

KNOWLEDGE DISCOVERY FOR CLINICAL DECISION SUPPORT SYSTEM IN PATIENT RECORDS

by

Dev Budhathoki

July, 2018

Director of Thesis: Kamran Sartipi, PhD

Major Department: Computer Science

Abstract: Knowledge discovery from the patient's health records is a challenging task for the medical specialists. The knowledge generated from the patient's records can assist specialists to make an effective decision and recommend more precise diagnosis. This provides the basis for decision-making process with the recommendation for patient diagnosis and expertise advice by retrieving the information from the knowledgebase. This research aims at utilizing data mining techniques to discover patterns and relationships in between diagnosis and corresponding symptoms. The extracted patterns are used to assist the physician to determine the precise diagnosis with patient's context. We consider graph database-Neo4j to develop a knowledgebase that stores knowledge in the ontological form of patterns and relationships and use the knowledgebase in clinical decision support system to provide recommendations of next possible symptoms and diagnosis for the effective recommendation. In addition, we integrate the expert knowledge with our knowledgebase and explore the feature of graph visualization, with more detail information of patterns and connection of associated patterns in the knowledgebase.

KNOWLEDGE DISCOVERY FOR CLINICAL DECISION SUPPORT SYSTEM
IN PATIENT RECORDS

A Thesis

Presented to The Faculty of the Department of Computer Science
East Carolina University

In Partial Fulfillment of the Requirements for the Degree
Master of Science in Computer Science

by

Dev Budhathoki

July, 2018

Copyright Dev Budhathoki, 2018

KNOWLEDGE DISCOVERY FOR CLINICAL DECISION SUPPORT SYSTEM
IN PATIENT RECORDS

by

Dev Budhathoki

APPROVED BY:

DIRECTOR OF THESIS:

Kamran Sartipi, PhD

COMMITTEE MEMBER:

Nasseh Tabrizi, PhD

COMMITTEE MEMBER:

Mark Hills, PhD

CHAIR OF THE DEPARTMENT

OF COMPUTER SCIENCE:

Venkat Gudivada, PhD

DEAN OF THE

GRADUATE SCHOOL:

Paul J. Gemperline, PhD

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Dr. Kamran Sartipi, without his guidance and supervision this thesis would not have happened. Although it was challenging, his feedbacks, constant support and valuable comments throughout my work helped me to complete my thesis.

I would like to thank my committee members Dr. Mark Hills and Dr. Nasseh Tabrizi for proof-reading the manuscript, thus making this thesis more intelligible. I feel lucky to have Dr. Tabrizi as my academic advisor and my sincere admiration to Chairman Dr. Venkat Gudivada, they have always supported and guided me towards the right direction. Also, I would like to express my special thanks to all the faculties and staffs of computer science who directly or indirectly contributed by facilitated me with a warm atmosphere to complete my thesis by providing adequate knowledge, ideas, and techniques which I could utilize.

I would like to express eternal appreciation to my parents and family for their continuous encouragement, motivation, and unconditional support. Thank you for being understanding and supportive. Special thanks to my fellow students for keeping me focused throughout these years: Anil Adhikari, Bigyan Pandit, Rabindra Khanal, and Dasra Khadka.

Table of Contents

LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF LISTINGS	xi
1 INTRODUCTION	1
2 TOOLS AND TECHNOLOGIES	5
2.1 Neo4j	5
2.2 Cypher Query Language	5
2.3 Py2neo	6
3 BACKGROUND	7
3.1 Concept Lattice	7
3.2 Frequent Pattern Mining	8
3.3 Association Mining	8
3.4 Knowledge-Driven Decision Support System	9
3.5 Graph Database	10
3.5.1 The Property Graph Model	10
3.6 Data Mining	12
3.7 Ontology and Resource Descriptive Framework (RDF)	14

4	RELATED WORK	16
4.1	Formal Concept Analysis	16
4.2	Decision Support Systems	18
4.3	Data Mining and Artificial Intelligence (AI)	19
5	APPROACH	21
5.1	Association Mining	22
6	ARCHITECTURE	29
7	EXPERIMENTATION	36
7.1	Dataset	36
7.2	Preprocessing	40
7.3	Data Mining and Pattern Generation	42
7.4	Post Processing	44
7.4.1	MAG Creation	44
7.4.2	Finding the Relationships among the Patterns	45
7.4.3	Creating a Knowledgebase	46
7.4.4	Integrating Expert Knowledgebase	49
8	ANALYSIS OF KNOWLEDGEBASE	51
8.1	Verify Number of Nodes	51
8.2	Evaluate Indegree and Outdegree	52
8.2.1	Degree Histogram of MAG	54
8.2.2	Check Triangle Formation and Transitive Closure Check	54
8.3	Evaluate Number of Symptoms in MAGs	55
8.4	Evaluate Number of Diagnosis in MAGs	56
8.4.1	All Shortest Path	58

9	CASE STUDY	59
10	FUTURE ENHANCEMENT AND CONCLUSION	64
	BIBLIOGRAPHY	66

LIST OF TABLES

7.1	Data Mining Parameters Variation	42
7.2	Sample Frequent Itemset	43

LIST OF FIGURES

3.1	Graph Representation	12
4.1	Concept-Lattice	17
5.1	Knowledge Discovery Process	22
5.2	MAG Representation	26
6.1	CDSS Architecture [1]	30
7.1	Diagnosis Dictionary	37
7.2	Raw Dataset	38
7.3	Symptoms Dictionary	39
7.4	Annotated Dataset	40
7.5	Transformed Dataset	41
7.6	Symptoms Bucket	41
7.7	MAGs with Itemsets and Baskets	45
7.8	Relationship Between MAGs	46
7.9	Example: MAG Node with Itemsets and Baskets	47
7.10	Example: MAG165 Connections	48
8.1	Example: MAG, Diagnosis and Symptoms Count	52
8.2	MAGs Outdegree	53

8.3	MAGs Indegree	53
8.4	Degree Histogram Graph	54
8.5	MAGs and Symptoms Count	56
8.6	MAGs Diagnosis Count	57
8.7	MAG36 Connection	57
8.8	All shortest path between MAG36 and MAG575	58
9.1	Output 1	61
9.2	Output 2	61
9.3	Output 3	61
9.4	Output 4	62
9.5	Overall Connection of MAG36	62

LIST OF LISTINGS

Chapter 1

Introduction

Electronic health records (EHR) maintain medical and historical patient's data which are growing tremendously due to the availability of advanced information technology in healthcare domain. Several records and data elements are generated from a single patient during their diagnosis, treatment, and billing [2]. EHR includes relevant information, such as insurance information, demographic data, and even data from personal wellness devices. Huge capitals are invested in order to provide effective healthcare for a patient as the statistics from global EHR market share was estimated about USD 20.55 billion in 2016 [3]. Extracting all the symptoms and determining proper diagnosis to provide effective treatment could reduce a cost for the patient. A large amount of patient data could be used to extract the patterns and relationships associated. EHR data are in a raw format that includes medical history, vital signs, progress notes, diagnoses, medications, immunization dates, allergies, lab data and imaging report [4]. Knowledge discovery from the patient's health records is a challenging task. Data Mining is one of the popular techniques that provides a way to discover hidden patterns and relationships among variables in large dataset [5]. Data mining is the technique of knowledge extraction which is implicit in data that potentially contains useful information. Evaluation of hidden patterns and trends within the data could provide a better understanding of disease progression and manage-

ment. Data mining techniques are used in different sectors such as decision support, prediction, forecasting, and estimation. Data mining, machine learning, and big data analytics are actively used to create predictive models and advanced processes to provide accurate and knowledgeable decision support.

In healthcare domain, quality services with appropriate diagnosis and effective treatment at an affordable cost are challenging. A simple mistake in diagnosis may cause a major damage with disastrous outcomes. An automated decision support system can assist the healthcare personnel to make correct decisions during the examination of a patient that makes the diagnostic process more objective and reliable. Determining the patterns and relationships from the large dataset generates the knowledge that guides the personnel during the decision-making process. Not only patterns and knowledge from the data, the patient medical history and advice from domain experts must be considered to empower decision making [6]. A Clinical Decision Support System (CDSS) should interact with health personnel, analyze and find patterns and connections in EHR, integrate it with knowledge from domain experts, searches the patterns in knowledge database and provides recommendations for patient diagnosis, treatment and expertise advice which is stored in the knowledge base. Physician and health practitioners retrieve stored medical knowledge in order to take the effective decision with the current condition of a patient in order to improve the practitioner's medical practice [7].

In this thesis, we first explore concept lattice analysis technique to gather useful information from data. With this technique, we gather frequent itemsets (a group of symptoms) that could help the physicians to explore the associated diagnose corresponded to the concept. However, we are interested in using these concepts to design and develop a knowledge base that could assist physician during decision-making process to recommend more appropriate diagnosis with the context. The context is the

current condition of the particular patient which matches with their symptoms and diagnosis. We designed and developed a clinical decision system with knowledgebase by integrating experts and mined knowledge from EHR in a graph database that performs the following tasks:

1. Collect patient symptoms as users input, parse and tokenizes them.
2. Update the context by combining new symptoms of patient and historical symptoms.
3. Search frequent itemset that matches the context in the knowledge base with their degree of relevance.
4. Recommend all possible diagnosis and symptoms which are relevant to the current context that provides comparing measure to assist the health personnel in making the proper decision.
5. Update context with physician decision of new context and repeat (5) until the appropriate diagnosis is identified.
6. Provide visualization of highly qualitative extracted knowledge on highly scalable graph database Neo4j.

The principal contribution of this thesis is:

1. To discover the knowledge from the data applying data mining techniques and to develop a knowledgebase.
2. To integrate expert knowledge in the knowledgebase.
3. To develop a prototype clinical decision support system that uses knowledgebase for the decision-making process.

4. To introduce the exploration technique in graph database that provides visualization for decision making.

Chapter 2

Tools and Technologies

In our research, different tools and technologies were used to create knowledgebase and clinical decision support system. Python is the language which we used for the implementation and data manipulation operation. Here in this section, we discuss in detail.

2.1 Neo4j

Neo4j is the most popular graph database management system developed by Neo4j Inc. according to DB-Engines ranking [8]. It is a high-performance graph store with all the features expected of a mature and robust database, like a friendly query language and ACID transactions.

It is implemented in Java and accessible from driver software written in different languages such as Java, .Net, JavaScript and Python using the Cypher Query Language through a transactional HTTP endpoint, or through the binary "bolt" protocol. In our research, we used py2neo driver [9] toolkit to access the graph database.

2.2 Cypher Query Language

Cypher is a declarative language that describes patterns in graphs visually using an ASCII-art syntax [10]. Cypher allows for expressive and efficient querying and

updating of the graph store. Cypher is very powerful language. Cypher is designed to be a human query language that is suitable for both developers and operations professionals to work with real-world data. It is SQL-inspired language. It uses various clauses similar to SQL.

2.3 Py2neo

Py2neo wraps all the libraries to access Neo4j graph database with HTTP request and provides a higher level API, admin tools, an interactive console and cypher lexer. It is a client library and toolkit for working with Neo4j within Python applications and from the command line. In our research, we used py2neo driver to create the user interface that connects the neo4j graph database to perform all the operation such as creating nodes and relationships, search operation, delete operation along with cypher queries.

Chapter 3

Background

In this section, we discuss various background knowledge required for our approach.

3.1 Concept Lattice

A formal context is a triplet (G, M, I) where G and M are two non-empty sets has $I \subseteq G \times M$ binary relation between G and M . The elements of G are objects and elements of M are attributes and I the incidence of the context (G, M, I) . For $A \subseteq G$ and $B \subseteq M$, we define

$$A' = \{m \in M \mid (g, m) \in I, \forall g \in A\}$$

$$B' = \{g \in G \mid (g, m) \in I, \forall m \in B\}$$

For all $A_1, A_2, A \subseteq G$ and $B_1, B_2, B \subseteq M$ satisfy following rules [11]

$$1. A_1 \subseteq A_2 \Rightarrow A_2' \subseteq A_1'$$

$$2. B_1 \subseteq B_2 \Rightarrow B_2' \subseteq B_1'$$

$$3. A \subseteq A''$$

$$4. A' \subseteq A'''$$

$$5. B \subseteq B''$$

$$6. B' \subseteq B'''$$

$$7. A \subseteq B' \iff B \subseteq A'$$

A pair (A, B) is a formal concept of (G, M, I) if and only if $A \subseteq G$, $B \subseteq M$, $A' = B$, and $A = B'$.

A is called the extent and B is the intent of the concept (A, B) . The concepts of a given context are naturally ordered by the subconcept-supercontext relation defined by $(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 (\iff B_2 \subseteq B_1)$. The set of all formal concepts of a context (G, M, I) is called the concept lattice of the context (G, M, I) .

3.2 Frequent Pattern Mining

Frequent patterns are patterns which consist of itemsets, subsequences, and substructures in a dataset with a frequency more than a specified threshold[12]. For example, milk and bread in the transactional dataset that appears frequently together is a frequent itemset. A subsequence could be buying first a TV, then a DVD player, and then various CDs and DVDs occurring frequently in a shopping history database is considered as a frequent sequential pattern. A substructure refers to different structural forms, such as subtrees, subgraphs, or sublattices, which may be combined with itemsets or subsequences. If these substructures occur frequently in a graph database, then it is a frequent structural pattern. Frequent pattern mining plays a vital role in mining associations, sequences, and finding interesting relationships among data.

3.3 Association Mining

The concept of association mining was first introduced by Agrawal [13] in the form of association rules mining, aiming at analyzing customer purchase habit by extracting associations between items in customer shopping baskets. An itemset is a set of items that frequently appear in shopping baskets. An itemset is a set of items with the car-

dinality of k which is called k -itemset. The support of an itemset is the number of transactions (i.e. baskets) that contain that itemset in the transaction database. The Apriori algorithm [13] passes over the transaction database multiple times to discover frequent k -itemsets that appear in transactions more than a user-specified threshold, namely minimum support (minsup). In the first pass, the algorithm counts the support of individual items and determines the frequent 1-itemsets. In each subsequent pass, the algorithm selects different frequent $(k-1)$ -itemsets found in the previous pass to generate candidate k -itemsets by joining those frequent $(k-1)$ -itemsets. The candidate k -itemset will be deleted from candidate list if any of its subsets is not frequent, i.e., each subset of a candidate itemset must itself be frequent. Each candidate k -itemset must also have minimum support in order to be considered as frequent k -itemset. This iterative process will terminate when the algorithm cannot generate any larger frequent k -itemset.

3.4 Knowledge-Driven Decision Support System

A knowledge-driven decision support system (KD-DSS) provides specialized problem-solving expertise stored as facts, rules, procedures, or in similar structures [14]. It is an interactive software-based system intended to suggest or recommend actions to decision makers using a knowledgebase. The knowledge discovery process may consist of following steps: i) data integration and feature selection; ii) data mining to discover and gain knowledge; iii) knowledge interpretation and representation. The knowledge is explicitly represented via automatic tools as ontology or rules which assist in DSS behave like an intelligent consultant [1]: supporting decision makers by gathering and analyzing evidence, identifying and diagnosing problems, proposing possible actions, and evaluating the proposed actions. KD-DSS has the capabilities

of self-learning, identifying the associations between raw data, integrating with data mining techniques to discover hidden patterns, and performing heuristic optimizations [15]. These abilities turn KD-DSS into an intelligent process which improves the accuracy of decision making.

3.5 Graph Database

The graph database is a database designed to establish the relationships between different data as a first-class citizen in the data model [16]. In a real world, the data are usually connected and related. To establish relationships, store and query these relationships effectively, graph database provides the proper framework. Compare to other databases, the graph database reduces the query time through expensive JOIN operations.

Accessing nodes and relationships in a native graph database is an efficient, constant-time operation and allows you to quickly traverse millions of connections per second per core. Highly connected and complex data can be modeled as a graph in the graph database. It is possible to search the pattern from the graph database and easy to explore the connection in the form of a graph. User queries with patterns and starting point, graph database can explore all the connected neighbor nodes- collecting all the information related to nodes and relationship.

3.5.1 The Property Graph Model

Property graph model is similar to the entity-relationship model or object model. It consists of nodes and relationships.

1. Nodes: Nodes are the basic entity of the graph consist of a number of attributes that store data in form of key-value pairs called properties. Nodes are usually

tagged with labels that could represent the roles. The metadata information such as index or constraint information can be also attached to the node.

2. Relationships: One node is connected to the other node with the help of relationship. A relationship is directional with start and end nodes. Similar to the nodes, the relationship also consists of properties in the form of key-value pairs. The name is assigned to the relationship with respect to the semantically relevant connection. Generally, quantitative properties are related to the relationship, such as weight, distances, time, intervals, costs, strength. A node having a relationship cannot be deleted without deleting the affiliated relationships.

It is difficult to find the knowledge from huge data set as data could be stored in different formats, the size of the data could be huge and accessed time increases while dealing with huge amount of data. This leads to the concept of big data. Big Data is characterized by 3V's namely Variety- various data format of data, Volume- quantity of data and Velocity- the rate at which the data is produced. In order to manage big data that are highly interconnected, graph databases are "Unstructured data" and very powerful. Traversal in the graph database is easier and takes less time for highly interrelated data compare to structure database like RDMS. The graph database is highly scalable. There are many famous graph databases like Neo4j, AllegroGraph7, OrientDB8. Social networking websites, semantic web, ontologies, hospital data are highly suitable for storage in graph databases.

A graph consists of nodes and edges that define relations between them. Edges have their own labels and properties. In the graph database, the data is stored as nodes and relationship that are accessed by queries during graph traversal operations. Graph traversal provides more flexibility. It is easy to render the graph database and

expand to very large graph datasets. A simple graph representation has been depicted in the 3.1.

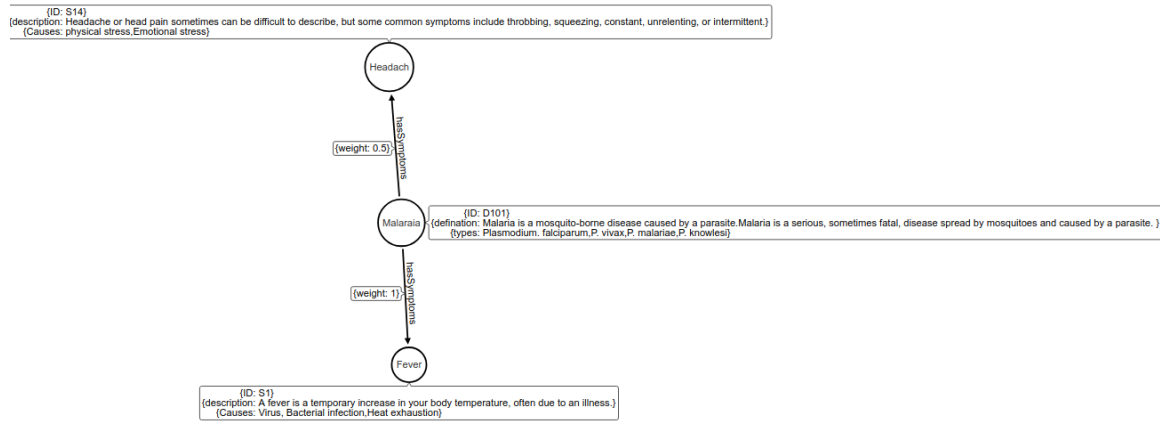


Figure 3.1: Graph Representation

The database shown depicts the relationship between symptoms and diagnosis and how that data is being stored in a database. The nodes have their own properties which are stored in form of key-value pairs. Each node can be given a label to call with later during querying. Same is the scenario with relations. The edges define relationships and relationship types. They can also have properties which are also stored as key-value pairs.

3.6 Data Mining

Data Mining is considered one of the conventional methods to retrieve hidden knowledge from different dimensions from a given dataset. The structure of the dataset where we apply mining determine the success of methods along with the efficiency of its algorithm. Data mining comprises a sequence of steps or combination of methods or mathematical solutions or machine learning procedures to establish a relationship between the raw data in a dataset, with a view providing a global meaning to the

given dataset. Data mining consist of a number of combination of mathematical calculations or machine learning processes to recognize a relationship between the attribute values in a raw dataset that provides the meaning to the given dataset.

Despite advances in computing, faster processors and high-speed networks, the performance of relational database applications is becoming slower and slower. However, the performance of relational database applications is afflicted when it comes to the context of big data. The performance deteriorates not only with high and rapidly growing volume and velocity of data, but also in its variety, complexity, and interconnectedness in the dataset. Real world dataset is usually densely connected and as the volume, velocity and variety of data increase the data relationships also grows even faster.

Experimenting with data mining on different databases other than relational database has abundant benefits. In the data mining process, the relationship of values in the dataset needs to be established with less error. So the graph database comes into the acts, as it could provide a database structure that helps to establish a relationship between the attributes of the dataset. The following different properties are the major significance of the graph database.

- Graph Representation of data
- Establishing the Relationship is easy
- Simplicity in query formulation.
- Data Visualization.
- Interoperability.
- Easy to map different forms of data.
- Easy to design conceptual database.

The major advantage of graph database over relational databases are: The relational database should not contain multivalued attributes however, the graph databases can

have a definite number of multivalued attributes. A single node can have different attributes. A node in a graph database has incoming and outgoing edges considered as indegree and outdegree of the node. These connections are equivalent to the connection between two different tables in RDBMS. It is easy to apply graph algorithm to the graph representation which is more effective and fast.

3.7 Ontology and Resource Descriptive Framework (RDF)

Ontology is used to build an appropriate model of the structure of a system. It includes concepts, concept taxonomies, relationships between concepts, and properties that describe concepts, axioms and constraints to define a domain [17]. There are a large number of languages for expressing ontologies. Some of the standard languages are RDF (Resource Description Framework), OWL (Web ontology language) [18], KL-ONE, vCard [19] and such others. The ontologies are used in the various domain for knowledge management [20] and modeling domain knowledge [21].

RDF concept for structuring the Communication flow. We define a model to structure the workflow by using RDF. The acronym RDF stands for Resource Description Framework. It was initially outlined as metadata data model by World Wide Web (W3C). It is now being used generally as conceptual description or modeling of information enacting in the web semantics. It uses different syntax notations and data serializable format. It is now being used for knowledge management. Since it is a common framework, application designers can leverage the availability of common RDF parser and processing tools. The ability to exchange information between different applications means that the information may be made available to applications other than those for which it was originally created.

Expressions in RDF are represented as triples, in general, consisting of a subject,

a predicate (also called a property) that denotes a relationship to an object. A usual representation of the property that holds between subject and object could be as a row in a table in a relational database. Then the table has two columns, corresponding to the subject and the object of the RDF triple. The name of the table corresponds to the predicate of the RDF triple. Relational databases have an arbitrary number of columns and a row expresses a relationship between entities. Such a row, or relation, has to be decomposed for representation in an RDF triple structure. The patterns and relationship in the graph database - Neo4j has been implemented similarly to RDF format.

Chapter 4

Related Work

In this Chapter, we discuss the related works on different topics such as Formal Concept Analysis, Decision Support System and Data mining that has been done in our research.

4.1 Formal Concept Analysis

Formal Concept is the mathematical model consisting of the theory of lattice and ordered sets that were originally developed by Garrett Birkhoff in 1930s [22]. FCA analyzes data and provides the mechanism to describe the relationship between a particular set of objects and a particular set of attributes. Concept lattice analysis provides a mechanism to identify groups of objects that have common attributes. FCA has been used to identify the relationships between different modules and their attribute values in legacy code. Several techniques for forming modules from legacy code use concept lattice analysis [23, 24], where the lattice is used to identify the relationships between program FCA acquires knowledge and provides visualization to explore the knowledge present in the medical dataset [25, 26]. Machine learning techniques [27] and data mining [28] techniques such as neural networks and decision trees are used for predicting purpose in the large dataset. In paper [29], the author finds the relationship between "diseases" and "symptoms" by representing as objects

4.2 Decision Support Systems

Decision Support Systems is an interesting and widely used area of research since the mid-1970s. DSS is an interactive computer-based system that helps users in judgment and choice activities [30]. In the early stage, DSS consists of the knowledgebase, inference engine, and user support. The knowledgebase is an intelligent database to maintain and retrieve knowledge from related domains to use in inference engine [30]. The inference engine is the part of the expert system which makes logical decisions based on the knowledge about a specific situation. DSSs are defined in various domains such as healthcare (Clinical DSS), organizational decision support system (ODSS), Group decision support system (GDSS), etc. A business decision-making model discussed in [31] supports several aspects in business rules lifecycle and describes a method for extracting business rules from decision support system. Currently, recommendation services are widely popular in E-commerce markets such as Amazon and eBay based on DSS theory to suggest proper services based on individual interests and their navigational behavior. In paper [32], Wen introduced a recommendation service with service-oriented architecture which uses data mining algorithms to analyze customers shopping history. Their approach is based on service-oriented recommendation technologies such as recommendation engines, data mining, content-based approach and collaborative filtering. Their approach to recommend products is content-based that is based on associations between products.

Medical practice mostly relies on the available scientific evidence and clinical guidelines that are used for the recommendation for a large group of patients. CDSS proposed before are not very accurate and limited since they usually don't consider the specific characteristics of the patient and don't provide personalized clinical recommendation [33, 34, 35]. Clinician often refers to the medical knowledge available

through medical guidelines sites like Pubmed, UpToDate to find possible conditions and recommendation decisions but these resources are not customized to specific patient conditions.

Several CDSS are implemented in the decision-making process. Some of them are "WizOrder" - helped to reduce medical errors with clinical decisions during order entry with the features such as restructuring clinical workflows, providing relevant educational materials [36], "ATHENA" that provides the clinical decision for hypertension that is evidence-based to recommend drug therapies and control their impact on blood pressure [37]. CDSS are implemented in clinical settings however their evaluation might not be different. Some of the research indicates CDSS improves the clinical practice [38, 39] whereas some indicate CDSS doesn't improve the outcomes [40]. Data-driven approaches in big data are more often compare to personalized healthcare decision support system [41]. CDSS are implemented with different machine learning approaches. For an instance, a number of algorithm such as rule-based approach [42], SVM [43] and artificial neural networks [44]. Although some CDSS issue accurate diagnostic recommendations for specific diseases, our CDSS uses the Apriori algorithm to find the patterns and relationships between the patterns from the patient symptoms and diagnosis. These patterns and relationship along with the expert knowledge are integrated to provide the recommendation for the more appropriate diagnosis. With this approach, our CDSS updates the context with patients symptoms and recommends all the diagnosis related to the context.

4.3 Data Mining and Artificial Intelligence (AI)

Applying data mining and AI techniques on EHR data creates many opportunities for improving delivery, efficiency, and effectiveness of different sector of health care [45]

such as operations management, chronic disease treatment and prevention, association analysis, preventive health care, evidence-based treatment, and population tracking. There are mainly two other sources for knowledge. The first source is knowledge from experts in the form of guidelines. These guidelines are created by using many methods, such as systematic reviews and Meta-analysis. The second source is the application of data mining techniques on EHR data. EHR contains a very large and historical dataset that changes continuously and contains useful hidden knowledge. These hidden knowledge are extracted using data mining and AI services into the active CDSS to continuously update its knowledgebase by the most recent patterns.

Chapter 5

Approach

In this Chapter, we discuss the approach for knowledge discovery from medical health records of patients with data mining techniques.

Decision support system is the system that helps during decision making to the user. Different types of decision support system in different areas use knowledge to provide an important decision in order to solve the problem. In our research, we extracted the knowledge for decision support system similar to the Consultant as a Service (CAAS) [1]. Consultant as a Service tool that helps in decision making to the user who is unknown about the system and to get some services. The consultant as the service performs its services with semantic analysis, data mining and cloud-based concept [46].

We introduced a novel concept that can help doctor, nurse or health practitioner for making a decision for personalized health with data mining approach and graph database - Neo4j. Data mining results to discover patterns and the connection between these patterns in the data. Patterns are represented in the graph as nodes and the connections are represented as the relationship in Neo4j.

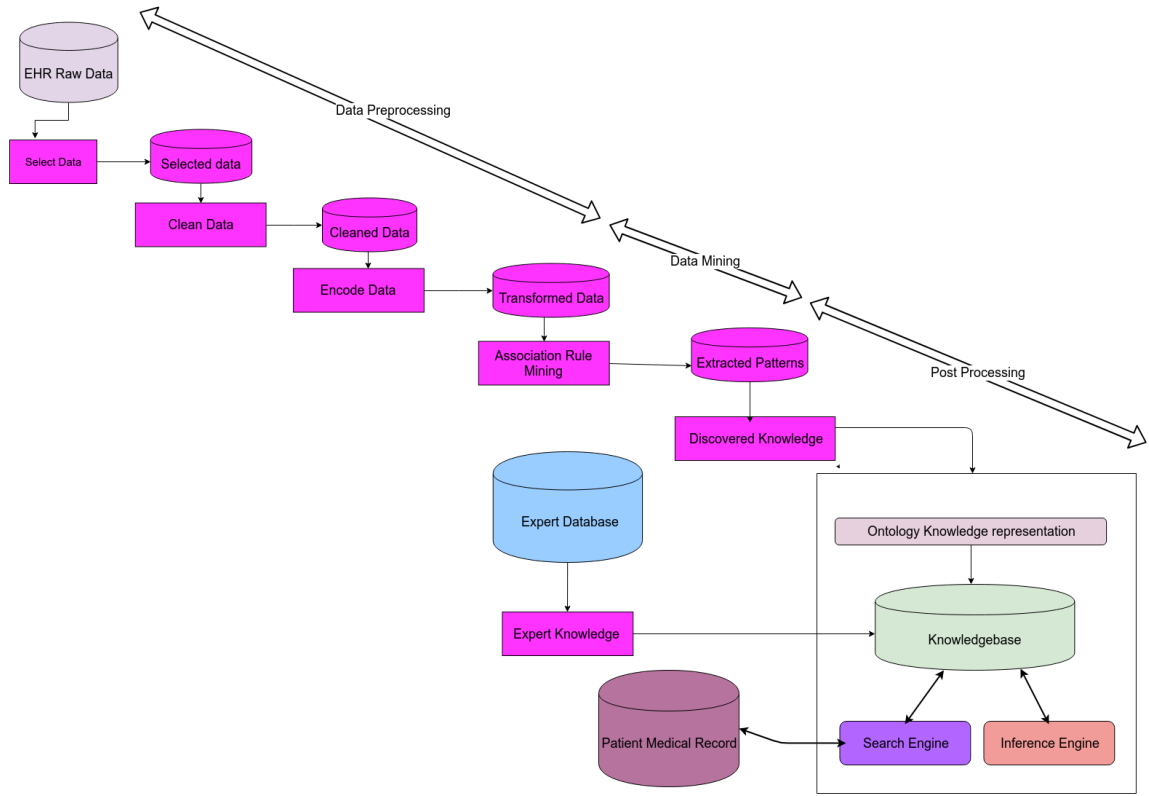


Figure 5.1: Knowledge Discovery Process

5.1 Association Mining

The historical records of patients dataset from EHR system consists of information with different attributes such as patient name, age, gender, symptoms, diagnosis, and medication. Extraction of knowledge from the EHR data needs series of processes as shown in Figure 5.1 (detail discussion on Chapter 7). First of all, the data selection process helps to select the interested attributes to consider for the data mining process. Symptoms and diagnosis are selected from the EHR data to determine patterns. Then the data cleaning is performed to remove records with unknown values in EHR data. Cleaned data are then transformed into encoded form for data mining process. This transformed form is fed to the data mining algorithm for extracting patterns and

knowledge.

Data mining results in the frequent itemset. The itemset is a collection of one or more items. We obtain itemset - groups of symptoms sharing the common set of diagnoses known as baskets where support of the itemset is greater than or equal to the minsup threshold. Association rules are in the of the form implication expression $X \Rightarrow Y$, where X and Y are itemsets.

The number of frequent items and the quality of association rules is maintained with the help of support and confidence. The support is the ratio of the number of item or item set in data set to the number of transaction that contains the item or itemset. It's a relative frequency of item or itemset in the transaction. The association rule with high support is more reliable as the number of present items or item set is more reliable. In our approach low support is used to retrieve a hidden relationship in a data set.

We apply the Apriori algorithm on our dataset that generates frequent itemsets and association rules from frequent itemsets satisfying user support. The related diagnosis and shared attribute values by these diagnoses represent the common context.

We obtain maximum associated events which share the same set of attributes and we call these events as "baskets" and shared attribute value by these events as "itemset". We name this whole group where we have the baskets with its itemset as MAG (Maximum Association Group). The frequent itemsets can result in many MAGs with different itemsets which might be the subset of some other large frequent itemset. We remove these subsets and find MAG with the larger basket and larger itemset. This process removes all of redundant MAGs and eliminates overlap between two MAG.

For instance, if we have frequent itemsets with attribute values $\langle s1, s2, s3 \rangle$ as itemset which falls in baskets $\langle d1, d2 \rangle$ and other frequent itemset with attribute

values $\langle s1, s2 \rangle$ as itemset and $\langle d1, d2 \rangle$ as baskets then we remove the $\langle s1, s2 \rangle$ itemset for the baskets $\langle d1, d2 \rangle$. If we consider with health data example, MAG with all the group of patients under diseases as "baskets" $\langle \text{Malaria, Typhoid} \rangle$ having symptoms as itemset $\langle \text{Cold, Cough} \rangle$, is subset of larger MAG with group of patients having symptoms $\langle \text{Cold, Cough, Fever} \rangle$ as itemset under same "baskets" $\langle \text{Malaria, Typhoid} \rangle$. We remove subset MAG and only keep larger MAG.

Association mining results in a number of MAGs consisting of all the events with baskets of diagnosis and its itemsets of symptoms. Each MAG with k frequent itemset containing k number of symptoms and corresponding baskets of diagnosis. During the examination of a patient, health personal collects symptoms. These collected symptoms with historical symptoms of the patient are matched with equivalent k number of symptoms in k - frequent itemset. More than one itemset could be matched with the k number of symptoms in k itemsets, therefore our decision support engine will recommend next highly probabilistic symptoms and corresponding diagnosis for $k+1$ itemsets in descending order. $k+1$ itemsets consist of one additional symptom compare to k itemsets. Physician chooses the most appropriate additional symptom that matches with respect to the patient context. Further our decision support engine searches for the next possible MAGs with $k+2$ itemsets and corresponding diagnosis. With the increase in itemset in MAG, the diagnosis in MAG decreases and are more specific and appropriate. This process is continued till we reach the final MAG to result in a more specific and appropriate diagnosis for the patient. In this way, our decision support engine helps doctors, nurse, physician and health practitioner for making the proper decision during the examination of a patient. Following are different MAG's examples with the different number of itemset as shown in Figure 5.2.

Frequent 2 itemset

$[itemset : \langle s1, s2 \rangle, baskets : \langle d1, d2, d3, d4, d6 \rangle]$

[*itemset* :< *Headache, Fever* >, *baskets* :< *Sarcoidosis, Tuberculosis, Cataracts, RootInfection, AcuteBronchitis* >]

Frequent 3 itemset:

[*itemset* :< *s1, s2, s3* >, *baskets* :< *d1, d2, d3* >]

[*itemset* :< *Headache, Fever, Lossof Appetite* >, *baskets* :< *Sarcoidosis, Tuberculosis, Cataracts* >]

[*itemset* :< *s1, s2, s4* >, *baskets* :< *d1, d2, d4, d6* >]

[*itemset* :< *Headache, Fever, Fatigue* >, *baskets* :< *Sarcoidosis, Tuberculosis, RootInfection, AcuteBrochitis* >]

Frequent 4 itemset:

[*itemset* :< *s1, s2, s3, s5* >, *baskets* :< *d1, d2* >]

[*itemset* :< *Headache, Fever, Lossof Appetite, BloodinCough* >, *baskets* :< *Sarcoidosis, Tuberculosis* >]

[*itemset* :< *s1, s2, s3, s55* >, *baskets* :< *d1, d3* >]

[*itemset* :< *Headache, Fever, Fatigue, BlurVision* >, *baskets* :< *Sarcoidosis, Cataracts* >]

[*itemset* :< *s1, s2, s4, s6* >, *baskets* :< *d1, d6* >]

[*itemset* :< *Headache, Fever, Fatigue, PersistentCough* >, *baskets* :< *Sarcoidosis, AcuteBronchitis* >]

[*itemset* :< *s1, s2, s4, s187* >, *baskets* :< *d1, d4, d74* >]

[*itemset* :< *Headache, Fever, Fatigue, Toothache* >, *baskets* :< *Sarcoidosis, RootInfection, OralCancer* >]

The ontology is created by using graph concepts to traversal from one MAG to other as shown in Figure 5.2. The confidence is defined to find the degree of relevance while traveling from one MAG to another MAG. The graph consists of nodes and relationship. The MAGs are represented as nodes and relationships between them are represented by extra symptoms contained by next successor MAG. The following formula is used to calculate the confidence between MAGs.

$$Weight = \frac{\text{number of baskets (less number of diagnosis) in successor MAG}}{\text{number of baskets (more number of diagnosis) in preceding MAG}}$$

This degree of relevance helps to determine the converging and predicting power while taking the decision. If the value of confidence is low, it may converge more and results in a more specific diagnosis.

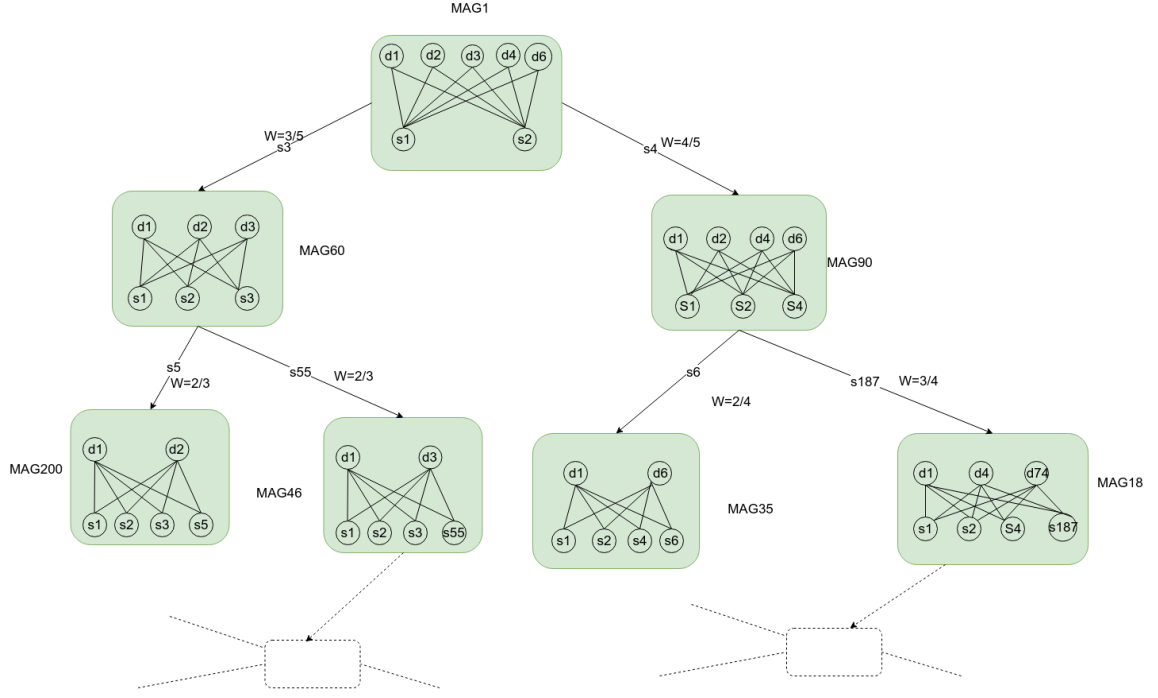


Figure 5.2: MAG Representation

Our approach is responsive, provides the most appropriate decision to determine the accurate result. It makes the decision by updating the context with the user's provided answer. So when the doctor, nurse or any other health practitioner provides the symptoms from the historical data along with new symptoms of the patient, then it searches for the MAG which contains the itemset that matches the context. MAGs are connected with other MAGs with different symptoms supported by the degree of relevance as weight. Doctor choose the transition among all connected MAGs and receives the next symptoms according to the new context.

For an instance, in Figure 5.2 during the examination of the patient if the patient without historical data has common symptoms $s1$ (Headache) and $s2$ (Fever) then it matches the MAG1 having diagnosis $< d1$ (Sarcoidosis), $d2$ (Tuberculosis), $d3$ (Cataracts), $d4$ (Root Infection) and $d6$ (Acute Bronchitis) $>$ in the knowledge

graph. There are two possible option either the next symptom is s3 (Loss of Appetite) and update its context to MAG60 with diagnosis <d1 (Sarcoidosis), d2 (Tuberculosis),d3 (Cataracts) > or the next symptom is s4 (Fatigue) and update its context to MAG90 with diagnosis < d1 (Sarcoidosis), d2 (Tuberculosis),d4 (Root Infection) and d6 (Acute Bronchitis) >. The health personal decides to choose s4(Fatigue) as it is more appropriate to the patient condition that updates the context to MAG90. The health personal receives the next question with two new possible symptoms s6 (Persistent Cough) and s187 (Toothache) that matches MAG35 and MAG18 with possible diagnosis < d1 (Sarcoidosis), d6 (Acute Bronchitis) > and < d1 (Sarcoidosis),d4 (Root Infection), d74 (Oral Cancer) > respectively. The health personal chooses symptom s6 (Persistent Cough) to be more appropriate symptom to the patient condition reaching the final MAG and indicates the diagnosis < d1 (Sarcoidosis), d6 (Acute Bronchitis) > diagnosis to be most appropriate with the patient context. The health personnel could recommend diagnosis related to lungs. On the other hand, if the health personal choose s187 (Toothache) that match MAG18 with diagnosis < d1 (Sarcoidosis), d4 (Root Infection), d74 (Oral Cancer) > related to dental diagnosis and could travel further to other connected MAGs that could result in more appropriate sets of diagnosis with the context provided. In the beginning only, if the health personal decides to choose symptom s3 (Loss of Appetite) to be more appropriate instead of symptom s4 (Loss of Appetite) then it matches the MAG60 with the diagnosis < Sarcoidosis, Tuberculosis, Cataracts > and repeats the same process with next possible symptoms s5 (Blood in Cough) and s55 (Blur Vision) and further traveling to other MAGs: MAG200 with diagnosis < d1 (Sarcoidosis), d2 (Tuberculosis) > and MAG46 with diagnosis < d1 (Sarcoidosis), d3 (Cataracts) >. The process continues till it reaches the final MAG and receives the most appropriate diagnosis with patient context. In the traversal process, we move from the MAGs with less

number of symptoms - "itemset" and a large number of diagnosis- "baskets" towards the MAGs with more "itemset" but less "baskets". This process results in shrinkage of the basket matching more number of the itemset. The process continues until there are no matches symptoms in the knowledge graph.

Chapter 6

Architecture

In this chapter, we discuss the architecture and working mechanism of our Clinical Decision Support System.

Once the knowledgebase is constructed by creating an ontology that represents groups of symptoms and diagnoses in MAG integrated with the knowledge from the expert database, the health personnel such as doctor, nurse and health practitioner could query the knowledgebase with symptoms of the patient to the CDSS. The symptoms are treated as a current context for the patient that is referred by the search engine. The search engine matches the MAG in the knowledgebase with the context and displays all the possible diagnosis that belongs to the patient. Our knowledgebase consists the mined knowledge of other patients record with same symptoms and diagnosis with the patient. This helps in the decision-making process. This knowledge could be visualized by querying the graph database- Neo4j. We developed an architecture that could assist the health personnel during decision-making process by retrieving the most appropriate diagnosis that matches with the current context of the patient providing more information about the diagnosis and the symptoms integrated from the expert database to our knowledgebase.

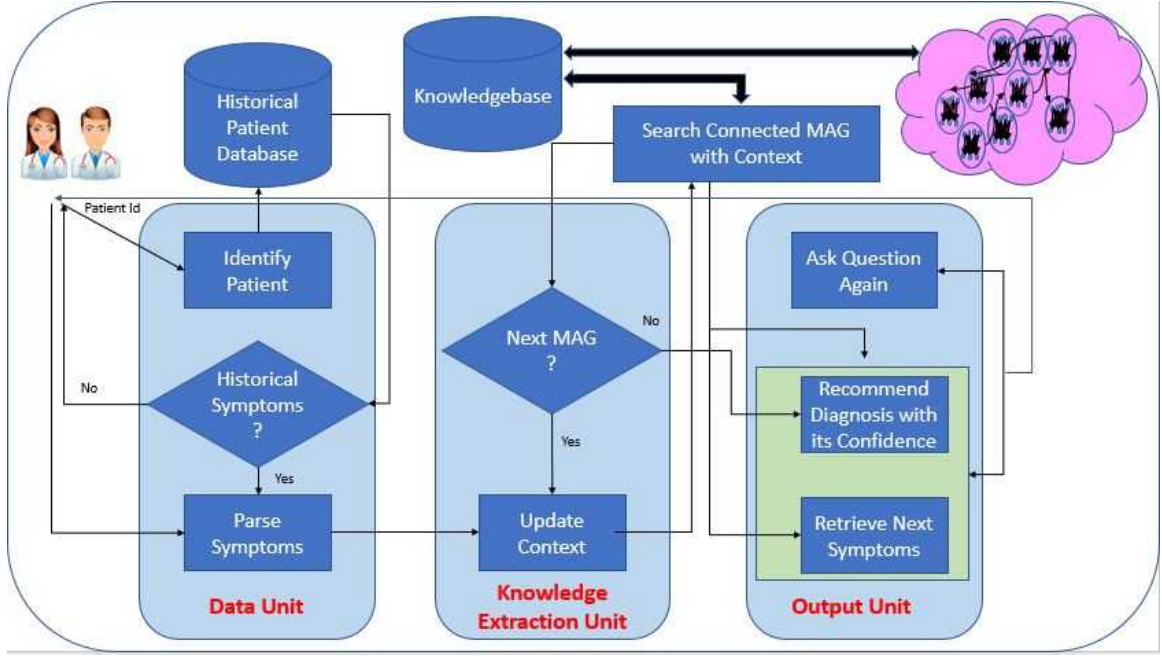


Figure 6.1: CDSS Architecture [1]

The architecture of CDSS is represented in Figure 6.1. The architecture consists of 3 different parts.

1. **Data Unit:** This unit interacts with the user and provides the user interface. First of all the health personnel as a user, input the PatientID corresponding to the patient under examination in order to look through the symptoms present in the historical database. There is no historical record of the patient if the patient is recently admitted with their new patientID. Our system identifies this condition and notifies the user that the patient has no historical record and is newly admitted. In another case when the patient already has historical records then our system captures all the historical symptoms and notify the user about it. The user is then asked to enter new symptoms of the patient separated by a comma and "and" conjunction. These historical symptoms and new symptoms provided by the user are collectively parsed to get all unique

symptoms related to patient and considered as the context of the patient. This context is used by the Knowledge Extraction unit of CDSS for matching the appropriate MAG which matches with the context.

2. Knowledge Extraction Unit: The knowledge extraction unit basically works through two mechanisms:

- (a) Matching the Appropriate MAG with Context:

The context of the data unit is considered as the updated context in the knowledge extraction unit. The knowledge extraction unit extracts the knowledge from the knowledgebase with the updated context. The updated context is searched in the knowledgebase to find the appropriate MAG. The search engine searches the appropriate MAG and returns the details such as the name of MAG, all the diagnoses of matched MAG, all the connected MAGs, the weight of their connection, list of possible symptoms contain in connected MAGs and list of possible symptoms contain in connected MAGs.

- (b) Determining and Choosing next Connected MAG: When the system displays the next symptoms with next possible MAGs and the possible diagnosis respective to MAGs in descending order to its weight, the user gets a prior knowledge to choose more appropriate symptoms among the listed symptoms. The weight sorts the next symptoms according to the converging relevance of MAG. The user could know which connected MAG converges to less number of diagnosis if that symptom is chosen by the user. The user chooses the next symptom by conforming to the patient whether he/she has that symptom. The inference engine in Knowledgebase infer the next connected MAG with the chosen symptom and displays the details

of the preceding MAG with diagnosis and symptoms. The next symptom entered by the user through the recommendation of CDSS and prior symptoms hold by patient_Symptoms list collectively update the context to find next connected MAGs. The process continues until the user exits by terminating the process or continuing until the final MAG is reached to get final precise diagnosis corresponding to the final updated context.

3. Output Unit: This unit interacts with the user to ask about the next appropriate symptoms through the list of the possible symptoms. This unit is also responsible to display the user recommendation of diagnosis with its current context and provides the detail of next possible connected MAGs with the confidence value. This unit provides detail information on MAGs diagnosis and symptoms.

To illustrate the working mechanism of the architecture we develop an Algorithm 1 that depicts how our CDSS helps health practitioners during the decision-making process utilizing the knowledge from the knowledgebase. In line 2, we represent and store the knowledge in the form of a graph. All the MAGs with symptoms and diagnosis, integrated information from the expert database and the relationship between them are represented in the graph database-Neo4j using cypher query. This results in the formation of ontologies where the symptoms and diagnosis are connected with different MAGs with additional information in their attributes. The MAGs are interconnected with other MAGs by finding the additional symptoms connecting the MAG. This knowledgebase is built offline so that the processing speed is high while querying the graph database.

The patient ID of the patient is entered by the health personnel to retrieve the historical database as shown in line 3. The patient_Symptoms variable stores the

Algorithm 1 Working of CDSS

```
1: procedure ALGORITHM
2:   Create Graph  $G(MAG, relation\_list[relations])$ 
3:   Patient_ID = ID of patient
4:   patient_Symptoms = refer_Historical_Database(Patient_ID)
5:   if patient_Symptoms is empty : then
6:     display("No historical record for the patient")
7:   new_Symptoms = new symptoms of patient
8:   patient_Symptoms = patient_Symptoms + new_Symptoms
9:   next_symptoms = Search_MAG(patient_Symptoms) in  $G(N, E)$ 
10:  display(next_Symptoms)
11:  travel_MAG = empty list
12:  Status = True
13:  Top :
14:  while (Status==True) : do
15:    choice = input("Do you like to continue further Y/N")
16:    if choice== "N" : then
17:      display(travel_MAG)
18:      Status = False
19:      Goto Top
20:    else:
21:      choice_symptoms = user chooses Symptoms from next_symptoms list
22:      patient_Symptoms.append(choice_symptoms)
23:      if choice_symptoms not valid : then
24:        display("Enter symptom again, not correctSymptom!")
25:        GotoTop
26:      else:
27:        next_symptoms, MAG = Search_MAG(patient_Symptoms)
28:      if next_symptoms not empty : then
29:        travel_MAG.append(MAG)
30:        display(next_symptoms)
31:      else:
32:        display("Final MAG, No further connection")
33:        exit()
```

list of historical symptoms of the particular patient in line 4. If there is no record of the patient in the historical database then CDSS prompts with the notification that "No historical record for the patient" in line 5 and 6. New symptoms of the patient are then entered in line 7. These new symptoms along with the historical

symptoms of the patient are used as the context in line 8. This context is provided to the search engine to determine appropriate MAG in Knowledgebase in line 9. The inference engine in the knowledgebase finds all the possible connected MAG with the current context. All the possible connected MAGs consist of one additional symptom compares to the MAG that matches the context. Since MAG with the context could be connected to many other MAGs, the weight attribute value of each connected MAG node shows the correlation between two MAG in Neo4j. The CDSS displays all the possible next_Symptoms corresponding to all connected MAG in descending order of their weight in line 10. This provides user to know which diagnosis converges to what degree with the current context.

To trace the MAGs traversal, a travel_MAG stores all the MAGs traveled so far in the knowledgebase in line 11. The status Flag and Top label are referred to control the flow of the system in line 12 and 13. The CDSS provides the user option to terminate the search process anytime when they determine appropriate diagnosis to the patient in line 14 and 15. Setting the choice flag to "N" stops the process updating status to False and exits further traversal by displaying the traveled MAGs so far as shown from line 16 to 19. If the choice flag is not "N" then the user enters one of the symptoms recommended by the possible connected MAGs. If the user enters invalid symptoms not present in connected MAG, CDSS alert the user with the message "Enter Symptom again, Not correct Symptom!" as shown from line 23 to 25. The entered symptom is added with patient_Symptoms list and the context is updated. The updated context is used by the search engine to determine the appropriate MAG and next_symptoms as shown in line 26 and 27. The new possible symptoms are displayed along with possible MAGs, its symptoms and diagnosis information as shown in line 28 to 30. In this way, our system provides the prior possible scenario of the user decision by displaying the knowledge of next possible symptoms and

diagnosis for health personnel that helps in the decision-making process. The CDSS continues this process until there are no connected MAG and next_symptoms in the knowledgebase. When the user reaches final MAG the system notifies the user with the message "Final MAG, No further Connection" as shown in line 31 to 34.

Chapter 7

Experimentation

In this section, we discuss the dataset that is used and the experimentation process performed during the development of the knowledgebase.

7.1 Dataset

In our research, we used open dataset <https://www.kaggle.com/plarmuseau/symptom-disease-recommender> uploaded by Paul Larmuseau. This dataset consists of diagnosis and symptoms related to eye domain. The data was represented in the form of sparse knowledge matrix where row index was symptoms and column index was the diagnosis. Row and column index were encoded to their respective symptoms and diagnosis code, whose description was provided in different data dictionary files. There were 131 symptoms and 110 diagnoses in the dataset. The nonzero elements for each column collectively give the symptoms corresponding to the diagnosis. The description of the dataset are discussed as below:

1. *dia.t.csv*

This file was the data dictionary for the diagnosis in our dataset. The file consists of the diagnosis id and its description in comma separated file (CSV). This file was converted into the two-dimensional data structure as the DataFrame python data manipulation library called Pandas [47]. This file was used to map the diagnosis code and diagnosis description. Figure 7.1 is the sample rows of this file.

	did	diagnose
0	1	Abdominal aortic aneurysm(enlarged major blood...
1	2	Abdominal swelling
2	3	Abdominal trauma
3	4	Abrasions (scrapes)
4	5	ACE inhibitor induced cough blood pressure med...
5	6	acetaminophen overdose Adverse reaction to ace...
6	7	Tylenol acetaminophen poisoning
7	8	Achilles tendonitis (heel tendon inflammation)
8	9	Achilles tendon rupture (heel tendon tear)
9	10	Acid LSD abuse
10	11	Acidosis (excessive acid in the body)

Figure 7.1: Diagnosis Dictionary

The data type for each field in this metadata file is as below:

- *did* → Numeric
- *diagnosis* → String

2. *sym_dis_matrix.csv*

This file was in CSV format. It consists of a row index with symptoms and column index with diseases. This file was converted to DataFrame for performing data manipulation operation. The data was transformed and fed to the data mining algorithm. Data was represented into the matrix format with different numeric values indicated as below. The raw dataset is shown in Figure 7.2.

- 0.0 → not presented
- 1.0 → common
- 2.0 → life-threatening
- 3.0 → common pediatrics

	eye	5.0	14.0	25.0	26.0	71.0	82.0	85.0	98.0	107.0	...	1365.0	1367.0	1395.0	1397.0	1461.0	1473.0	1475.0	1479.0	1483.0	1511.0
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	2.0	0.0	0.0	3.0	3.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	4.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	5.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	9.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	11.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	13.0	0.0	0.0	4.0	4.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	15.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	17.0	3.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	19.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 7.2: Raw Dataset

3. *sym_t.csv*

This file was the data dictionary for the diagnosis in our dataset. It consists of the symptom Id and its description in comma separated file(CSV). It was converted into DataFrame and used to map symptom code and symptoms description. Figure 7.3 is the sample row of this file. The data type for each field in this file are as below:

- *syd* → Numeric
- *symptom* → String

syd		symptom
0	1	Upper abdominal pain
1	2	Lower abdominal pain
2	3	Abscess (Collection of pus)
3	4	Alcohol abuse
4	5	Anxiety (Nervousness)
5	6	Arm ache or pain
6	7	Back ache or pain
7	8	Bleeding tendency
8	9	Blood in vomit
9	10	Bloody diarrhea
10	11	Pain or soreness of breast
11	12	Calf pain
12	13	Chest pressure
13	14	Chills
14	15	Change in behavior
15	16	Constipation

Figure 7.3: Symptoms Dictionary

7.2 Preprocessing

In order to apply data mining in our dataset, the data should be selected, cleaned and transformed so that it is applicable to feed to data mining algorithm as shown in the Figure 5.1. We checked incompleteness of data, noisiness (containing errors or duplicates) or inconsistent (containing discrepancies) in data. The incompleteness checking includes lacking values and lacking attributes of interest, noisiness includes checking duplicates and errors and inconsistency whether the values assigned provides the same meaning across all diagnosis and symptoms. Encoding of diagnosis and symptoms are performed to represent diagnosis as D1, D2,, D109, D110, and symptoms as S1, S2,S130, S131. The sparse dataset was transformed into the new dataset which contained only symptoms and diagnosis in the encoded form for each diagnosis. The Figures 7.4 and 7.5 show before and after transformation of the old and new dataset.

	S	D4	D13	D23	D24	D67	D78	D81	D93	D100	...	D1110	D1134	D1139	D1140	D1141	D1142	D1152	syd	symptom	symp_index
0	S0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	Upper abdominal pain	S0
1	S1	0.0	0.0	3.0	3.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2	Lower abdominal pain	S1
2	S3	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4	Alcohol abuse	S3
3	S4	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5	Anxiety (Nervousness)	S4
4	S5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6	Arm ache or pain	S5
5	S6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7	Back ache or pain	S6
6	S7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8	Bleeding tendency	S7
7	S8	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9	Blood in vomit	S8

Figure 7.4: Annotated Dataset

	1	2	3	4	5	6	7	8	9	10	...	15	16	17	18	19	20	21	22	23	24
0	S16	S153	S181	S182	D4	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	S25	S111	S187	D13	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	S1	S3	S4	S8	S12	S14	S18	S20	S27	S37	...	S120	S165	S168	S175	S228	S229	S230	S248	D23	NaN
3	S1	S12	S16	S20	S27	S45	S51	S80	S87	S137	...	S195	S200	S203	S212	S213	S214	S215	S217	S225	D24
4	S58	S61	S168	D67	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	S21	S103	S110	S157	D78	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	S25	S57	S60	S152	S177	S189	S190	D81	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	S25	S154	S182	D93	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	S20	S40	S86	S93	S103	S110	S111	S157	S231	S266	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	S25	S265	D129	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	S93	S111	S265	S266	D136	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	S14	S20	S27	S47	S48	S68	S86	S93	S97	S103	...	S165	S228	S229	S230	S231	S234	S264	S265	D138	NaN
12	S112	S168	D152	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	S0	S1	S6	S28	S31	S53	S112	S168	S246	S248	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 7.5: Transformed Dataset

The symptoms present in all diagnosis were categorized into different buckets with respect to the number of symptoms present. The following bar chart represented in the Figure 7.6 shows the symptoms bucket and their frequency.

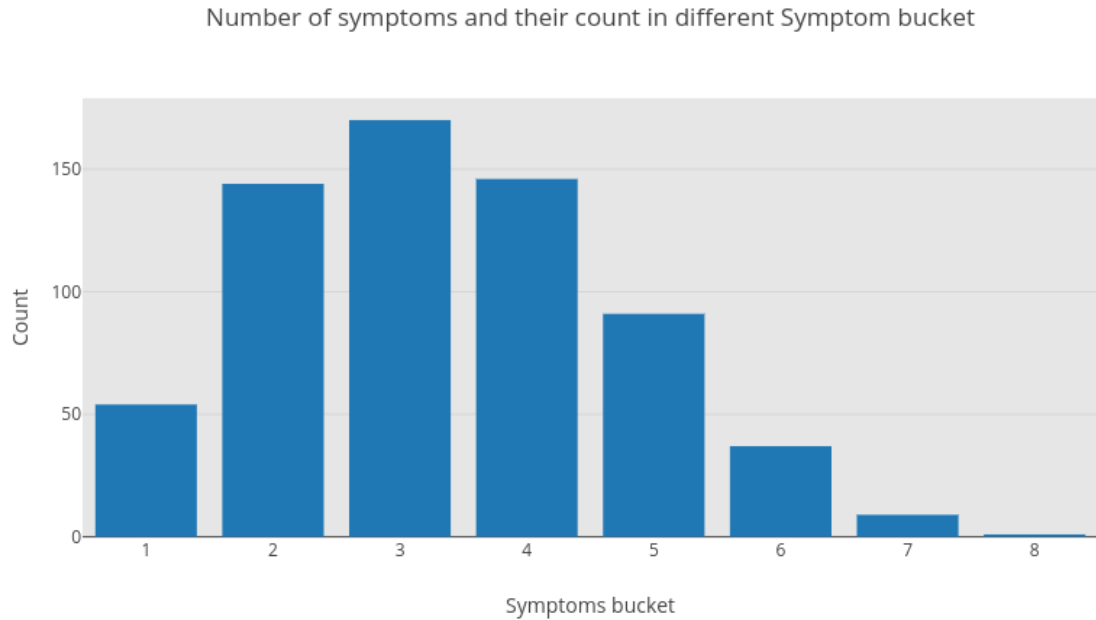


Figure 7.6: Symptoms Bucket

7.3 Data Mining and Pattern Generation

The new transformed dataset was fed to the Apriori algorithm. The algorithm generates all the frequent itemsets and association rules. The number of frequent itemsets depends upon the minimum support provided as the parameter to the Apriori algorithm. We experimented with taking different values of minimum support and confidence. The Table 7.1 shows the number of the frequent items and association rules generated with the variation of minimum support count and confidence. The Table 7.2 shows the sample of the frequent item generated.

Data Mining Parameters			
Minimum Support	Confidence	FrequentItems_Count	Rules_Count
0.02	1	652	5592
0.02	0.8	652	5610
0.03	1	166	178
0.03	0.8	166	196
0.04	1	73	25
0.04	0.8	73	43

Table 7.1: Data Mining Parameters Variation

The Table 7.1 clearly shows, with the increase in minimum support count the frequent itemset decreases and with the increase in confidence the generated rules also decreases.

Itemset	Minimum Support
(S8)	0.0273
(S129)	0.0273
(S137)	0.0273
(S16)	0.0273
(S26)	0.0273
(S12)	0.0273
(S142)	0.0273
(S175, S228)	0.0273
(S12, S1)	0.0273
(S25, S21)	0.0273
(S103, S234)	0.0273
(S20, S120)	0.0273
(S142, S228)	0.0273
(S264, S20)	0.0273
(S142, S11)	0.0273

Table 7.2: Sample Frequent Itemset

We wanted to capture almost every frequent itemsets in data sets. Thus, in order to capture many frequent itemsets that resemble different patterns in our data, we chose the minimum support of 0.02 and confidence of 0.8. These frequent items were used to generate all the MAG with the baskets of "diagnosis" that shared common symptoms. Algorithm 2 discussed in the Section 7.4.1 is used for creating the MAG.

7.4 Post Processing

The patterns and relationships obtained from the data mining process were used to create the knowledgebase. All the processes needed for creating knowledgebase were performed in the post-processing phase. This section discusses the detail of it.

7.4.1 MAG Creation

The frequent itemsets resulted from the Apriori algorithm and new transformed dataset was passed as the input parameter for the create_MAG function. For every symptom in frequent itemset, it checks whether it is the subset of new_transformed_data. If so, diagnosis from new_transformed_data and symptoms from frequent item set were added into MAG. The Figure 7.7 shows MAGs with diagnosis as baskets and symptoms as itemsets and their cardinality.

Algorithm 2 MAG Creation

Input: Frequent_itemset and new_transformed_data

Output: MAG consisting baskets as list of diagnosis and itemset as list of symptoms

```
1: procedure CREATE MAG
2:   declare MAG as empty list
3:   convert all the frequent_itemset to set
4:   for pattern in frequent_itemset: do
5:     itemset = emptylist
6:     basket = emptylist
7:     for item in new_transformed_data: do
8:       if pattern subset of item[symptoms] : then
9:         if item [diagnosis] not in basket: then
10:          basket.append(diagnosis)
11:       itemset.append(pattern)
12:       itemset.append(basket)
13:       if itemset not in MAG: then
14:         MAG.append(itemset)
return MAG
```

	Node	Symptoms	Diagnosis	Diagnosis_Count	Symptoms_Count
300	MAG300	['S86', 'S184', 'S120']	['D786', 'D1046', 'D448']	3	3
301	MAG301	['S68', 'S142', 'S231']	['D203', 'D138', 'D448']	3	3
302	MAG302	['S27', 'S93', 'S111']	['D804', 'D1070', 'D138', 'D725']	4	3
303	MAG303	['S234', 'S111', 'S157']	['D1075', 'D138', 'D725']	3	3
304	MAG304	['S68', 'S93', 'S157']	['D138', 'D1075', 'D448']	3	3
305	MAG305	['S103', 'S111', 'S86']	['D1070', 'D100', 'D1109', 'D138', 'D448']	5	3
306	MAG306	['S111', 'S93', 'S40']	['D435', 'D1070', 'D100']	3	3
307	MAG307	['S234', 'S93', 'S111']	['D1075', 'D138', 'D725']	3	3
308	MAG308	['S165', 'S142', 'S228']	['D203', 'D138', 'D448']	3	3
309	MAG309	['S230', 'S14', 'S229']	['D23', 'D138', 'D448']	3	3
310	MAG310	['S20', 'S228', 'S229']	['D23', 'D138', 'D448']	3	3
311	MAG311	['S103', 'S93', 'S157']	['D100', 'D725', 'D138', 'D1075', 'D448']	5	3
312	MAG312	['S14', 'S20', 'S229']	['D23', 'D138', 'D448']	3	3
313	MAG313	['S165', 'S175', 'S229']	['D23', 'D203', 'D448']	3	3

Figure 7.7: MAGs with Itemsets and Baskets

7.4.2 Finding the Relationships among the Patterns

The MAGs consist of itemset - list of symptoms sharing the same diagnosis - "baskets" represent patterns in our data. One pattern might be related and could be the subset of another pattern. We developed an algorithm to find the subset among different patterns that could find all the relationship among related patterns.

The itemset consists of a set of symptoms, so we were interested in finding the relationship with other symptoms between different MAGs. All the possible relationship between the MAGs were discovered by finding the additional symptom that differs between two MAG. The additional symptom established the relationship between two MAG. The Figure 7.8 shows the sample relationship discovered between different MAGs.

1	From	To	Initial	Final	Relation
2	'S40'	'S21','S40'	MAG0	MAG82	'S21'
3	'S40'	'S264','S40'	MAG0	MAG148	'S264'
4	'S231'	'S231','S229'	MAG1	MAG174	'S229'
5	'S122'	'S122','S115'	MAG3	MAG133	'S115'
6	'S246'	'S246','S112'	MAG4	MAG101	'S112'
7	'S246'	'S248','S246'	MAG4	MAG105	'S248'
8	'S246'	'S0','S246'	MAG4	MAG160	'S0'
9	'S212'	'S212','S217'	MAG5	MAG75	'S217'
10	'S212'	'S195','S212'	MAG5	MAG106	'S195'
11	'S185'	'S122','S185'	MAG7	MAG80	'S122'
12	'S184'	'S184','S120'	MAG9	MAG110	'S120'
13	'S165'	'S165','S229'	MAG10	MAG67	'S229'
14	'S165'	'S165','S142'	MAG10	MAG114	'S142'
15	'S165'	'S165','S175'	MAG10	MAG124	'S175'
16	'S165'	'S165','S231'	MAG10	MAG153	'S231'

Figure 7.8: Relationship Between MAGs

The MAG0 consists of symptom [s40] was related to MAG82 that consist of symptoms [s21, s40] with additional symptom s21 compare to MAG0. The symptom s21 creates a relationship between these two MAGs. Similarly, from the above table symptom, s264 creates the relationship between MAG0 and MAG148.

7.4.3 Creating a Knowledgebase

The generated MAG and obtained relationships were then used to create a connected knowledge graph. Graph database that is used to represent real word data known as Neo4j to create a knowledge graph. MAGs were represented in the form of nodes of a graph. The baskets of diagnosis and itemsets of symptoms were also represented in form of nodes. The connection between MAGs and its corresponding symptoms was established by the relationship with label name "hasSymptom", whereas the connection between MAGs and its corresponding diagnosis was established by the relationship with the label "hasDiagnosis". The connection between two MAGs was established by the relationship called "connectedTo".

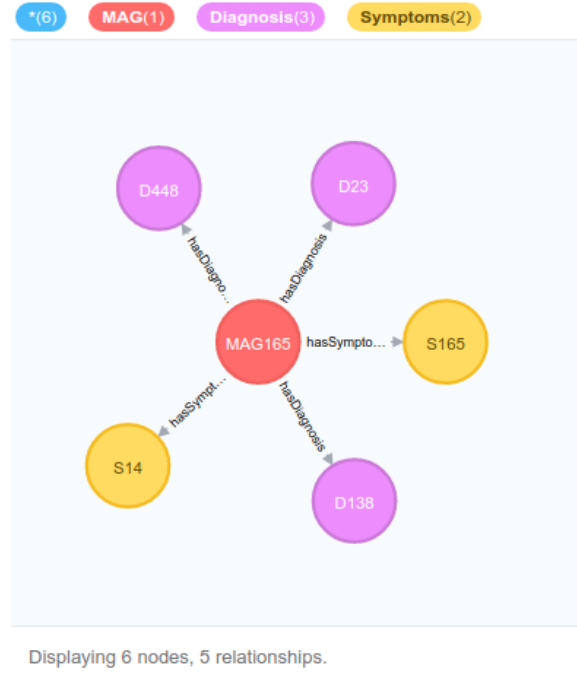


Figure 7.9: Example: MAG Node with Itemsets and Baskets

The Figure 7.8 shows MAG 165 has relationship `has_symptoms` with symptoms node containing symptoms `[s14, s165]` and `has_Diagnosis` relationship with diagnosis node containing diagnosis `[d448, d23, d138]`. Neo4j also provided clear and complete visualization of diagnosis and corresponding symptoms related to MAG. For instance, we were able to represent different types of nodes with different colors. In our model, MAGs were represented by red, symptoms were represented by yellow and diagnosis were represented by purple with its label at the top.

The Figure 7.10 shows the connection of MAG165 with its other MAGs: MAG310 and MAG 260. The connection between MAGs was established by "connectedTo" relationship. These two MAGs shares some common symptoms `[s14, s165]` and diagnosis `[d448, d23, d138]` along with MAG165. Additional symptom `s20` and `s228` are present in MAG310 and MAG260 respectively. These connected MAGs are connected further to other MAGs and provide an alternative to take different decisions by understanding

the current context of patients. Traveling from MAG with less number of symptoms to more number of symptoms helps to narrow down the possible diagnosis.

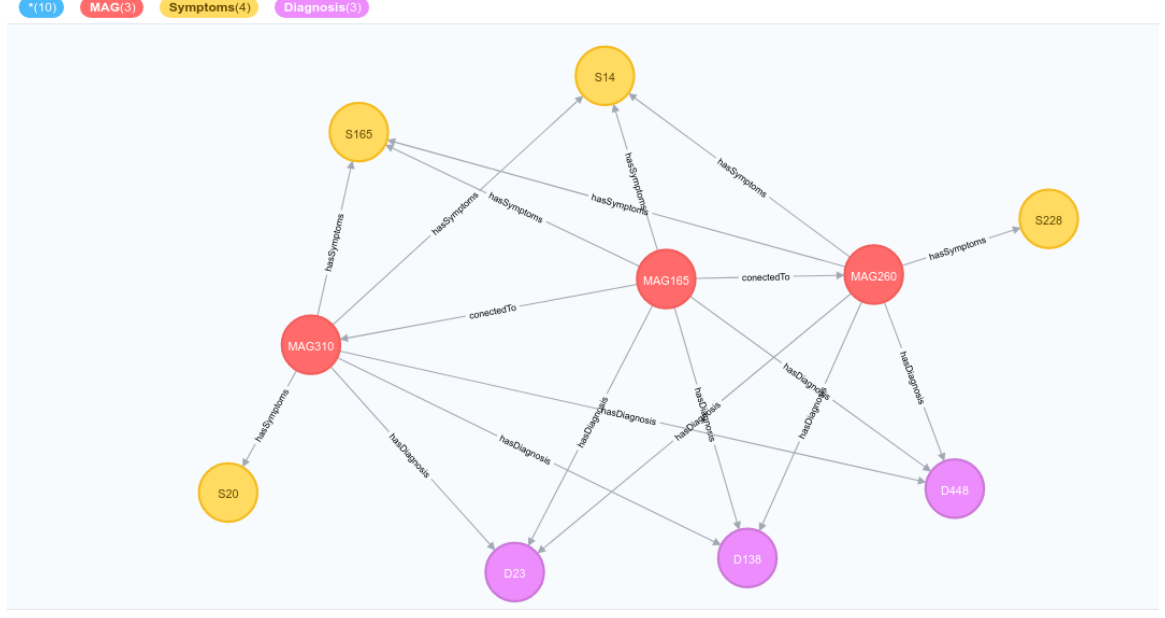


Figure 7.10: Example: MAG165 Connections

For the traversal from one MAG to another MAG as mentioned in our approach, we calculate the weight between two MAG with the formula mentioned above. The connected MAGs were sorted in descending order with their Weight.

$$Weight = \frac{\text{number of baskets (less number of diagnosis) in successor MAG}}{\text{number of baskets (more number of diagnosis) in preceding MAG}}$$

This provides decision making measures for health personnel to consider for finding possible symptoms and diagnosis in different MAGs. This can assist the health personnel during the decision-making process and guides further by providing visualization and all possible diagnosis that could be possible in the current context of the patient.

7.4.4 Integrating Expert Knowledgebase

We were focused to make our knowledgebase more knowledgeable. The more detail expert knowledge related to symptoms and diagnosis could assist further during the decision-making process. The details knowledge of diagnosis and symptoms during examination give a broad view to more specific diagnosis that matches with the context. This helps to understand the context clearly and to choose the next MAG more effectively. The expert knowledge such as what are the tests recommended for the diagnosis, what sort of treatments are possible, and different medication for these diagnoses was integrated into our knowledgebase. Also, symptoms causes, types, causes, sign and general medicine prescribe by the experts was integrated. We gather this knowledge from "www.mayoclinic.org/" and "<https://www.medicinenet.com/>", and considered an expert knowledge database. We integrated the expert knowledge in our Neo4j database with the help of cypher query.

The Figures 7.1 and 7.2 shows the view after the integration of expert knowledge for headache diagnosis and symptom node in the knowledgebase.

```
1 "treatment": "[Pain-relieving medications, Preventive medications]",
2 "medication": "[Pain relievers, Triptans, Ergots, Anti-nausea medications
3 , Opioid medications]",
4 "tests": "[Blood tests, MRI, CT, Spinal tap]",
5 "id": "D153"
" name" : "Headache"
```

Listing 7.1: Diagnosis Node Detail Structure

```
1 "sign": "[Pain that begins in the back of the head and upper neck. The  
    most intense pressure may be felt at the temples or over the  
    eyebrows where the temporalis and frontal muscles are located.]",  
2 "name": "Headache",  
3 "id": "S86"  
4 "description": "A headache or head pain sometimes can be difficult to  
    describe, but some common symptoms include throbbing, squeezing,  
    constant, unrelenting, or intermittent. The location may be in one  
    part of the face or skull, or may be generalized involving the whole  
    head.",  
5 "types": "[primary,Secondary]"
```

Listing 7.2: Symptom Node Detail Structure

Chapter 8

Analysis of Knowledgebase

In this Chapter, we evaluate our knowledgebase by analyzing indegree, outdegree, connections of all the nodes in the knowledgebase and visualize with different graphs.

8.1 Verify Number of Nodes

We performed an analysis of Knowledge graph to evaluate the knowledgebase. The representation of knowledge in the form of graph and the actual knowledge after the data mining was verified by querying a graph database using cypher query. We verified the number of MAG nodes, diagnosis nodes and symptoms nodes in the Knowledge Graph.

The number of MAGs obtained from the data mining in our experimentation equals to the number of MAGs represented in the knowledgebase. The knowledgebase consists of distinct 652 MAGs nodes, 109 Diagnosis nodes and 54 Symptoms nodes as shown in Figure 8.1. Also, we discovered 1307 relationship between the MAGs. This count verifies the number of nodes with respect to the number of patterns and relationships resulted from data mining. The number of nodes and relationships obtained from data mining is a good number considering the size of the dataset that we have used to develop a prototype system.


```
$ MATCH (n:MAG),(s:Symptoms),(d:Diagnosis) return count(distinct n ) as MAG, count(distinct s) as Symptoms, count...
```

	MAG	Symptoms	Diagnosis
	652	54	109

Figure 8.1: Example: MAG, Diagnosis and Symptoms Count

8.2 Evaluate Indegree and Outdegree

The indegree and outdegree MAGs were also analyzed querying the knowledge graph. Indegree of MAGs shows how many MAG nodes are connected to the particular MAG, where as the outdegree shows how many MAG nodes are connected from the particular MAG.

The result for in-degree and outdegree has been plotted as the line graph in Figures 8.3 and Figure 8.2 for top 50 MAG nodes in descending order. The MAG651 has the highest indegree of 7 and MAG36 has highest outdegree 15. The purpose of this evaluation is to visualize the connection between MAGs in the form of indegree and outdegree number.

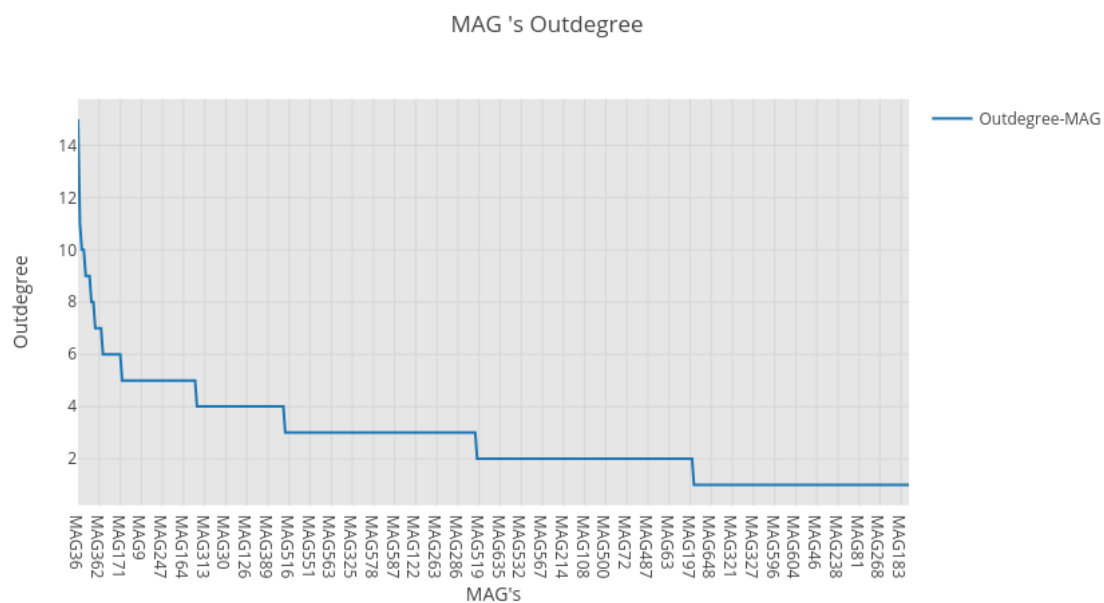


Figure 8.2: MAGs Outdegree

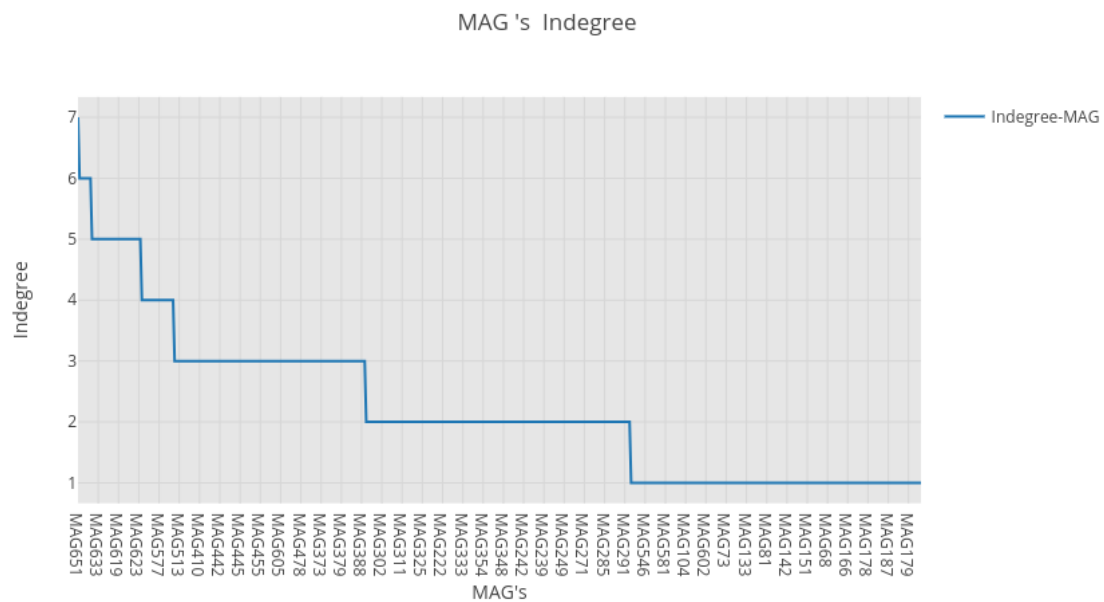


Figure 8.3: MAGs Indegree

8.2.1 Degree Histogram of MAG

We analyzed our knowledge graph by finding the number of MAGs having different degrees. The line graph is shown in Listing 8.4 indicates that the graph consists of the highest number of MAGs with degree 3 with 140 MAGs count.

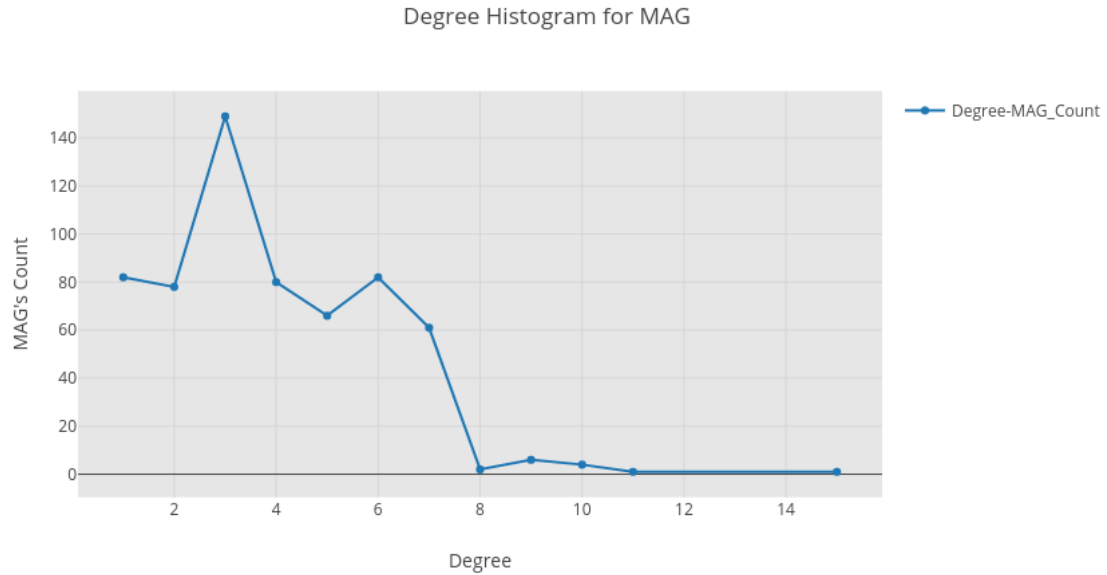


Figure 8.4: Degree Histogram Graph

8.2.2 Check Triangle Formation and Transitive Closure Check

We analyze and found that the knowledge graph was free from any triangular relationship between the MAGs. The cypher query in Listing 8.1 resulted in null record which indicates our graph was free from triangular relationship.

```
1 match (a:MAG) -[:connectedTo]->(b:MAG) -[:connectedTo]->(c:MAG) -[:  
    connectedTo]->(a) return distinct a, b, c
```

Listing 8.1: Checking Triangle Formation

Also, we checked the transitive closure between the MAGs that helps to know the bigger cycle formation. Given a reflexive binary relation R and a , b and c are vertices's in the graph G , construct the minimal (with respect to inclusion) relation R^+ that contains R and has the transitivity property; that is, if aR^+b and bR^+c , then aR^+c . We verify that MAG nodes in our knowledgebase are free from the transitive relationship by querying with the cypher query as shown in Listing 8.2.

```
1 match (a:MAG) -[:connectedTo*]->(b:MAG) -[:connectedTo*]->(c:MAG) -[:
    connectedTo*]->(a) return distinct a, b, c
```

Listing 8.2: Checking Transitive Closure

8.3 Evaluate Number of Symptoms in MAGs

We evaluate the MAGs with their itemset of symptoms by querying the graph database. The query result shown in Figure 8.5 indicates MAGs and the number of symptoms in descending order.

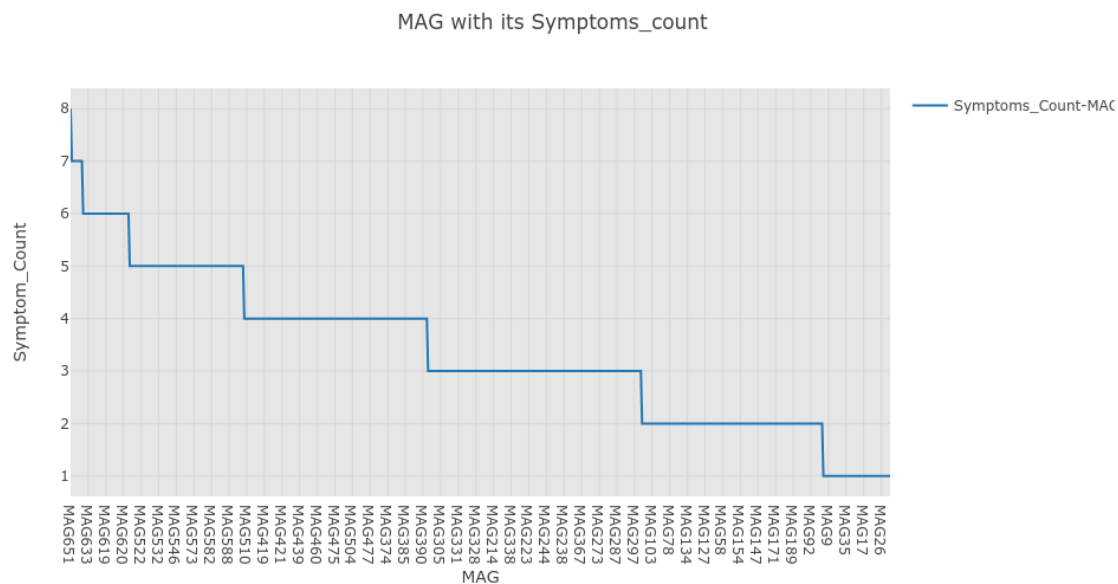


Figure 8.5: MAGs and Symptoms Count

8.4 Evaluate Number of Diagnosis in MAGs

We evaluate the MAGs with their basket of diagnosis by querying the graph database. The query result shown in Figure 8.6 indicates MAGs and number of diagnosis in descending order. From the graph, we can infer MAG36 consists of highest number of diagnosis with 53 counts.

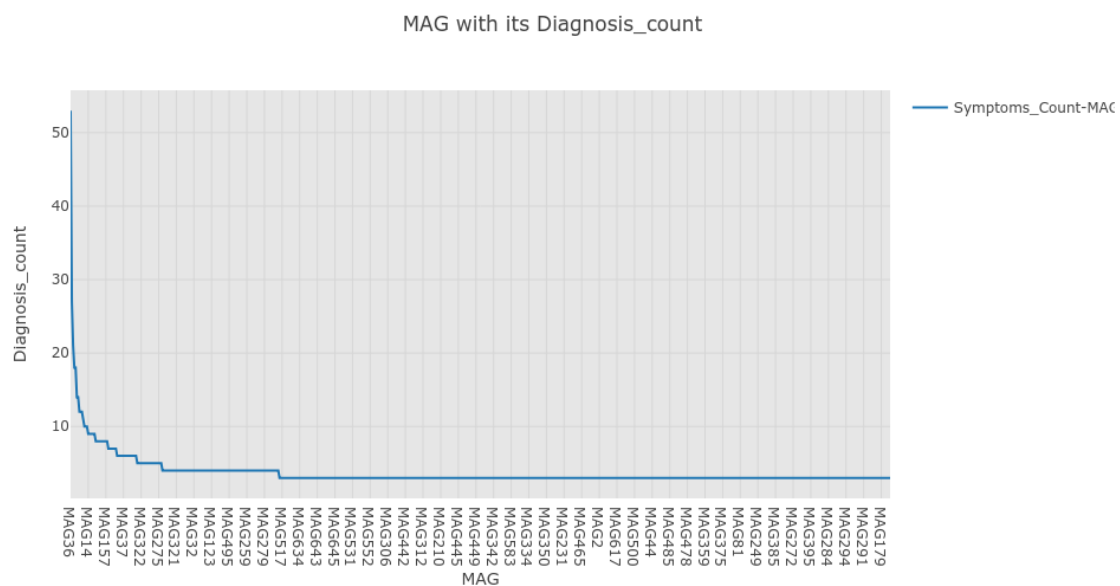


Figure 8.6: MAGs Diagnosis Count

The Neo4j provides the visualization of all the connected MAGs of MAG36 in the graph shown in Figure 8.7.

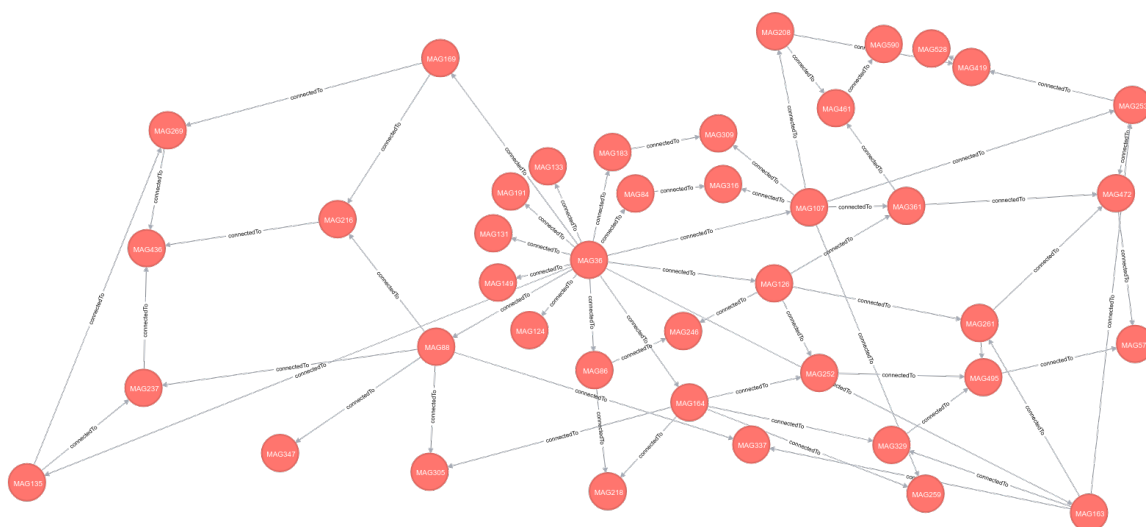


Figure 8.7: MAG36 Connection

8.4.1 All Shortest Path

Finding the shortest path from one node can be performed with shortest paths algorithm provided by Neo4j. We were able to find all the shortest path between source and destination with **allshortestpaths** function. The result shown in Figure 8.8 shows all the shortest path with source node MAG36 and destination node MAG575 having path length more than three. There were 12 paths that could be traverse from source node MAG36 to destination node MAG575.

PATH
["MAG36", "MAG107", "MAG361", "MAG472", "MAG575"]
["MAG36", "MAG107", "MAG253", "MAG472", "MAG575"]
["MAG36", "MAG126", "MAG361", "MAG472", "MAG575"]
["MAG36", "MAG126", "MAG261", "MAG472", "MAG575"]
["MAG36", "MAG126", "MAG261", "MAG495", "MAG575"]
["MAG36", "MAG126", "MAG252", "MAG495", "MAG575"]
["MAG36", "MAG163", "MAG261", "MAG472", "MAG575"]
["MAG36", "MAG163", "MAG261", "MAG495", "MAG575"]
["MAG36", "MAG163", "MAG253", "MAG472", "MAG575"]
["MAG36", "MAG163", "MAG329", "MAG495", "MAG575"]
["MAG36", "MAG164", "MAG329", "MAG495", "MAG575"]
["MAG36", "MAG164", "MAG252", "MAG495", "MAG575"]

Figure 8.8: All shortest path between MAG36 and MAG575

Chapter 9

Case Study

To demonstrate the functionality of the proposed CDSS, we used the eye dataset with disease and symptom relationship from an open source website-”kaggle”. We investigated several scenarios that are possible during the examination of patients in various circumstances to evaluate the results. Different cases are discussed in this section that is encountered during the use of the system.

1. Patient record not present in the historical database

We took a scenario where the user entered the PatientID which is not present in the historical database. We consider a case of patient Jennifer with PatientID: P0004 who is newly admitted patient and doesn’t have any historical record in the database. For this case, the system prompts out with the alert message ”No historical message found”. Now the user needs to enter all the new symptoms of the patient in this case.

After that, the system prompts with the message to enter new symptoms of the patients. The user enters the symptoms of the patient which are parsed and searched in the knowledgebase to match the MAG node consisting these symptoms. To illustrate this scenario, **we took a case of a patient having new symptoms s111 (Visual Problem), s231 (Face numbness), s93 (Speech**

Problem), s103 (Unsteady gait) and s20 (Dizziness) for which our system recommended three appropriate diagnosis d138 (Cerebral vascular accident stroke), d100 (Brain tumor cancer of the brain), d448 (Multiple sclerosisMS). For the symptoms entered as s111 (Visual Problem) which match with MAG36 that consists of 53 diagnoses in it. Our system determined 14 possible connection from MAG36 to other MAGs as shown in the Figure 9.1. All the 14 connected MAG contains corresponding different additional symptoms that are recommended to the user with the possible symptoms and its weights. We chose symptom s231 (Face numbness) correspondence to MAG107 to be more appropriate with 8 patient diagnosis d448 (Multiple sclerosisMS), d100 (Brain tumor cancer of the brain), d1075 (Lacunar stroke), d138 (Cerebral vascular accident stroke), d203 (Diabetes high blood sugar), d43 (Meniscus injury knee cartilage injury). This shows diagnosis number are converging to lesser and appropriate diagnosis. MAG107 is further connected with 5 other MAGs as shown in the Figure 9.2. we chose s93 (Speech Problem) to be more correlated with the patient condition that matches with the MAG253 connection from MAG107. This further recommended 2 connected MAGs (MAG472 and MAG419) as shown in the Figure 9.3. Since S103 (Unsteady gait) was more appropriate to the patient condition corresponding to MAG472, with less number of diagnosis (d448, d100, d1075, d138, d435) as shown in the Figure 9.4. MAG472 is further connected to final node MAG575 with symptoms s20 (Dizziness). This resulted in the final recommendation of diagnosis set (d138, d100, d448). In this way, our system was able to recommend more and most appropriate 3 diagnoses out of 53 diagnoses at the initial match. The Figure 9.5 shows the overall MAG connection scenario from MAG36 to MAG575.

	ConnectedMAG	Diagnosis	MAG	New_DiagnosiCount	Old_DiagnosiCount	Weight	possibleDiagnosis	possibleSymptoms
0	MAG88	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	11	53	0.207547	[D1060, D699, D810, D1089, D488, D100, D1070, ...	[S40, S111]
1	MAG131	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	10	53	0.188679	[D699, D810, D487, D488, D707, D1110, D1068, D...	[S21, S111]
2	MAG163	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	10	53	0.188679	[D448, D1109, D100, D725, D1075, D138, D1070, ...	[S93, S111]
3	MAG133	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	9	53	0.169811	[D487, D747, D484, D435, D182, D13, D183, D300...	[S187, S111]
4	MAG126	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	8	53	0.150943	[D448, D594, D1109, D100, D725, D1075, D138, D...	[S103, S111]
5	MAG164	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	8	53	0.150943	[D448, D810, D1096, D100, D725, D138, D435, D804]	[S20, S111]
6	MAG107	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	6	53	0.113208	[D448, D100, D1075, D138, D203, D435]	[S231, S111]
7	MAG169	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	6	53	0.113208	[D810, D100, D809, D136, D807, D597]	[S266, S111]
8	MAG86	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	6	53	0.113208	[D434, D760, D725, D138, D1070, D804]	[S27, S111]
9	MAG135	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	5	53	0.094340	[D810, D138, D203, D809, D807]	[S264, S111]
10	MAG191	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	3	53	0.056604	[D176, D1092, D487]	[S182, S111]
11	MAG149	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	3	53	0.056604	[D435, D100, D760]	[S110, S111]
12	MAG84	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	3	53	0.056604	[D138, D203, D448]	[S165, S111]
13	MAG124	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	3	53	0.056604	[D138, D448, D807]	[S48, S111]
14	MAG183	[D699, D1089, D1096, D100, D1134, D13, D922, D...	MAG36	3	53	0.056604	[D138, D203, D448]	[S228, S111]

Figure 9.1: Output 1

	ConnectedMAG	Diagnosis	MAG	New_DiagnosiCount	Old_DiagnosiCount	Weight	possibleDiagnosis	possibleSymptoms
0	MAG253	[D448, D100, D1075, D138, D203, D435]	MAG107	5	6	0.833333	[D448, D100, D1075, D138, D435]	[S93, S231, S111]
1	MAG259	[D448, D100, D1075, D138, D203, D435]	MAG107	4	6	0.666667	[D138, D435, D100, D448]	[S231, S20, S111]
2	MAG361	[D448, D100, D1075, D138, D203, D435]	MAG107	4	6	0.666667	[D138, D100, D448, D1075]	[S231, S103, S111]
3	MAG316	[D448, D100, D1075, D138, D203, D435]	MAG107	3	6	0.500000	[D138, D203, D448]	[S165, S231, S111]
4	MAG208	[D448, D100, D1075, D138, D203, D435]	MAG107	3	6	0.500000	[D138, D100, D448]	[S231, S86, S111]
5	MAG309	[D448, D100, D1075, D138, D203, D435]	MAG107	3	6	0.500000	[D138, D203, D448]	[S228, S231, S111]

Figure 9.2: Output 2

	ConnectedMAG	Diagnosis	MAG	New_DiagnosiCount	Old_DiagnosiCount	Weight	possibleDiagnosis	possibleSymptoms
0	MAG472	[D448, D100, D1075, D138, D435]	MAG253	4	5	0.8	[D138, D100, D448, D1075]	[S93, S231, S103, S111]
1	MAG419	[D448, D100, D1075, D138, D435]	MAG253	3	5	0.6	[D138, D100, D448]	[S93, S231, S86, S111]

Figure 9.3: Output 3

ConnectedMAG	Diagnosis	MAG	New_DiagnosiCount	Old_DiagnosiCount	Weight	possibleDiagnosis	possibleSymptoms
0	MAG575 [D138, D100, D448, D1075]	MAG472	3	4	0.75	[D138, D100, D448]	[S93, S20, S103, S231, S111]

Figure 9.4: Output 4

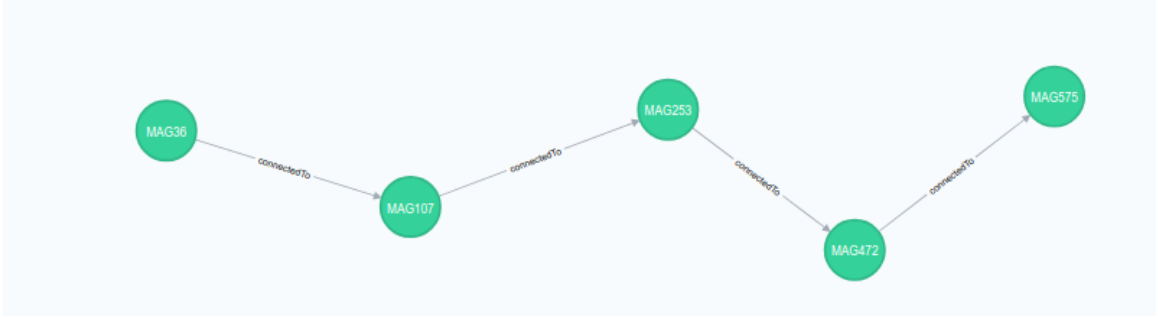


Figure 9.5: Overall Connection of MAG36

2. Patient historical record present in data

The second case takes into account when the patient symptom is present in the historical database. We consider a patient record of John with PatientID: P0032 having a prior record in the historical database with symptoms s25 {Eye pain (Irritation)} and s111 (Visual problems). Our system matches the corresponding MAG122 with the patient symptoms in the knowledgebase and determines 14 different diagnoses in it. The doctor asks John for recent other symptoms other than historical symptoms. John stated he has the s21 {Double vision (Diplopia)} symptoms too. The doctor enters the new symptoms to the CDSS. The system searches for the appropriate MAG with current context with symptoms s25, s111 and s21 and recommends the other connected MAGs. The symptom matches to MAG364 with the diagnosis {d699 (Temporal arthritis inflamed scalp artery), d1139 (Retinopathy), d487 (Orbital cellulitis soft tissue eye infection)}. As there was no further connection from MAG364, our system

notifies doctor that MAG364 is final MAG.

We were able to conclude that our approach is accurate and very effective during decision-making process with above case studies.

Chapter 10

Future Enhancement and Conclusion

Although we were able to predict the accurate result from the data, there is still open space for future enhancement and research. The decision process requires user involvement to choose most appropriate choice among the recommended options. The decision is affected by the knowledge of the user. The weight is one of the factors that could be considered for making the decision. However, this process could be improved if we are able to capture the user choice during the decision-making process and use it while another user encounters similar context. Also, our approach could be used as a basis for other datasets in different areas such as marketing, medical and security to obtain the patterns and relationships for creating the knowledgebase and utilizing in the decision-making process.

Our approach is the data-driven approach so the result of the system depends upon the data fed to the data mining algorithm. We were able to manually evaluate the result in our small dataset. However, the evaluation process might be complex and time-consuming in the large dataset. Although we haven't referred any medical expertise to evaluate the system accuracy, our system is able to recommend the result that complies with the data provided during knowledgebase creation. Our system might recommend less appropriate result if the user decided to choose less relevant symptoms with the patient condition during the decision-making process. Also, if the

data cleaning process is not performed well then that might result in the possibility of recommending less appropriate diagnosis or even loss of different patterns from the data.

We developed the prototype Clinical Decision Support System that helped in making proper decision to recommend more appropriate diagnosis for effective decision. The knowledgebase was created using Neo4j database consisted of MAGs with item-sets - a group of symptoms, baskets - a group of diagnosis and expert knowledge. The knowledgebase was referred by CDSS to recommend the next possible symptoms of the patients with their context. Our approach facilitated with prior knowledge of next possible diagnosis and symptoms so that health practitioner get the possible results in advance during the decision-making process. Ultimately, it resulted in a more appropriate and closer diagnosis by determining possible symptoms of the patient. Our novel approach considered data mining technique to represent the mined knowledge in the form of graph nodes and integrated the expert knowledge with more information about the diagnosis and symptoms in the knowledgebase. We also introduced visualization and exploration of knowledge in the knowledge graph that helped in easy access to more detail information.

BIBLIOGRAPHY

- [1] A. Yarazavi and K. Sartipi, “A semantic-driven and interactive approach to mobile decision support services,” *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 2014.
- [2] K. Kaur and R. Rani, “Managing data in healthcare information systems: Many models, one solution,” *Computer*, vol. 48, no. 3, pp. 52–59, Mar 2015.
- [3] [www.grandviewresearch.com](https://www.grandviewresearch.com/industry-analysis/electronic-health-records-ehr-market), “Electronic health records (ehr) market analysis by product (client server-based, web-based),” <https://www.grandviewresearch.com/industry-analysis/electronic-health-records-ehr-market>, 2017, [Online; accessed 28-DEC-2017].
- [4] [www.healthit.gov](https://www.healthit.gov/providers-professionals/faqs/what-information-does-electronic-health-record-ehr-contain), “What information does an electronic health record (EHR) contain?” <https://www.healthit.gov/providers-professionals/faqs/what-information-does-electronic-health-record-ehr-contain>, 2017, [Online; accessed 28-DEC-2017].
- [5] D. Kaur and A. Paul, “Performance analysis of different data mining techniques over heart disease dataset,” *International Journal of Current Engineering and Technology*, vol. 4, no. 1, pp. 220–224, 2014.
- [6] S. H. El-Sappagh and S. El-Masri, “A distributed clinical decision support system architecture,” *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 1, pp. 69–78, 2014.
- [7] E. G. Giannopoulou, “Application of data mining and text mining to the analysis of medical near miss cases,” in *Data Mining in Medical and Biological Research*. InTech, 2008.
- [8] [https://db-engines.com](https://db-engines.com/en/ranking/graph+dbms), “Db-engines ranking of graph dbms,” <https://db-engines.com/en/ranking/graph+dbms>, 2018, [Online; accessed 18-MAY-2018].
- [9] [py2neo.org](http://py2neo.org/v4/), “The py2neo v4 handbook,” <http://py2neo.org/v4/>, 2018, [Online; accessed 18-MAY-2018].

- [10] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor, “Cypher: An evolving query language for property graphs,” in *ACM SIGMOD International Conference on Management of Data (SIGMOD 2018)*, 2018.
- [11] Y. Yao, “Concept lattices in rough set theory,” in *Fuzzy Information, 2004. Processing NAFIPS’04. IEEE Annual Meeting of the*, vol. 2. IEEE, 2004, pp. 796–801.
- [12] J. Han, H. Cheng, D. Xin, and X. Yan, “Frequent pattern mining: current status and future directions,” *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.
- [13] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” in *Acm sigmod record*, vol. 22. ACM, 1993, pp. 207–216.
- [14] D. Power, “Decision support systems: Concepts and resources for managers,” 01 2002.
- [15] M. M. Hamad and B. A. Qader, “Knowledge-driven decision support system based on knowledge warehouse and data mining for market management,” *Global Journal of Management And Business Research*, vol. 13, no. 10, 2014.
- [16] neo4j.com, “What is a graph database?” <https://neo4j.com/developer/graph-database/>, 2018, [Online; accessed 18-MAY-2018].
- [17] N. Guarino, D. Oberle, and S. Staab, “What is an ontology?” in *Handbook on ontologies*. Springer, 2009, pp. 1–17.
- [18] I. Horrocks, P. F. Patel-Schneider, and F. Van Harmelen, “From shiq and rdf to owl: The making of a web ontology language,” *Web semantics: science, services and agents on the World Wide Web*, vol. 1, no. 1, pp. 7–26, 2003.
- [19] K. Breitman, M. A. Casanova, and W. Truszkowski, *Semantic web: concepts, technologies and applications*. Springer Science & Business Media, 2007.
- [20] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz, “Ontologies for enterprise knowledge management,” *IEEE Intelligent Systems*, vol. 18, no. 2, pp. 26–33, 2003.
- [21] J. N. Liu, Y.-L. He, E. H. Lim, and X.-Z. Wang, “A new method for knowledge and information management domain ontology graph model,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 1, pp. 115–127, 2013.

- [22] Wikipedia, “Formal concept analysis,” https://en.wikipedia.org/wiki/Formal_concept_analysis, 2018, [Online; accessed 04-04-2018].
- [23] M. Siff and T. Reps, “Identifying modules via concept analysis,” *IEEE Transactions on Software Engineering*, vol. 25, no. 6, pp. 749–768, 1999.
- [24] A. Van Deursen and T. Kuipers, “Identifying objects using cluster and concept analysis,” in *Proceedings of the 21st international conference on Software engineering*. ACM, 1999, pp. 246–255.
- [25] D. A. S. Tăut, C. Săcărea, and A. V. S. Tăut, “Knowledge visualization for supporting communication in cardiovascular risk assessment hypotheses,” in *Software, Telecommunications and Computer Networks (SoftCOM), 2015 23rd International Conference on*. IEEE, 2015, pp. 249–253.
- [26] C. Săcărea, “Investigating oncological databases using conceptual landscapes,” in *International Conference on Conceptual Structures*. Springer, 2014, pp. 299–304.
- [27] H. M. Zolbanin, D. Delen, and A. H. Zadeh, “Predicting overall survivability in comorbidity of cancers: A data mining approach,” *Decision Support Systems*, vol. 74, pp. 150–161, 2015.
- [28] D. Delen, G. Walker, and A. Kadam, “Predicting breast cancer survivability: A comparison of three data mining methods,” *Artif. Intell. Med.*, vol. 34, no. 2, pp. 113–127, Jun. 2005. [Online]. Available: <http://dx.doi.org/10.1016/j.artmed.2004.07.002>
- [29] A. Yousefi, N. Mastouri, and K. Sartipi, “Scenario-oriented information extraction from electronic health records,” in *Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium on*. IEEE, 2009, pp. 1–5.
- [30] K. Sartipi, N. P. Archer, and M. H. Yarmand, “Challenges in developing effective clinical decision support systems,” in *Efficient Decision Support Systems-Practice and Challenges in Biomedical Related Domain*. InTech, 2011.
- [31] D. Rosca, S. Greenspan, M. Feblowitz, and C. Wild, “A decision making methodology in support of the business rules lifecycle,” in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, Jan 1997, pp. 236–246.
- [32] Q. Wen and J. He, “Personalized recommendation services based on service-oriented architecture,” in *Services Computing, 2006. APSCC’06. IEEE Asia-Pacific Conference on*. IEEE, 2006, pp. 356–361.

- [33] S. Tsumoto, "Automated extraction of medical expert system rules from clinical databases based on rough set theory," *Information sciences*, vol. 112, no. 1-4, pp. 67–84, 1998.
- [34] S.-H. Cho, J. Jeon, and S. I. Kim, "Personalized medicine in breast cancer: a systematic review," *Journal of breast cancer*, vol. 15, no. 3, pp. 265–272, 2012.
- [35] S. Molinaro, S. Pieroni, F. Mariani, and M. N. Liebman, "Personalized medicine: Moving from correlation to causality in breast cancer," *New Horizons in Translational Medicine*, vol. 2, no. 2, p. 59, 2015.
- [36] R. A. Miller, L. R. Waitman, S. Chen, and S. T. Rosenbloom, "The anatomy of decision support during inpatient care provider order entry (cpoe): empirical observations from a decade of cpoe experience at vanderbilt," *Journal of biomedical informatics*, vol. 38, no. 6, pp. 469–485, 2005.
- [37] M. K. Goldstein, R. W. Coleman, S. W. Tu, R. D. Shankar, M. J. O'Connor, M. A. Musen, S. B. Martins, P. W. Lavori, M. G. Shlipak, E. Oddone *et al.*, "Translating research into practice: organizational issues in implementing automated decision support for hypertension in three medical centers," *Journal of the American Medical Informatics Association*, vol. 11, no. 5, pp. 368–376, 2004.
- [38] M. W. Jaspers, M. Smeulders, H. Vermeulen, and W. Linda, "Peute. 2011." effects of clinical decision-support systems on practitioner performance and patient outcomes: A synthesis of high-quality systematic review findings., " *Journal of the American Medical Informatics Association: JAMIA*, vol. 18, no. 3, pp. 327–34.
- [39] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach, "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. british medical association 2005; 10 (3)," *Clinical research ed PubMed Abstract— Publisher Full Text— PubMed Central Full Text OpenURL*, 2010.
- [40] L. Moja, K. H. Kwag, T. Lytras, L. Bertizzolo, L. Brandt, V. Pecoraro, G. Rigon, A. Vaona, F. Ruggiero, M. Mangia *et al.*, "Effectiveness of computerized decision support systems linked to electronic health records: a systematic review and meta-analysis," *American journal of public health*, vol. 104, no. 12, pp. e12–e22, 2014.
- [41] S. Ram, W. Zhang, M. Williams, and Y. Pengetnze, "Predicting asthma-related emergency department visits using big data," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 4, pp. 1216–1223, July 2015.
- [42] J.-H. Eom, S.-C. Kim, and B.-T. Zhang, "Aptacdss-e: A classifier ensemble-based clinical decision support system for cardiovascular disease level prediction," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2465–2479, 2008.

- [43] E. Çomak, A. Arslan, and İ. Türkoğlu, “A decision support system based on support vector machines for diagnosis of the heart valve diseases,” *Computers in Biology and Medicine*, vol. 37, no. 1, pp. 21–27, 2007.
- [44] K. Zarkogianni and K. S. Nikita, “Personal health systems for diabetes management early diagnosis and prevention,” *Handbook of Research on Trends in the Diagnosis and Treatment of Chronic Conditions*, p. 465, 2015.
- [45] N. Ramakrishnan, D. Hanauer, and B. Keller, “Mining electronic health records,” *Computer*, vol. 43, no. 10, pp. 77–81, 2010.
- [46] A. Yarazavi and K. Sartipi, “Consultant-as-a-service: an interactive and context-driven approach to mobile decision support services,” in *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research*. IBM Corp., 2013, pp. 274–282.
- [47] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. ” O’Reilly Media, Inc.”, 2012.

