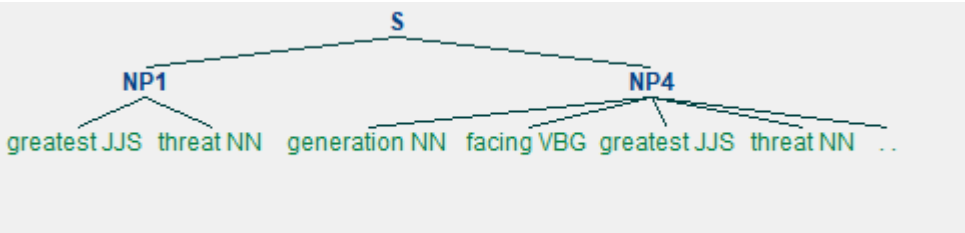

Figure 25: Tree for Sentence 1.

Sentence: *Human activity has increased the average temperature of the Earth.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(average,temperature)(Human,activity)

NP2:<NN.?><NN.?>(temperature,Earth)

NP4:<NN.?><VB.?><JJ.?><NN.?><.>(activity,increased,average,temperature)

The tree that created by the rules is presented in figure 26.

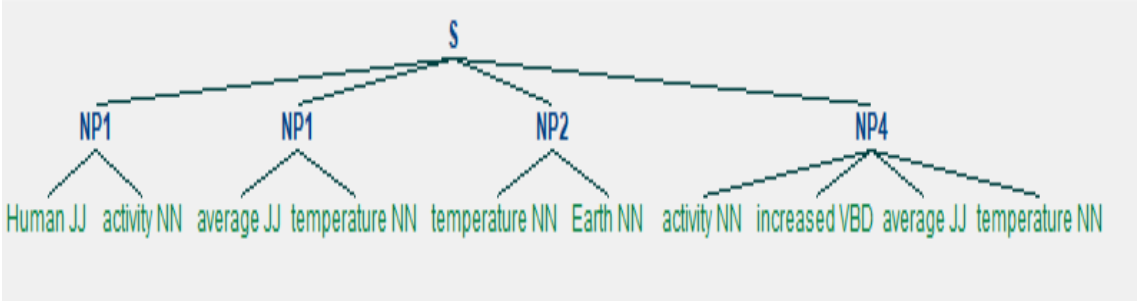

Figure 26: Tree for Sentence 2.

Sentence: *Climate change is affecting wildlife, many animals face risk.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?> (many,animals)

NP2: <NN.?><NN.?>(Climate,change)

NP3: <NN.?><VB.?><NN.?><.>(animals,face,risk) (change,affecting,wildlife)

The tree that created by the rules is presented in figure 27.

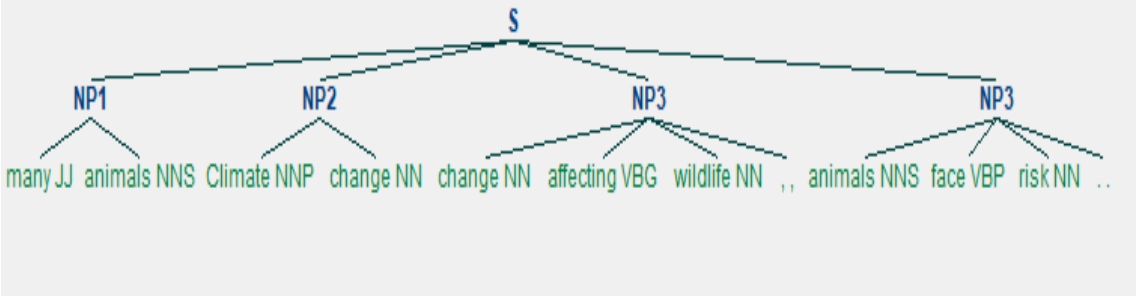

Figure 27: Tree for Sentence 3.

Sentence: *Planet Earth needs our help.*

Rule:

NP3: <NN.?><VB.?><NN.?><.>(Earth,needs,help)

The tree that created by the rule is presented in figure 28.

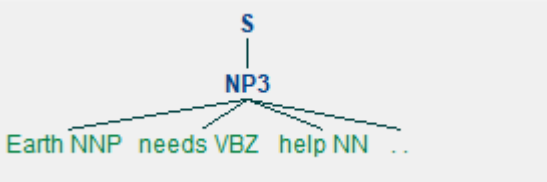

Figure 28: Tree for Sentence 4.

Sentence: *We have to act efficiently.*

Rule:

NP8: <PRP><VB.?><RB.?><.>(We,act,efficiently)

The tree that created by the rule is presented in figure 29.

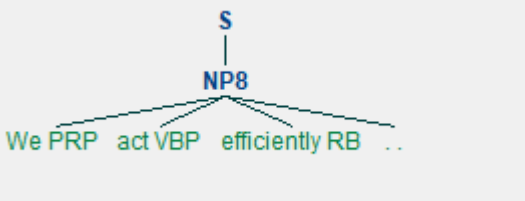

Figure 29: Tree for Sentence 5.

Sentence: *We have to change quickly our lifestyle.*

Rule:

NP5: <PRP><VB.?><RB.?><NN.?><.>(We,change,quickly,lifestyle)

The tree that created by the rule is presented in figure 30.

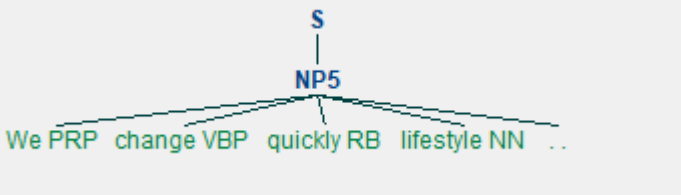

Figure 30: Tree for Sentence 6.

Sentence: *Earth is a beautiful and unique home.*

Rules:

NP1: (<JJ.?><,>)*<JJ.?>+<NN.?>(unique,beautiful,home)

The tree that created by the rule is presented in figure 31.

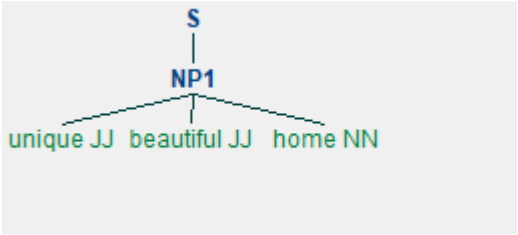

Figure 31: Tree for Sentence 7.

Sentence: *Some easy, daily changes can make a big difference.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(easy,changes)(big,difference)

NP4:<NN.?><VB.?><JJ.?><NN.?><.>(changes,make,big,difference)

The tree that created by the rules is presented in figure 32.

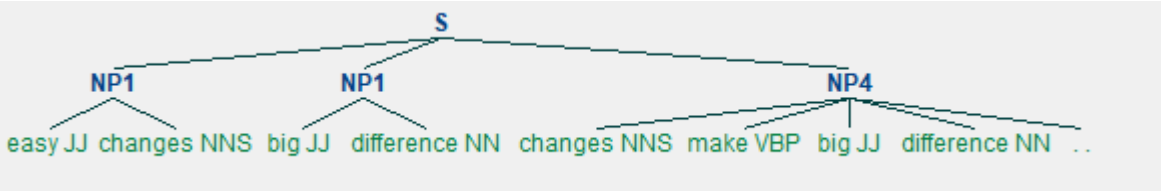

Figure 32: Tree for Sentence 8.

Sentence: *We should all of us use reusable cups.*

Rules:

NP1(<JJ.?><,>)*<JJ.?>+<NN.?>:(reusable,cups)

NP7:<PRP><VB.?><JJ.?><NN.?><.>(We,use,reusable,cups)

The tree that created by the rules is presented in figure 33.

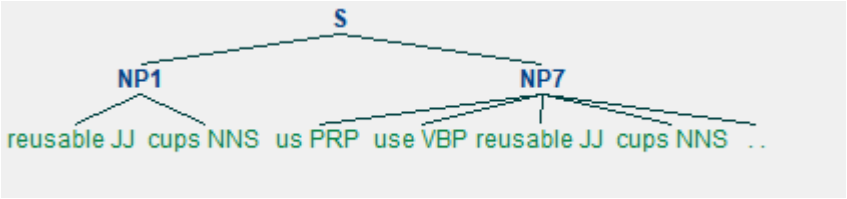

Figure 33: Tree for Sentence 9.

Sentence: *Single-use plastic are made from fossil-fuel.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(fossil,fuel)

NP2: <NN.?><NN.?>(Single-use,plastic)

NP4:<NN.?><VB.?><JJ.?><NN.?><.>(plastic,made,fossil,fuel)

The tree that created by the rules is presented in figure 34.

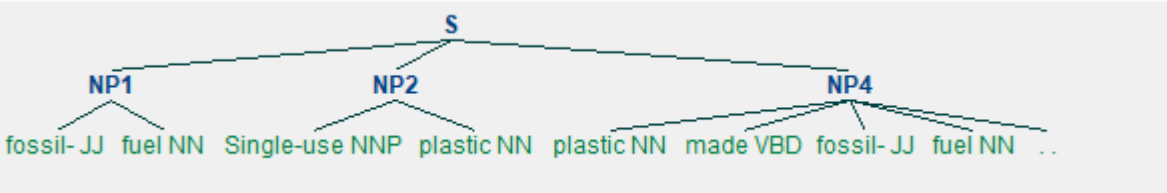

Figure 34: Tree for Sentence 10.

Sentence: *We should think our transportations very carefully.*

Rule:

NP9:<PRP><VB.?><NN.?><RB.?><.>(We,think,transportations,carefully)

The tree that created by the rule is presented in figure 35.

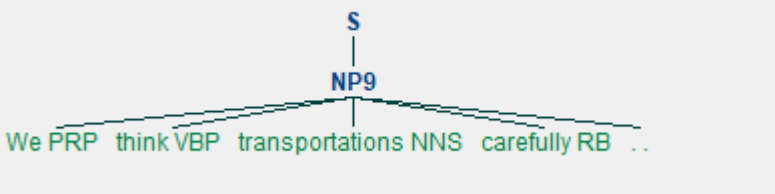

Figure 35: Tree for Sentence 11.

Sentence: *The unnecessary use of cars is a major cause of global warming.*

Rule:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(major,cause)(global,warming)(unnecessary,use)

The tree that created by the rule is presented in figure 36.

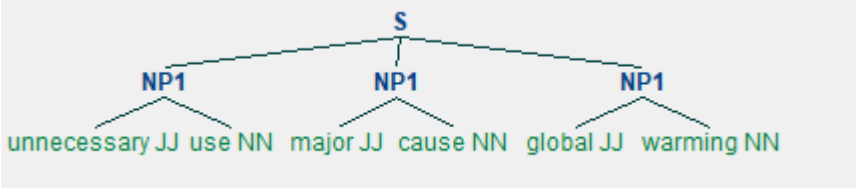

Figure 36: Tree for Sentence 12.

Sentence: *We have to work for the environment now.*

Rule:

NP6: <PRP><VB.?><NN.?><.>(We,work,environment)

The tree that created by the rule is presented in figure 37.

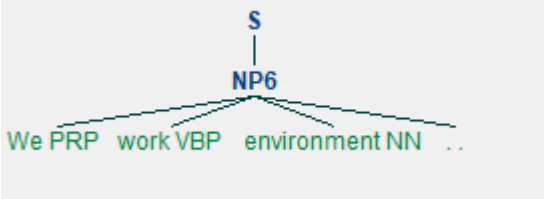

Figure 37: Tree for Sentence 13.

Sentence: *The environment improves the quality of life.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(quality,life)

NP4:<NN.?><VB.?><JJ.?><NN.?><.>(environment,quality,life)

The tree that created by the rules is presented in figure 38.

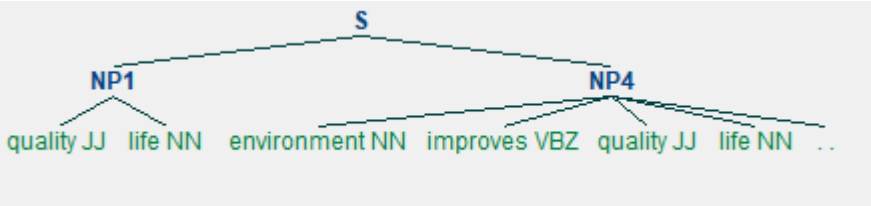

Figure 38: Tree for Sentence 14.

5.3 Evaluation of application.

Table 5: Rules and times of use in text with good performance.

Rules	Times of use uses
NP1	13
NP2	3
NP3	3
NP4	5
NP5	1
NP6	1
NP7	1
NP8	1
NP9	1

Table 5 represents the rules and the times that those rules are used to create the graph. All rules are applied in the text at least one time. Remarkably, rules NP3 and NP4 are used three and five times, respectively. The NP1 rule is used most of the time as it is the one that covers more than one-word sequence. The final graph presented in 39 is a reliable conversion of text into a graph. It is readable, and conclusions about the text can be easily reached.

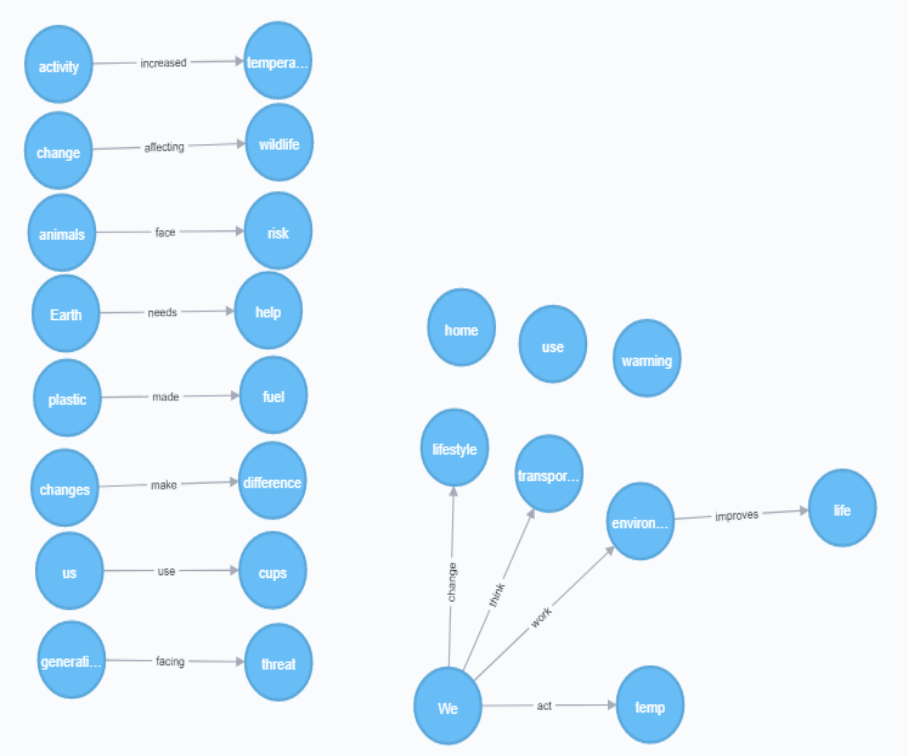

Figure 39: Final graph in Neo4j.

5.4 Text with poor performance

The text for evaluation below it is simple without particular syntax and also consists subsequent and large sentences. It is about the same size as the previous one.

"It's real. It's happening. It's accelerating. And it's our fault. Human activity | particularly the production of greenhouse gasses fossil fuel emissions | is reshaping our planet, effecting rapid environmental change at a rate never seen before. Global temperature averages are creeping upward, seas are warming, rising and becoming more acidic, and extreme weather events such as droughts, wildfires, floods and powerful storms are more commonplace. Here's where you'll find the latest on the effects of climate change, and the measures that scientists, world leaders and innovators are taking to reduce our harmful impact on the planet and mitigate the damage already done."

This text composed of 102 words, after stop word removal the words are 71, i.e., the text is reduced by 31%.

Sentence: *Human activity | particularly the production of greenhouse gasses from fossil fuel emissions | is reshaping our planet, effecting rapid environmental change at a rate never seen before.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(human,activity)(fossil,fuel)(rapid,enviromental,change)

NP2:<NN.?><NN.?>(production,greenhouse)(change,rate)

NP4:<NN.?><VB.?><JJ.?><NN.?><.>(greenhouse,gasses,fossil,fuel)

The tree that created by the rules is presented in figure 40.

Sentence: *Global temperature averages are creeping upward, seas are warming, rising and becoming more acidic, and extreme weather events such as droughts, wildfires, floods and powerful storms are more commonplace.*

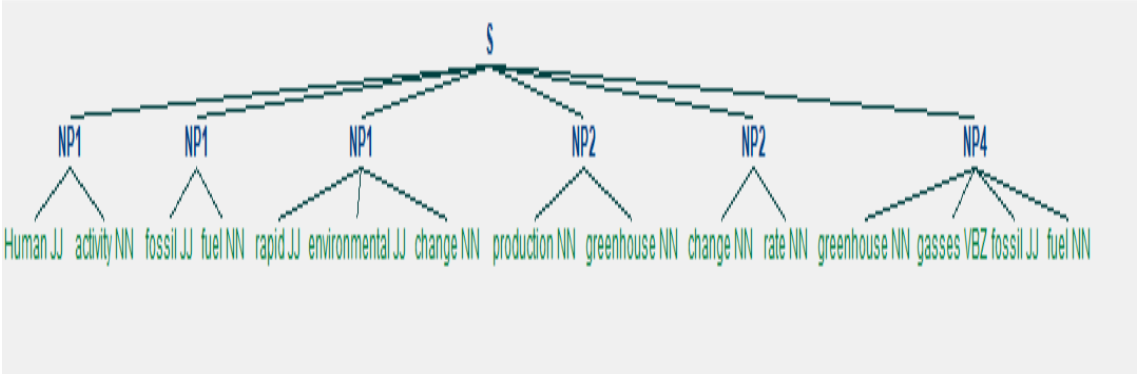

Figure 40: Tree for Sentence 1.

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(Global,temperature)(acidic,extreme,weather) (powerful,storms)

NP2:<NN.?><NN>(storms,commonplace)

The tree that created by the rules is presented in figure 41.

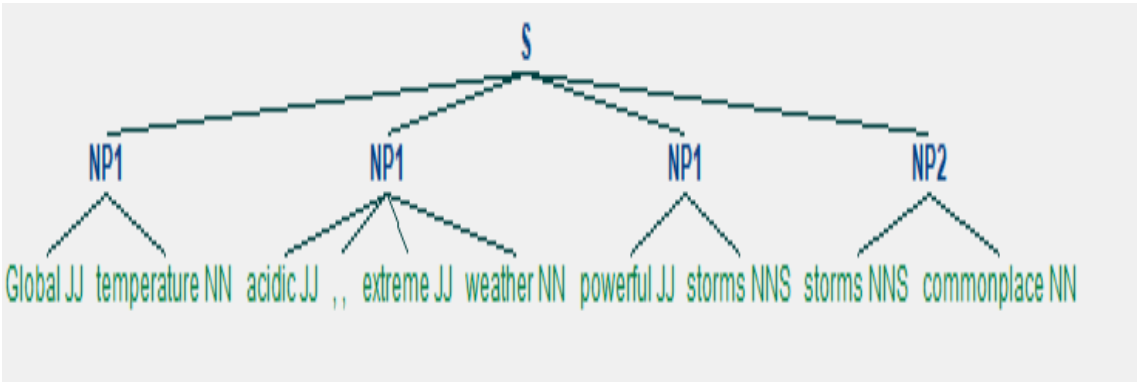

Figure 41: Tree for Sentence 2.

Sentence: *Here's where you'll find the latest on the effects of climate change, and the measures that scientists, world leaders and innovators are taking to reduce our harmful impact on the planet and mitigate the damage already done.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(lattest,effects)(harmful,impact)

NP2:<NN.?><NN>(impact,planet)(mitigate,change)

NP3:<NN.?><VB.?><NN.?><.>(affects,climate,change)

The tree that created by the rules is presented in figure 42.

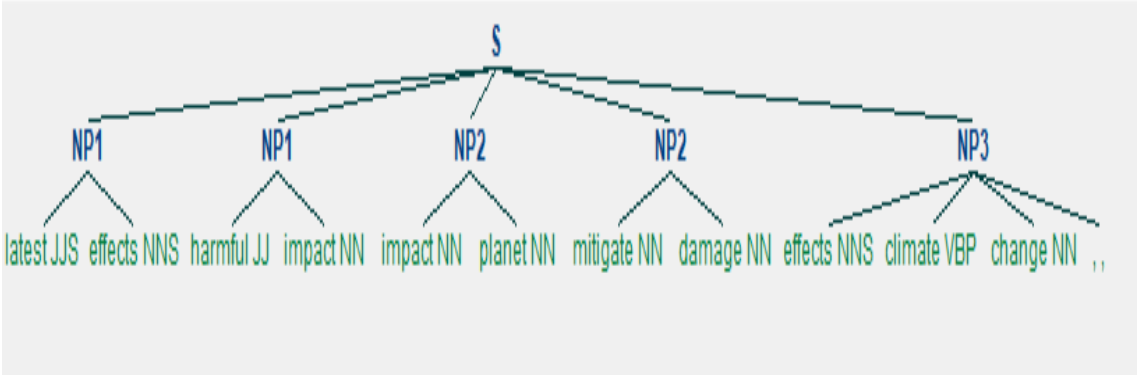

Figure 42: Tree for Sentence 3.

Table 6: Rules used in poor performance text.

Rule	Times of use
NP1	8
NP2	5
NP3	1
NP4	1

This text uses four of nine rules at least once, as shown in table 6 below|a total of 15 rules used in this text. The most used rules are NP1 and NP2, which attribute properties to a noun. For some sentences, there is no matching in rules. The rules that create relationships are relatively few as the text is not in the proper syntax form. The final graph as it is shown in 43 contains minimum information.

text.png

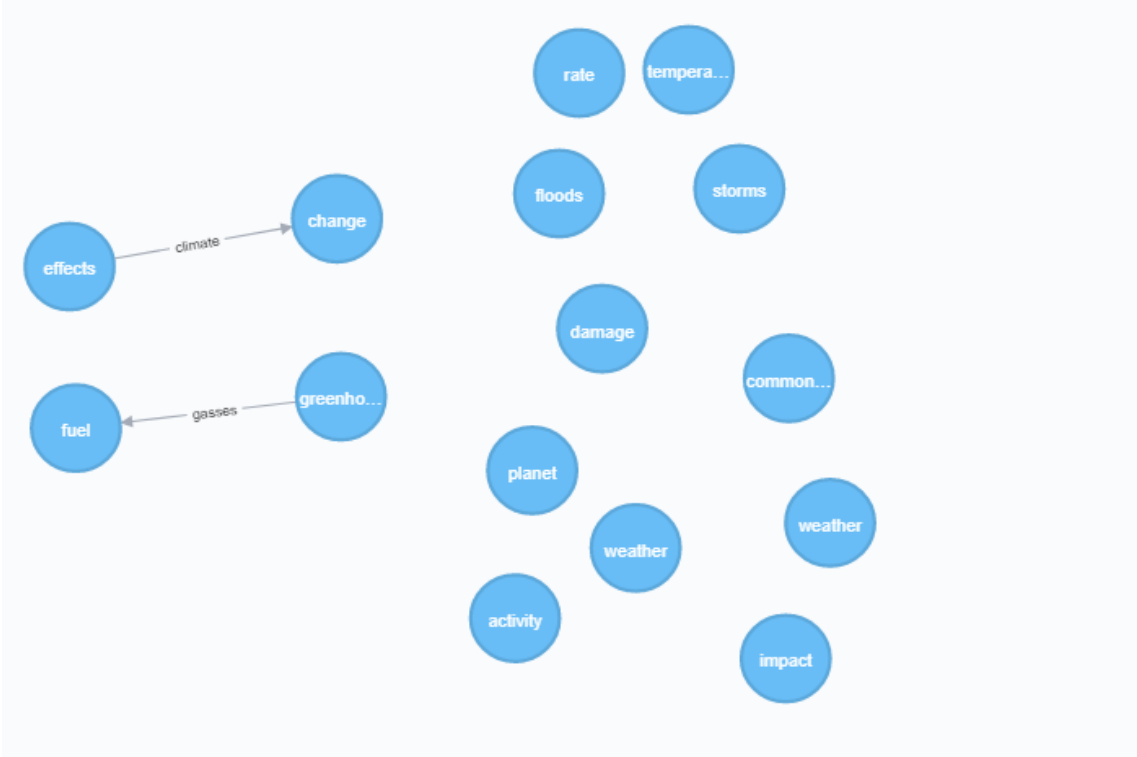

Figure 43: Final graph for the text with poor performance.

5.5 Corrected Text

A preprocessing performed to improve application performance in the previous text. The preprocessing focused on syntax changes while not changing the meaning or style of the text. The aim was to make the sentences have at least one verb, a subject, and an object, so the omit was added in cases that were not complete. Also, subsequent sentences have been eliminated, and some extensive sentences turned into smaller ones. The corrected text is:

"Climate change is real. Climate change is happening. Climate change is accelerating and is our fault. Human activity is reshaping our planet. Particularly the production of greenhouse gasses from fossil fuel emissions. Human activity is affecting rapid environmental change at a rate never seen before. Global temperature averages are creeping upward. Seas are warming. Seas are rising and becoming more acidic. Extreme weather events are happening such as droughts, wildfires, floods and powerful storms. Those weather events are more commonplace. Here is where you will find the latest on the effects of climate change. Here are the measures. The scientists, world leaders and innovators are taking to reduce our harmful impact on the planet. They try to mitigate the damage."

The text is composed of 127 words, after stop word removal the words are 81, i.e., text is reduced by 36,2%.

Sentence: *Climate change is real. Climate change is happening.*

Rules: NP2: <NN.?><NN.?>(Climate,change)

The tree that created by the rule is presented in figure 44.

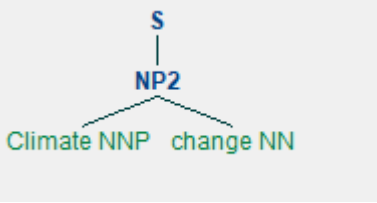

Figure 44: Tree for Sentence 1 in corrected text.

Sentence: *Climate change is happening.*

Rules: NP2: <NN.?><NN.?>(Climate,change)

The tree that created by the rule is presented in figure 45.

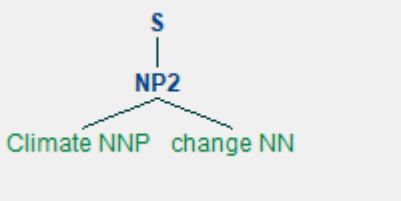

Figure 45: Tree for Sentence 2 in corrected text.

Sentence: *The climate change is happening and it is accelerating.*

Rules:

NP2: <NN.?><NN.?>(Climate,change)

NP3: <NN.?><VB.?><NN.?><.>(change,accelerating,fault)

The tree that created by the rules is presented in figure 46.

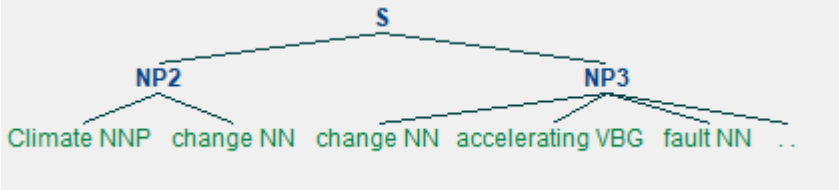

Figure 46: Tree for Sentence 3 in corrected text.

Sentence: *Human activity is reshaping our planet.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(Human,activity)

NP3:<NN.?><VB.?><NN.?><.>(activity,reshaping,planet)

The tree that created by the rules is presented in figure 47.

Figure 47: Tree for Sentence 4 in corrected text.

Sentence: *Particularly the production of greenhouse gasses from fossil fuel emissions.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(particularly,production)(fossil,fuel)

NP2:<NN.?><NN>(production,greenhouse)

NP4:<NN.?><VB.?><JJ.?><NN.?>(greenhouse,gasses,fossil,fuel)

The tree that created by the rules is presented in figure 48.

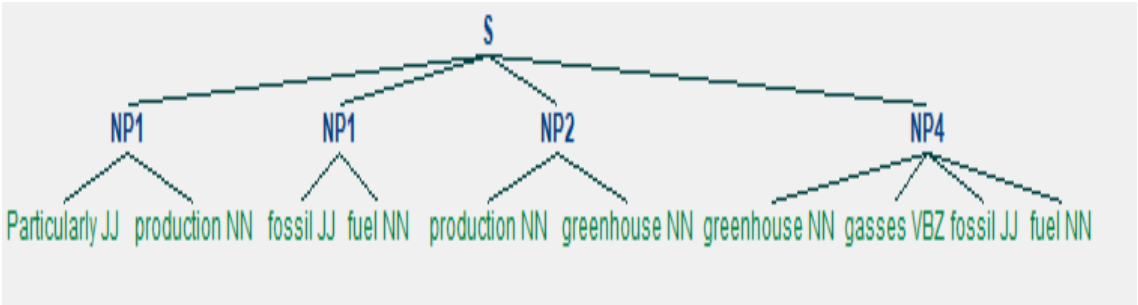

Figure 48: Tree for Sentence 5 in corrected text.

Sentence: *Human activity is affecting rapid environmental change at a rate never seen before.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(Human,activity)(rapid,environmental,change)

NP2:<NN.?><NN>(change,rate)

The tree that created by the rules is presented in figure 49.

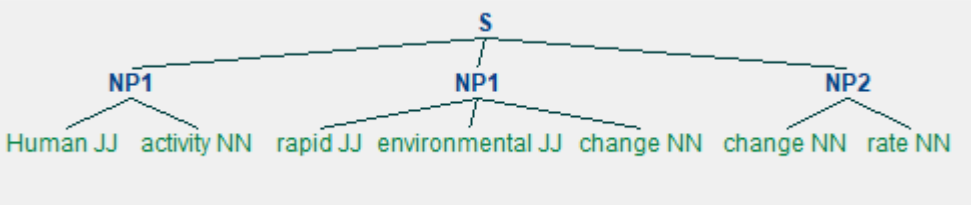

Figure 49: Tree for Sentence 6 in corrected text.

Sentence: *Global temperature averages are creeping upward.*

Rule:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(Global,temperature)

The tree that created by the rule is presented in figure 50.

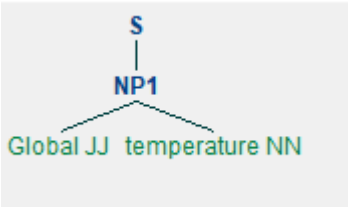

Figure 50: Tree for Sentence 7 in corrected text.

Sentence: *Seas are warming.*

Rule:

NP2:<NN.?><NN>(Seas,warming)

The tree that created by the rule is presented in figure 51.

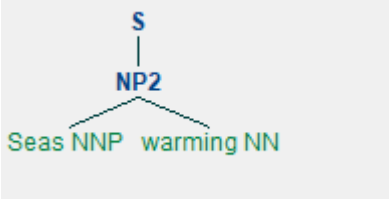

Figure 51: Tree for Sentence 8 in corrected text.

Sentence: *Extreme weather events are happening such as droughts, wildfires, floods and powerful storms.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(Extreme,weather)(powerful,storms)

NP3:<NN.?><VB.?><NN.?><.>(events,happening,droughts)

The tree that created by the rules is presented in figure 52.

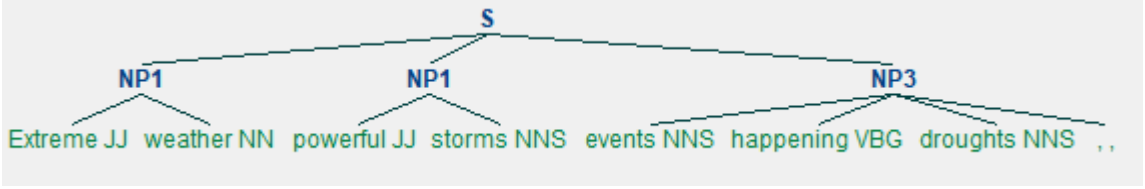

Figure 52: Tree for Sentence 9 in corrected text.

Sentence: *Those weather events are more commonplace.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(weather,events)

NP2:<NN.?><NN>(events,commonplace)

The tree that created by the rules is presented in figure 53.

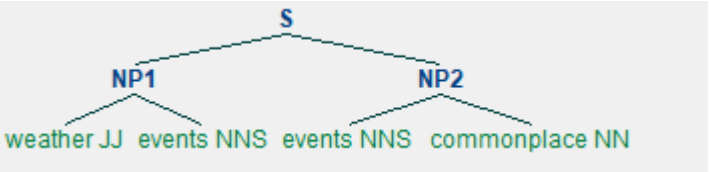

Figure 53: Tree for Sentence 10 in corrected text.

Sentence: *Here is where you will find the latest on the effects of climate change.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(latest,effects)

NP3:<NN.?><VB.?><NN.?><.>(effects,climate,change)

The tree that created by the rules is presented in figure 54.

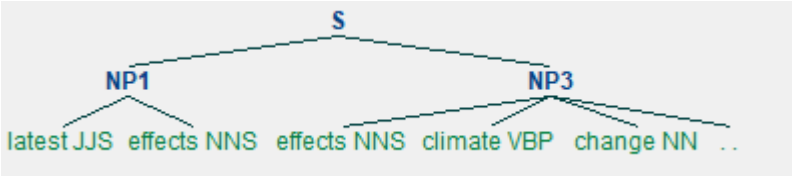

Figure 54: Tree for Sentence 11 in corrected text.

Sentence: *The scientist, world leaders and innovators are taking to reduce our harmful impact on the planet.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(harmful,impact)

NP2:<NN.?><NN>(sea,level)(impact,planet)

The tree that created by the rules is presented in figure 55.

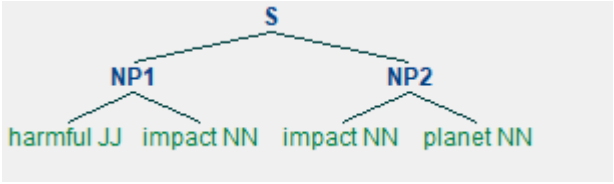

Figure 55: Tree for Sentence 12 in corrected text.

Sentence: *They try to mitigate the damage.*

Rules:

NP1:(<JJ.?><,>)*<JJ.?>+<NN.?>(mitigate,damage)

NP7:<PRP><VB.?><JJ.?><NN.?><.>(They,try,mitigate,damage)

The tree that created by the rules is presented in figure 56.

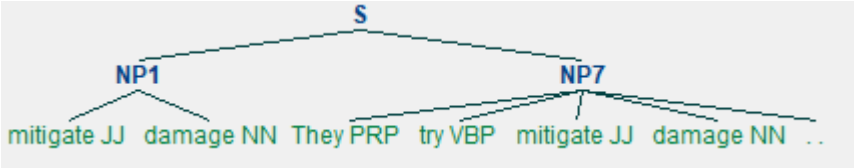

Figure 56: Tree for Sentence 13 in corrected text.

Table 7: Rules and times of use in corrected text.

Rules	Times of use
NP1	12
NP2	9
NP3	4
NP4	1
NP7	1

This text use five of nine rules at least one time. The most used rules are NP1 and NP2 which attribute properties to a noun. Table 7 proves that preprocessing improve the results. The rules and the number of use increased.

Figure 57: Graph of the corrected text.

5.6 Summary

Table 8 compares the performance of the application in the three texts above. Text_1, text_2 and text_3 represent the above cases. More specifically, text_1 is the text that uses all the rules, text_2 is the text with poor performance, and text_ 3 is the corrected one that comes from text_2. The size of the text is measured in words; all three texts are about the same size. The increase in the size from text_2 to text_3 is reasonable after the preprocessing. The percentage reduction of the words varies between 31%-36,2%. Of course, all three texts use the same library and the same methodology. This work aims to represent the relationships between entities in a sentence, so the number of relationships created in the graph is the most important index to evaluate. The increase in the percentage of verbs used as a relationship type after the syntax editing is remarkable. The increase between the text_2 and the text_3 is 33,4%. It is also important to create nodes with properties because no valuable information is lost. Text_3 has the biggest number of nodes with properties. In conclusion, the proposed model has quite satisfactory results in simple text. It has poor performance in more

complicated syntax forms, but it is noteworthy that there is a significant increase in all rates with some minor preprocessing.

Table 8: Summary Table of Evaluation.

	$text_1$	$text_2$	$text_3$
number of words	107	102	120
number of words after stop words removal	67	71	76
% reduction of words	37,3%	31%	36,6%
nodes	25	15	25
relationships	13	2	6
% of verbs that creates relationship	100%	16,6%	50%
nodes with properties	14	13	17
number of rules used	29	15	27

6 Conclusion and Future Research

6.1 Conclusion

In this thesis, the proposed method's goal is to extract a graph structure from simple text. The automatically generated graph from the Neo4j graph database is unreadable and incomplete in the information. The application aims to develop a graph making the best use of the valuable information from the text. The initial step was the preprocessing of the text. Preprocessing leads to better results by removing unnecessary information from the text and adding information about the sentence's syntax structure. The speech tagging method converts the words' sequence into a speech sequence, which is very important for the final process. The tag-patterns have considerable importance as they form the rules with whom the sentences separate and create the corresponding graph. As presented in section 5 of this thesis, the result shows that the proposed method achieves high performance when the sentence structure is relatively simple. Even in cases where the text has more complicated syntax (text_2), it is depicted in the graph. The application constitutes a good attempt at the extraction of graph-structured information from text. The final graph contains enough information about the initial text, making it efficient and editable for further analysis.

6.2 Future Research

Further research could be developing more rules and more general rules to use more parts of speech. They will make the application more efficient in real text data, especially for data that come from social media. Another essential step will be the use of subordinates' sentences. The processing of subordinate sentences by the created application is one point that could use much improvement. New rules could be made to correspond to all the types of sentences. A different approach for developing the rules could be starting the process from the verb and then searching for the rest of the elements. Another vital improvement would be the synergy in developing the application with a linguist specialist to understand the sentences' syntax better and hence improve the text processing.

7 References

References

[1] Krithika, LB and Akondi, Kalyana Vasanth, 2014. Survey on Various language Processing Toolkits, *World Applied Sciences Journal*,pp. 399-402.

[2] Rada Mihalcea, Hugo Liu, and Henry Lieberman,2006. NLP (Natural Language Processing) for NLP (Natural Language Programming). *Computational Linguistics and Intelligent Text Processing, 7th International Conference, CICLing* Mexico City, Mexico, 2006.

[3] Jones, Karen Sparck,1994. Natural language processing: a historical review.*Current issues in computational linguistics: in honour of Don Walker. Springer* Dordrecht, pp. 3-16.

[4] Mani, I. ,2001. *Automatic summarization (Vol. 3)*. John Benjamins Publishing.

[5] Machine Translation, The Stanford Natural Language Processing Group. https://nlp.stanford.edu/projects/mt.shtml

[6] Yadav, V., Bethard, S. ,2019. *A survey on recent advances in named entity recognition from deep learning models*. arXiv:1910.11470.

[7] wikipedia.org

[8] Liu, B., Zhang, L. ,2012. *A survey of opinion mining and sentiment analysis. In Mining text data* Springer, Boston, MA. (pp. 415-463).

[9] Mattew Mayo,2018.*The main approaches to Natural Language Processing Taskat:<https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html>Accessed 2018*

[10] Kendall Forney,2017*Pre-Processing in Natural Language Machine Learning Availiable at<https://towardsdatascience.com/pre-processing-in-natural-language-machine-learning-898a84b8bd47> Accessed 2017*

[11] Kannan, S., Gurusamy, V.,2014. Preprocessing techniques for text mining. International Journal of Computer Science Communication Networks, 5(1),pp. 7-16.

[12] Anna Turu Pi,Ozge Koroglu,Esteban Zimányi, 2017.*Graph Databases and Neo4J*Université libre de Bruxelles

[13] Needham, M., Hodler, A. E. (2019). Graph Algorithms: Practical Examples in Apache Spark and Neo4j. O'Reilly Media.

[14] Dorow, B.,2006. A graph model for words and their meanings.

[15] Nastase, MihalceaV., , R., Radev, D. R., 2015. *A survey of graphs in natural language processing*. Natural Language Engineering, 21(5),pp. 665-698.

[16] Database Management System (DBMS) https://searchsqlserver.techtarget.com/definition/database-management-system)

[17] Angles, R., Gutierrez, C.,2008. Survey of graph database models. ACM Computing Surveys (CSUR), 40(1) pp. 1-39.

[18] Loper, E., Bird, S., 2002. *NLTK: the natural language toolkit*.

[19] neo4j.com

[20] Miller, J. J,. 2013.*Graph database applications and concepts with Neo4j*. In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA (Vol. 2324, No. 36).

[21] Robinson, I., Webber, J., Eifrem, E.,2013 *Graph databases*. O'Reilly Media, Inc..

[22] py2neo.org

[23] Perkins, J.,2014. *Python 3 text processing with NLTK 3 cookbook*. Packt Publishing Ltd.

[24] Keselj, V.,2009. *Speech and Language Processing Daniel Jurafsky and James H. Martin* Stanford University and University of Colorado at Boulder.

[25] nltk.org