Novel Approaches for Relation Extraction in Biomedical Domain

Stanley Chika Onye

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
November 2018
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy in Applied Mathematics and Computer Science.

Prof. Dr. Nazim Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Applied Mathematics and Computer Science.

Asst. Prof. Dr. Nazife Dimililer
Co-Supervisor

Asst. Prof. Dr. Arif Akkeleş
Supervisor

Examining Committee

1. Prof. Dr. Rashad Aliyev
2. Prof. Dr. Benedek Norbert Nagy
3. Prof. Dr. Hayri Sever
4. Prof. Dr. Mehmet Reşit Tolun
5. Asst. Prof. Dr. Arif Akkeleş

ABSTRACT

Relation extraction an important field in Biomedical Natural Language Processing is the study of identifying relations between entity mentions. The extraction of relation instances over multiple sentence mention levels (intra- and inter-sentence levels) has been a challenge. In the intra-sentence level, the mention of a pair of entity is found in a single sentence, whereas in the inter-sentence level, they are found in spanning neighbouring sentences. The variations in the level of extractable information and performance from these levels have been a reason for this challenge.

In this thesis, we tackled this challenge by carefully examining the stages of text processing and relation instance construction of the candidate relation instances across the multiple sentence levels and further performed a combination of the relation instances over these mention levels in order improve the performance of the system. In the text processing stage, we performed sentence simplification after the sentences have been segmented in order to improve the information extracted through a dependency parse tree. During the extraction of the candidate relation instances, we applied some sentence structures and rules to help improve the level of the types of candidates selected.

We performed relation extraction using two systems. We developed a system that employs an optimization technique namely genetic algorithm, to combine the output of the classifiers trained using the candidate relation instances from both levels. We introduce the novel approach of using two decision-making under uncertainty techniques for our classifier selection. The other system is based on an ensemble of

two machine learning algorithms. We performed relation extraction by employing the candidate relation instances from the two levels in two forms. Firstly, the instances are merged after they have been classified individually, and secondly, the instances are merged before the classification. The system then introduces the novel use of a maximum probability-based voting algorithm to combine the results generated from these two forms. All the experiments in this study are performed using the BioCreative V chemical disease relation dataset which is the most comprehensive dataset in the domain.

Keywords: Classifier Ensemble, Decision-Making Techniques, Genetic Algorithms, Optimization Techniques, Relation Extraction, Text Mining.

ÖZ

Text içerisinde geçen varlıklar arasındaki ilişkileri bulmayı hedefleyen ilişki çıkarımı biyomedikal doğal dil işleme konusundaki önemli alanlardan biridir. İki varlık arasındaki ilişki tek bir cümle içerisinde tanımlanabileceği gibi, birbiriyle komşu iki veya daha fazla cümle ile de tanımlanabilir. Tek bir cümle içerisinde tanımlanan ilişkiler için "cümle-içi", tanımı komşu iki veya daha fazla cümle ile yapılan ilişkilere "cümleler-arası" ilişki terimleri kullanılmıştır. Cümle-içi ve cümleler-arası seviyelerde ilişkilerin çıkarımını yapmak, her iki seviyede elde edilen bilgilerin içerik ve miktar olarak farklı olması nedeniyle zorluk çıkarmaktadır.

Çalışmamızde, her iki seviyedeki aday ilişki örneklerinin oluşturulması için metin işleme ve ilişki örneği oluşturma aşamalarını dikkatle inceleyerek ve akabinde performansın daha da iyileştirilmesi için her iki seviyede tahmin edilmiş olan ilişki örneklerini sınıflayıcı kombinasyonları kullanılarak birleştirmek suretiyle bu zorluk aşılmıştır. Metin işleme aşamasında, metin cümlelere bölündükten sonra cümle basitleştirilmesi uygulanarak bağımlılık ayrıştırma ağacından çıkarılacak bilgilerin iyileştirilmesi sağlanmıştır. Aday ilişki örneklerinin çıkarılması sırasında, anlamlı ve doğru ilişki adayları seçebilmek için bazı kurallar ve cümle yapıları uygulanmıştır.

Tez kapsamında ilişki çıkarımı için iki ayrı sistem geliştirilmiştir. Geliştirilen ilk sistemde, her iki cümle seviyesindeki aday ilişki örnekleri ile eğitilen sınıflandırıcıların çıktıları eniyileme yöntemi ile birleştirmektedir. Eniyileştirme tekniği olarak genetik algoritma ve yenilik olarak sınıflandırıcı seçimi için belirsizlik teknikleri altında iki karar verme yaklaşımı kullanıldı. Geliştirilen diğer makine

öğrenimi sistemimizde, ilişki adayları cümle-içi ve cümleler arası seviyede ayrı ayrı derlenmiş ve bu iki veri kümesi birleştirilerek tüm ilişki adaylarını içeren üçüncü bir veri kümesi oluşturulmuştur. Bu şekilde oluşturulan üç veri seti ayrı ayrı iki makine öğrenimi algoritması kombinasyonunun eğitilmesi için kullanılmıştır. Bu aşamadan sonra tüm ilişki adayları kullanılarak eğitilen sistemin çıktısı ile cümle-içi ve cümleler-arası seviyelerinde eğitilen sınıflandırıcılarının çıktılarının birleşimi maximum probability voting algoritması kullanılarak birleştirilmiştir. İkinci sistemde sunulan yenilik farklı seviyelerin bu şekilde sınıflandırıcı kombinasyonları kullanılarak birleştirilmesidir. Bu çalışmadaki tüm deneyler, alandaki en kapsamlı veri kümesi olan BioCreative V kimyasal hastalık ilişkisi veri kümesi kullanılarak gerçekleştirilmiştir.

Anahtar Kelimeler: Eniyileştirme Teknikleri, Genetik Algoritmalar, İlişkisel Çıkarım, Karar Verme Teknikleri, Metin Madenciliği, Sınıflandırıcı Topluluğu

ASK THE LORD TO BLESS YOUR PLANS, AND YOU WILL BE SUCCESSFUL IN CARRYING THEM OUT.

PROVERBS 16:3

TO MY DAUGHTER, WIFE, PARENTS, AND SIBLINGS.

ACKNOWLEDGEMENT

I would start by thanking my friend, and supervisor, Asst. Prof. Dr. Arif Akkeleş for his immeasurable support and guidance throughout the course of my doctoral program. He did not just call me his student, but family. I am entirely grateful for his supervision, effort and continuous encouragement. To my co-supervisor, Asst. Prof. Dr. Nazife Dimililer, I am blessed to have met you. You did not only guide me with your wisdom through the paths required for a successful completion of my research, but you saw a man whom due to the countless setbacks he encountered during the course of his research did not just require the academic support but also the emotional support. In your affection, you took me like a son and I can proudly say that your motherly love has made me a better student and man.

I owe a lot to my family, and my parents, Chief Sir Samson and Pastor Chioma Onye deserves special praises for standing by me during the most difficult times of my studies and for their prayers, moral and financial supports and encouragements. To my siblings Chinagorom and Samson Jr Onye, I am forever indebted to you for how you supported me with your kind words of encouragement and affection and for how you took a lot of burdens from me to help me concentrate on my studies. To my wife, Mercy Onye, I am deeply thankful for your love, support and understanding. You stood by me and encouraged me through every phase of my research. To our newly born daughter, Kamsiyochukwu, you are the best present I could have dreamt of for the successful completion of my program.

I would like to express my gratitude to the dissertation defence committee, Prof. Dr. Rashad Aliyev, Prof. Dr. Benedek Norbert Nagy, Prof. Dr. Hayri Sever and Prof. Dr. Mehmet Reşit Tolun and to Assoc. Prof. Dr. Ahmet Rizaner for their valuable analysis, comments and suggestions on this work. I would like to thank my friends, colleagues and the director, Prof. Dr. Mustafa Ilkan and staff of SCT for their support through this process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BioCreative	Critical Assessment of Information Extraction systems in Biology
BN	Bayes Network
CDR	Chemical Disease Relation
CID	Chemical-induced Disease
CSS	Classifier Subset Selection
CTD	Comparative Toxicogenomics Database
CNN	Convolutional Neural Network
DCS	Dynamic Classifier Selection
FullM	Complete Base Classifiers and Minimax Regret
FullH	Complete Base Classifiers and Hurwicz Criterion
GA	Genetic Algorithm
HC	Hurwicz criterion
IE	Information Extraction
KB	Knowledge-based
LSTM	Long-Short Term Memory Units
ML	Machine Learning
ME	Maximum Entropy
MR	Minimax Regret
MCS	Multiple Classifier System
NB	Naïve Bayes
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Part-of-Speech

PPI	Protein-Protein Interaction
RH	Roulette Selection and Hurwicz Criterion
RHM	Roulette Selection, Hurwicz Criterion and Minimax Regret
RM	Roulette Selection and Minimax Regret
RTH	Roulette Selection, Tournament selection and Hurwicz Criterion
RTHM	Roulette Selection, Tournament Selection, Hurwicz Criterion and Minimax Regret
RTM	Roulette Selection, Tournament Selection and Minimax Regret
SCS	Static Classifier Selection
SVM	Support Vector Machine
TH	Tournament Selection and Hurwicz Criterion
THM	Tournament Selection, Hurwicz Criterion and Minimax Regret
TM	Tournament Selection and Minimax Regret

Chapter 1

INTRODUCTION

1.1 Motivation

The increase in the amount of predominantly unstructured or weakly structured text motivated the research in text mining. Text mining which is an important aspect of Natural Language Processing (NLP) is aimed at creating and implementing systems that can process, comprehend and discover new, formerly unknown information from them. During the past recent decades, a number of applications of text mining such as Information Extraction (IE), Information Retrieval, and Relation Extraction systems have been developed. The implementation of these systems has led to an increase in the scientific efforts dedicated to improving knowledge discovery in texts such as biomedical literature. IE is a process of generating structural data from unstructured text in order to extract desired information. IE plays a vital role in data management in computational linguistics. Biomedical literature is important for research in biology and medical fields. The studies on the relationship between biomedical entities such as protein-protein interactions [1], drug-drug interactions [2, 3], and chemical-disease relations [4] crucially depend on biomedical data for their existence [5].

Relation extraction systems in the biomedical domain have employed various approaches such as machine learning (ML), pattern recognition, and knowledge-based (KB) approaches. However, the ML systems have been the most frequently employed approach. Relation extraction is treated as a classification task in the ML-based

systems [6, 7, 8, 9, 10]. The ML-based systems are data-driven and are capable of deriving models for automated extraction from annotated data [11, 12, 13, 14, 15]. The input vectors for an ML classifier can be in the form of extracted features or structure representation such as graphs and trees or both. In order to implement both types of input vectors, discriminative classifiers such as support vector machines, decision trees, maximum entropy classifiers may be employed. The aim of biomedical relation extraction is to identify or extract relationships between entities such as drugs, chemicals, side effects and diseases. These relations may be described in a single sentence or two or more neighbouring sentences that mention the two entities. These two cases are referred to as intra sentence or co-occurrence level and inter-sentence or non-co-occurrence level. The amount and type of information that can be extracted from these two different levels of representations are clearly not always compatible.

ML methods meet challenges in coping with the variations of information across the mention levels. In order to address this problem, this thesis aims to find a suitable way extracting relation instances across these mention levels and developing a suitable way of combining them. We show that the use of the different mention levels in different arrangement and combination is capable of creating an effective biomedical relation extraction system.

1.2 Thesis Contributions

In this thesis, we study methods for relation extraction from biomedical literature. We develop two ML-based methods, namely Multiple Classifier System (MCS) that uses an optimization technique based on two decision-making techniques to generate the best possible classifier ensemble for the relation extraction task and a system that extracts relation instances from multiple sentence levels and finally combines them

through the use of a voting algorithm. We focus on the chemical disease relation extraction task where full annotated training data are available due to the BioCreative V Challenge. The major contributions of this thesis are as follows:

i. Chemical-induced disease (CID) relation extraction have mainly been performed using ML-based, knowledge-based, and Rule-based systems, however, in our first method, we introduce the novel approach of using an MCS which utilizes the Genetic Algorithm (GA) as the optimization technique.

ii. The implementation of GA as an optimization technique for a multidimensional classifier selection is made dynamic through the introduction of two decision-making under uncertainty techniques as voting algorithms and a voting bit attached to the chromosomes which determines what voting algorithm is applied on individual chromosomes. Additional variations are introduced in the system during evolution through the use of two selection techniques and two types of crossover.

iii. In the second method, we also introduce another novel approach where we deal with relation mentions on multiple sentence levels by using a classifier combination of two ML classifiers and then combine the results from these levels through the use of a voting algorithm in order to improve the performance of our system.

1.3 Thesis Outline

The remaining of this dissertation consists of the following chapters:

Chapter 2 provides background knowledge for relation extraction methods in the biomedical literature. Firstly, we introduce the concept of a relation extraction system, then we discuss the major components of a general relation system and the main approaches employed in Section 2.1. In Section 2.2, we discuss ML approach which

is the most frequently used approach relation extraction and then in Section 2.3, we discuss multiple classifier systems, its components, implementation and also the GA optimization technique which is implemented in this thesis. Finally, Section 2.4 provides a discussion on the available corpora in the biomedical domain.

The experimental settings employed in this thesis to develop two relation extraction methods are discussed in Chapter 3. It presents the dataset employed in Section 3.1, the base classifiers used in Section 3.2, a description of sentence segmentation and simplification processes performed during text processing in Section 3.3, the construction of the candidate relation instances in Section 3.4, the types of features used in Section 3.5. Finally, in Section 0, we describe the evaluation methods employed.

Chapter 4 presents a CID relation extraction system which is a multi-classifier system using GA. The GA-based system involves the novel implementation of two decision-making techniques for classifier selection. In this section, we first review existing methods and then present our method and discuss its results by comparing it to the other state-of-the-art systems. Finally, we discuss some possible improvements for our approach.

Chapter 5 provides another novel approach for chemical disease relation extraction from biomedical text. After describing the related methods used for chemical disease relation (CDR) extraction tasks, we present our method which involves the extraction of relation instances from multiple sentence mention levels. We describe the use of the combination of two ML classifiers for the CID relation extraction task. We then discuss how we combined the multiple sentence levels by using a voting algorithm to

further improve the performance of the system. Finally, we discuss the results and compare with that of the other state-of-the-art systems.

Chapter 6 summarizes the thesis with an overall discussion on the contributions and future work.

Chapter 2

BACKGROUND AND RELATED WORK

This chapter provides background knowledge on relation extraction and also a review of the existing techniques for relation extraction tasks in the biomedical domain. Additionally, it introduces the concepts required for understanding the works presented in this thesis. In Section 2.1 we describe the relation extraction task, followed by a description of the major components and approaches of a relation extraction system in Section 2.1.1. Section 2.2 discusses the ML approach used for the biomedical relation extraction tasks. In Section 2.3, we discuss the approach of using the multiple classifier systems, with an introduction to this approach given in Section 2.3.1, a discussion on the types of selection algorithms utilized in this approach in Section 2.3.2 and the optimization technique employed in one of the methods reported in this thesis. Finally, in Section 2.4, we describe the available corpora in the biomedical domain that are useful for the relation extraction tasks.

2.1 Relation Extraction

Relation extraction is a task aimed at identifying a relation between entity mentions in a literature with high efficiency and accuracy. Culotta et al. defined relation extraction as “the task of discovering semantic connections between entities” [16]. In biomedical context, relation extraction is a task aimed at extracting relations between biomedical entities mentioned in a life science literature. This task can be further divided into two subtasks: to check if there exists a relation between then entity mentions [7, 8, 12], and to find out the type of relationship that exists [17, 18].

2.1.1 Components of a Relation Extraction System

Due to the exponential growth in the quantity of biomedical text, it becomes increasingly more challenging for biocurators and researchers to be up-to-date with the related development in this field [19]. Over the years, biomedical research has gradually shifted focus from named entity mentions such as chemicals, diseases, or proteins to the entire biological system, thereby creating an urgent increase in the demand for the development of systems capable of extracting existing relationships between the biological entity mentions (e.g. chemical disease relations, protein-protein interactions) [20, 21] to enhance knowledge discovery and possibly to develop scientific hypotheses. This has resulted in the development of automated relation extraction systems capable of handling significantly more articles in order to help solve the time-consuming and strenuous demands associated with the manual transformation of unstructured text into a structured format [22]. Based on various approaches, different relation extraction systems have been proposed. Over the years, these approaches have evolved from the simple co-occurrence approaches to some very robust approaches such as ML-based, pattern-based, and knowledge-based [8, 12, 23, 24, 25]. A relation extraction system is mainly designed to consist of three main modules. These are the text pre-processing, parsing and relation extraction modules [17] as shown in Figure 2.1.

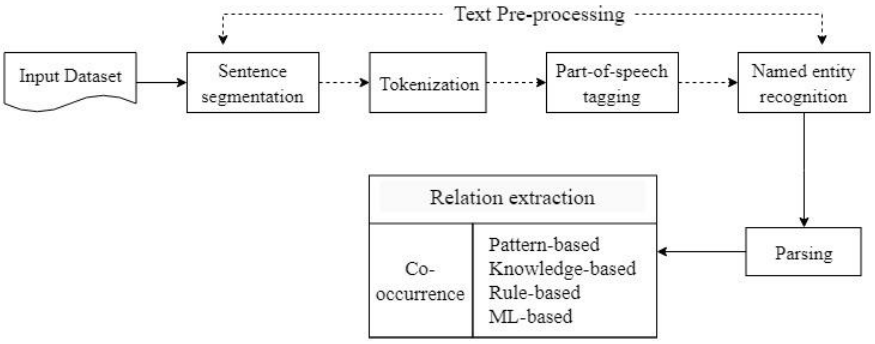

Figure 2.1. A general relation extraction pipeline

2.1.1.1 Text Pre-processing

The text pre-processing module comprises many other sub-processes such as sentence segmentation, tokenization, part-of-speech tagging and named entity recognition.

i. **Sentence segmentation:** Most current relation extraction systems perform relation extraction on a sentence level, therefore, the text processing module normally starts with a sentence segmentation process to segment an input text such as abstracts or a full document into separate sentences. In order to perform sentence segmentation, the sentence boundary has to be determined. This is a non-trivial task especially in the biomedical domain due to the irregularities associated with entity names (e.g. dot-1.1), decimal values (e.g. 0.79), inline citations (e.g. Onye et al. 2016, p. 39), and abbreviations (e.g. i.v., i.e.).

ii. **Tokenization:** In this step, a sentence is broken into individual tokens sequentially in their order of appearance in the sentence. Normally, the tokens are separated by the white space between them, however, this step is also non-trivial due to the irregularities associated with tokens in the biomedical domain. Some of the possible errors in tokenization results from the presence of white spaces in biological entity names (e.g. HCFC 124, retinal toxicity) and

hyphenation where it is difficult to determine the number of tokens to be returned.

iii. **Part-of-Speech (POS) tagging:** In this step, the grammatical form of a token is identified based on the token's structure and the context derived from its neighbouring tokens. A complete sentence is sent to the POS tagger and it assigns a part-of-speech tag (e.g. verbs, adverbs, nouns, adjectives) to the individual tokens. This step helps cope with the ambiguities that arise from words with different grammatical forms depending on the context, therefore, it is a vital step in the syntactic analysis. An example part-of-speech tagged sentence is shown in Figure 2.2 B.

iv. **Named Entity Recognition (NER):** In order to properly identify the biomedical entity mentions, the NER is performed. The recognition of biological entity names can be non-trivial due to the complexities resulting from the non-standardized formats of the entity names [26]. Some NER toolkits (e.g. BANNER [27]) help to provide high accuracy and efficiency for this recognition process. According to Bikel et al. NER is an essential step in the workflow of any relation extraction system [28]. However, some relation extraction systems now rely basically on the entity annotations provided through gold standard annotated corpora which enable the systems to recognise the existing biomedical entity mentions (e.g. names of chemicals and drugs) in the document.

2.1.1.2 Parsing

Parsing is a process of determining the syntactic structure of a sentence by analysing its constituent tokens based on the grammar of the language. A parser consists of two

main components, the sentence and the grammar. Sentence parsing can either be shallow or full parsing.

Shallow Parsing: By using the POS information, sequences of tokens are grouped into their respective syntactic groups such as Noun Phrases, Verb Phrases as shown in Figure 2.2 C. Shallow parsing which is also referred to as chunking constructs a more structured POS tagged sentence and it is a prerequisite stage in the construction of a fully parsed sentence.

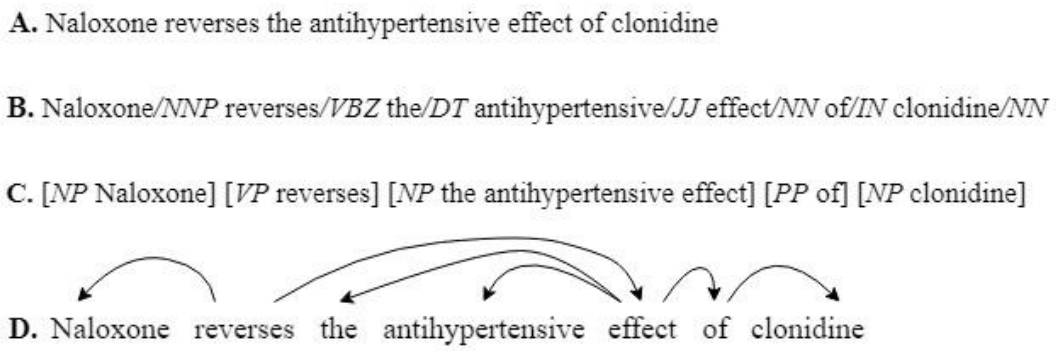

Figure 2.2. Part-of-speech tag and parser outputs of a sample sentence. A. The sample sentence. B. Part-of-speech tags for the sample sentence. C. The output of the shallow parser. D. The output of the dependency parser

Full Parsing: The combination of the outputs from the POS tagging and the shallow parsing helps a full parsing process to extract more information about the structural dependencies existing between phrases. Full parsing extracts the most elaborate information about the syntactic structure of a sentence to produce parse trees as outputs. An example of a full parser is the dependency parsing which finds the dependencies between the tokens. The output from dependency parsing is a parse tree consisting of leaves which are the tokens and edges representing the relationships between the tokens. An output from a dependency parser is shown in Figure 2.2 D.

2.1.1.3 Relation Extraction Module

This is the main module of any relation extraction system. Currently, different approaches are being employed to develop relation extraction systems. In this thesis, we have employed the ML-based approach and developed two different ML-based systems where one utilizes the GA as the optimization technique for MCS, and the other applies a combination of two ML classifiers.

2.2 Machine Learning

ML methods are the most commonly employed approaches in relation extraction. This section introduces the basic concepts of the ML methods and provides a discussion about their application in a relation extraction system. ML is classified as an artificial intelligence method and is based on statistical data to deduce general rules [29]. ML methods can be employed in multiple domains due to the advantage of their models being able to solve problems that are impossible to be represented or handled using explicit algorithms [29]. In general, irrespective of the difficulty in representation, ML models are capable of finding relations between the inputs and the outputs. This distinguishing ability has enabled the ML models to be successfully employed in many tasks such as classification, forecasting, pattern recognition and relation extraction. A review of the general ML algorithms including detailed algorithms for relation extraction can be studied [30].

In relation extraction, ML models utilize available data such as annotated text, sentences, to solve problems that are impossible to solve using explicit programming. ML models either learn rules by distinguishing data instances from each other or from the examples which reveal the structure of the underlying data. The outcomes of these ML models are either the learning rules or a prediction model that is used to predict

unknown data based on previously seen data [30]. For example, given biomedical data with a set of candidate chemical-disease relations, an ML method learns a model to predict this relation in unseen biomedical data [31]. This learning method corresponds to a supervised machine learning model that contains a training phase and a testing phase. During the training phase, the ML method creates a model through learning sets of properties with their own values of the examples given in the training data [31]. These properties are called features. The use of features helps the learning method to decide and group examples in classes (e.g. relation, no-relation). In order to differentiate the classes, most ML algorithms try to learn a set of distinctive values and combination particular to them. Additionally, in order to increase the robustness of a learning method and improve its computational efficiency, each example given in the training data must have distinguishing features [12]. The support vector machines (SVM) are one of the most frequently employed ML method for relation extraction tasks [32].

2.3 Multiple Classifier System

2.3.1 Introduction

The MCS consists of the use of a few different classifiers or a pool of base classifiers in the classifier ensemble. This approach has become an alternative to the use of a single classifier system in classification tasks. A single classifier system involves the selection of the most appropriate classification algorithm, parameter settings and feature subset for a given classification task. Due to a large number of classification algorithms and possible parameters, selecting the most appropriate classifier algorithm and finding the best settings for the parameters is not always trivial. Additionally, the selection of the optimum feature subset from the large set of features makes the process of selecting the best classifier even more complicated. In MCSs, an ensemble of the

base classifiers is constructed in order to generate the best possible classifier combination subsets from the base classifiers for the given classification task. In terms of classification tasks in different domains, the MCSs have proven to produce better accuracy compared to the individual classifiers that make up the ensemble [33, 34].

In designing an MCS, the individual classifiers are trained on samples labelled with corresponding class labels. The training data are mostly enhanced with discriminative features that are extracted from the given data set or from external knowledge resources. The individual predictions of the various classifiers are used in a predefined setting to classify new samples. For the total number of classifiers C in the ensemble, the output derived from the individual classifiers determines the possible ways they can be combined. Given a case where each classifier maps an input vector X to a specific class Y_i among N possible class labels, the outputs of the classifiers can be of either an abstract, rank, measurement or oracle level [35, 36]. An example of a typical MCS architecture is given in Figure 2.3.

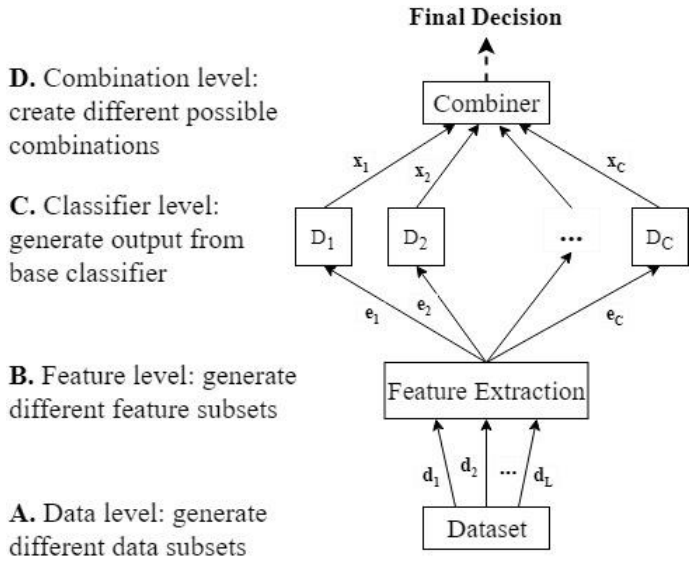

Figure 2.3. General Multiple Classifier System Architecture

The selection of classifiers for an ensemble is one of the most important decisions in an MCS framework. The classifiers that would produce the optimum solution when employed in the ensemble are selected at this phase. In most cases, the classifiers with weaker classification performances are not expected to be very useful when making the aggregate decision. The combination of similar classifiers is not expected to improve performance as they are expected to make similar mistakes. Therefore, the base classifiers in an MCS should possess some uniqueness and diversity. That is, they should have differences in their classification errors. Some types of diversity measures such as correlation, within-set generalization diversity, compound diversity, disagreement measures and Q statistics have been discussed by [37, 38]. In practice, the diversity among the base classifiers of an MCS can be increased by:

i. Applying the same classifiers with different parameter values.
ii. Using different classifier algorithms such as SVM, Decision Tree, Naïve Bayes, Bayes Network, etc. as the base classifiers.
iii. Using different feature combination on different classifiers.
iv. Training the classifiers on different training dataset or by using different subsets of the training data on different classifiers.

In other to design an MCS, decisions on the architecture of the base classifier, the type of output to be combined, and the selection criteria to be used for choosing the base classifiers D_i, from the repository of C classifiers and the fusion function F such that joint decision $D(\mathbf{x}) = F(D_1(\mathbf{x}),\ldots, D_C(\mathbf{x}))$, where $D_i(\mathbf{x})$ is the prediction of the i^{th} classifier for the given input $\mathbf{x}$.

2.3.2 Classifier Selection Criteria

The selection of the most suitable classifiers from a pool of classifiers is one of the most critical steps in designing an MCS. The classifier selection methods are grouped into two, namely: Static Classifier Selection (SCS) and Dynamic Classifier Selection (DCS). The main objective of these approaches is to achieve the optimum classification performance. However, there is one key distinction between them. In the SCS, the same set of classifiers in the ensemble is used in predicting all unseen samples while the DCS generally selects a set of different classifiers in the ensemble for each unseen sample. In the SCS approach, the base classifiers are trained on the training data and then by using the results of the combination from the development data the subset of classifiers with the optimum performance is selected. This selected ensemble is then used to classify the unseen data. However, for a corpus that has no development data, n-fold Cross Validation can be applied to the training set to find the optimum classifier subset to test the unseen data [37]. The DCS implementation has many approaches to finding an optimum classifier ensemble. In one of the approaches, the candidate classifiers expected to make the decision is dynamically determined based on the performance of the classifiers on the similar input values in the training data [36]. In another approach, the single best performing classifier or classifier ensemble in the neighbourhood of the unseen data is selected.

2.3.3 Search Algorithms used for Classifier Selection in MCSs

The search space containing all the candidate classifier combinations computed from all the present individual classifiers must be explored in order to achieve the most optimum classifier ensemble. This search can be performed using various methods such as Single Best, N Best, Forward Search, Backward Search, Exhaustive Search, and Evolutionary Search algorithms. The single and N best search algorithms

respectively consider the highest performing single or N classifiers from the pool of classifiers. The forward and backward search algorithms are typical examples of a greedy search algorithm and they stop when the evaluation function can no longer be improved from its current state with respect to the next step. The exhaustive search algorithm assumes that the number of the candidate classifier ensembles is small, making it impractical or unfeasible for an increase in the number of the base classifiers [39]. The evolutionary search is one method developed in order to avoid the challenges of an exhaustive search.

The process of an evolutionary search algorithm is better suited to handle a pool of a large number of classifier compared to the algorithms based on the greedy search approach. For the classifier selection process, these search algorithms have been successfully implemented in several studies [40]. Some of the evolutionary search algorithms are genetic algorithms, Bee Colony [41], Firefly [42, 43], and Ant Colony [44, 45]. GA is one most widely implemented evolutionary search algorithms [40, 46]. In this thesis, one of our methods for relation extraction in biomedical domain employed the GA as part of the designed architecture as discussed in Chapter 4. GA is a model that mirrors of a natural evolutionary system [47].

2.3.4 Genetic Algorithm

GAs are a type of optimization algorithm and have been used in fields like engineering and science as an adaptive algorithm for solving practical problems and also as a computational model of natural evolutionary systems [47]. Charles Darwin's theory of natural evolution inspired the heuristic search called GA [48]. This algorithm reflects the process of a natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

John Holland invented GAs in the 1960s which were later developed in the 1960s and the 1970s by Holland and his students and colleagues at the University of Michigan [47]. Intensive research has been dedicated to GA in order to bring about a lot of applications in the machine learning domain [49, 50, 51, 52]. Despite the numerous varieties of the GA present, the fundamental principles remain unchanged [46]. GA is designed to simulate a biological process, therefore, much of the relevant terminologies applied in GA is borrowed from biology. However, the entities that this terminology denotes to in GA are much simpler than their biological counterparts [53]. The main problem associated with GAs is the premature convergence to the local optima of the objective function [54]. The basic phases of GA are:

2.3.4.1 Initial Population

The population is made up of a set of individuals. Each individual is characterized by a set of parameters called Genes. Genes are combined into strings to form a chromosome. Chromosomes then make up the core GA which is usually represented by binary-encoded values (strings of 0s and 1s) of the candidate solutions to the optimization problem [55]. Each candidate solution is encoded as an array of parameter values [56]. For a problem with N dimensions, each chromosome is encoded as an N-element array.

$$\text{chromosome} = [p_1, p_2, \ldots, p_N]$$

Where each p_i is an actual value of the i^{th} parameter [56].

2.3.4.2 Fitness Function

The fitness function tests and quantifies the performance of a potential solution. The fitness of a chromosome is its ability to compete with other chromosomes and to achieve the desired task. The fitness of a chromosome is assigned a probability of survival proportional to its fitness.

2.3.4.3 Selection

Selection phase selects the fittest chromosomes for reproduction based on a user-defined probability distribution. The selected chromosomes pass their genes to the next generation of chromosomes. The common assumption is that only the fit chromosomes are selected and allowed to produce offspring. However, this would lead to the similarity of chromosomes in a few generations, and therefore decreased diversity. Two pairs of chromosomes called the parents can be selected based on a variety of some selection schemes such as roulette wheel selection, tournament selection, rank selection, and elitism. In this thesis, we employed the roulette wheel and tournament selection as discussed in Chapter 4.

2.3.4.4 Crossover

This is a very significant phase in GAs. The selected chromosomes are allowed to reproduce themselves through recombining and passing on their genotype to the next generation. For the mating pairs of chromosomes, a crossover point is chosen either by default or randomly from within the genes. Offspring are produced by exchanging some of the parents' genes among themselves until the chosen crossover point is reached. The new offspring are added to the population. For the one-point crossover, after the selected crossover point, the tails of the parent chromosomes are swapped to produce new offspring. If two or more crossover points are taken, then it is a multi-point crossover. In this thesis, we employed a one-point and two-point crossover as discussed in Chapter 4.

2.3.4.5 Mutation

The mutation operation generally involves a small change in the genotype of an individual to which it is applied. This change happens at a frequency called the

mutation rate. Mutation rate is defined as the probability at which a selected position of the genotype in an individual is mutated in every iteration of the GA evolution [54].

2.4 Biomedical Corpora

High-quality biomedical corpora are very important in the development of relation extraction systems. The creation of biomedical corpora is a time-consuming and error-prone task which has made the number of biomedical corpora available to be small. The Protein-Protein Interaction (PPI) task is one of the largest studied relation extraction task in the biomedical domain. Pyysalo et al [1] converted the five major PPI corpora (AIMed [57], BioInfer [58], HPRD50 [59], IEPA [60], and LLL [61]) into a unified format. A full list of the available biomedical corpora can be found in the [62]. Some biomedical relation corpora are listed in Table 2.1.

Table 2.1. Biomedical relation extraction corpora

Datasets	**Description**
AIMed	Protein-protein interactions
AnEM	Anatomical Entity Mention corpus
BioCreative 2 Gene Mention task	Gene and protein mentions
BioCreAtIvE II and III	Protein-protein interactions
BioCreAtIvE V	Chemical-disease relations
BioInfer	Protein-protein interactions
BioNLP 2009 and 2011 Shared Task	Biological events, such as gene expression, regulation, phosphorylation, etc.
BioText	Disease and treatment relationships
Drug-Drug Interaction Extraction 2011 and 2013	Drug-Drug Interaction
HPRD50	Protein-protein interactions
IEPA	Protein-protein interactions
LLL	Protein-protein interactions

Chapter 3

EXPERIMENTAL SETTINGS

3.1 Dataset

The BioCreative V corpus [4] contains 1500 documents [63] having only titles and abstracts. These articles are grouped into the training, development datasets and test dataset each of them having 500 articles. The training, development datasets and 400 articles of the test dataset were selected randomly from the Comparative Toxicogenomics Database (CTD)-Pfizer corpus [64]. This corpus was produced through the curation collaboration between CTD and Pfizer and consists of over 150,000 chemical-disease relations in 88,000 articles [64]. The remaining 100 articles for the test set were selected through a process ensuring that they would contain a similar distribution of entities as the training and development datasets. The entire BioCreative V corpus is manually annotated with the chemicals, diseases and their relations; the entity mentions have unique concept identifiers. Table 3.1 shows the statistical information on this corpus [63, 65].

Table 3.1. Statistics of the BioCreative V dataset

Dataset	**Articles**	**Sentences**	**Chemicals**		**Diseases**		**No. of CID relations**	**Average sentence lengths**	
			Mention	**ID**	**Mention**	**ID**		**Original**	**Simplified**
Training	500	4519	5203	1467	4182	1965	1038	20.70	19.25
Development	500	4395	5347	1507	4244	1865	1012	20.44	19.03
Test	500	4759	5385	1435	4424	1988	1066	20.38	18.03

3.2 Base Classifiers

3.2.1 Support Vector Machines (SVMs)

The support vector machine is a classifier designed for binary classification and it is one of the most commonly used ML classifiers in biomedical relation extraction tasks such as PPI [12, 13], text categorization [17], and CID relation extraction [66, 67]. During the training phase, SVM finds the optimal hyperplane separating two classes by maximizing the margin between the hyperplane and a subset of the training data points, called the support vectors that are nearest to the hyperplane. During the testing phase, the input vectors are classified as positive or negative depending on the side of the hyperplane they are mapped to. In order to compute the separating hyperplane for data that are not linearly separable, SVM utilizes a kernel function to transform the data into a higher dimensional space where it can be separated linearly. Some of the kernel functions used in SVM are polynomial, radial basis function (RBF), Gaussian, and sigmoid kernels. Figure 3.1 shows a hyperplane separating two classes that are linearly separable in two-dimensional space.

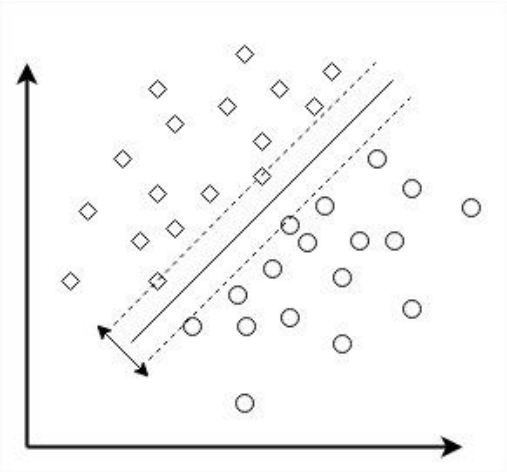

Figure 3.1. An example of a linearly separable binary classification task. A hyperplane separates the two classes (square and circle).

3.2.2 Decision Trees

3.2.2.1 J48 Algorithm

The J48 algorithm is an extension of the conventional decision tree algorithm, Iterative Dichotomise 3, and an implementation of the C4.5 algorithm by Ross Quinlan, [68]. The main advantage of the C4.5 algorithm over other decision trees is its good combination of both error rate and speed [69]. Some of the improvements of this algorithm are: (1) it generates the rules for the prediction of the target class, (2) it can process both numeric and discrete data, (3) it produces easily interpreted rules, (4) it prunes trees after they have been created in order to remove branches that are causing obstruction from reaching the leaf nodes and (5) it handles missing attribute values [68, 70].

The J48 uses the measure of data disorder called "Entropy". The Entropy $(\vec{z})$ is computed as [71]:

$$Entropy\ (\vec{z}) = -\sum_{i=1}^{n} \frac{|z_i|}{|\vec{z}|} \log\left(\frac{|z_i|}{|\vec{z}|}\right) \tag{3.1}$$

$$Entropy\ (i|\vec{z}) = \frac{|z_i|}{|\vec{z}|} \log\left(\frac{|z_i|}{|\vec{z}|}\right) \tag{3.2}$$

This algorithm aims at maximizing the gain, which is computed as:

$$Gain\ (\vec{z}, i) = Entropy\ (\vec{z}) - Entropy\ (i|\vec{z}) \tag{3.3}$$

The pruning step in the J48 decision tree, which is an implementation of the C4.5 algorithm, is performed in order to improve the generalization of the tree and it greatly affects the final performance. In order to reduce the error rate, the C4.5 algorithm performs pruning by replacing the internal node with a leaf node [72]. This algorithm performs an enhanced tree pruning method which reduces the misclassification errors

that may occur due to noise, redundant, irrelevant or too much information available in the training dataset [73]. It sorts the data at every node, thereby determining the best splitting attribute [73].

3.2.2.2 Random Forest

The random forest is a supervised learning algorithm. It is trained using the bagging method, which is based on the idea that the combination of learning models improve the final performance [74]. It fits a number of decision tree classifiers on multiple sub-samples of the dataset and uses averaging to enhance the predictive accuracy and to control overfitting. In essence, the random forest creates multiple decision trees and combines them to produce a more stable and accurate prediction. Another advantage of the random forest is its ability to be employed in both classification and regression tasks.

During the growing of the trees, the random forest algorithm adds randomness to the model by searching for the node to split among a random subset of features rather than choosing the most important feature in the complete set [74]. This approach increases diversity, which generally results in a better model. However, the random forest may not be the best classifier selection for a real-time classification task as the building of large subsets of trees can make the algorithm very slow.

3.2.2.3 Random Tree

The random tree is a supervised classifier that can be used for both classification and regression problems. The random tree is referred to as an ensemble-learning algorithm as it learns through the generation of various individual learner [74]. Just like the random forest, it also employs the bagging idea; however, unlike in the random forest where every node is split using the best features among the subsets of features chosen randomly, the random tree splits every node using the best split among all the features.

The classification mechanism for a random tree is as follows: (1) the random tree is assigned input vector, (2) it classifies the input vector with every tree in the forest, and (3) it gives the class label with the majority vote as the output [74].

Random trees are a combination of model trees and random forest since the model trees are merged with the ideas of random forest. This combination is used for split selection; thereby it creates balanced trees where one global setting for the ridge value works across all leaves, thus simplifying the optimization procedure [75, 76].

3.2.3 Bayesian Classifiers

The types of Bayesian Classifiers implemented in this work are Naïve Bayes (NB) and Bayes Networks (BN). These classifiers both deal with conditional probabilities.

The NBs methods are supervised learning algorithms and are based on the Bayes' theorem which makes two naive assumptions: (1) conditional independence between the predictive attributes given the value of the class variable, and (2) no hidden or underlying predictive attributes influence the prediction process [77]. The Naïve Bayesian model is simple and easy to build and is useful for very large datasets, as the model does not consist of complicated iterative parameter estimations. Despite its simplicity, it is still capable of outperforming some more sophisticated classifier algorithms, thereby making it a widely used model for classification tasks.

Considering a random variable C denoting the class of an instance and F denoting a feature vector of random attributes denoting the observed attribute values, let c denote a given class label and f denote a particular observed attribute value vector. The Bayes' theorem states that the probability of each class given the feature vector,

$$P(C = c \mid F = f) = \frac{P(C = c)\, P(F = f \mid C = c)}{P(F = f)} \tag{3.4}$$

Considering the naive assumption of the conditional independence between features, and since the event is a conjunction of the predictor value assignment such that $F_1 = f_1 \wedge \ldots F_{m-1} = f_{m-1} \wedge F_m = f_m$, then we obtain:

$$P\,(F = f \mid C = c) = P(\bigwedge_i F_i = f_i \mid C = c) \tag{3.5}$$

$$= P(\prod_i F_i = f_i \mid C = c) \tag{3.6}$$

Generally, the distribution of the denominator in Equation (3.4) is not directly estimated as it is just a normalizing factor; instead, the denominator is ignored and the equation is normalized so that the sum of P(C = c | F = f) is one over all the classes [77].

The BN combines a powerful knowledge representation and reasoning mechanism to represent events and causal relationships between them as conditional probabilities involving random variables [78]. Given the values of a subset of these variables, the BN can compute the probabilities of another subset of variables [78]. BN builds on the same principle as the NB; however, they are not restricted to representing distributions based on the strong independence assumptions of the Bayes theorem employed in the NB model. The BN gives the flexibility to build a representation of the distribution to the independence properties that appear reasonable in the current setting.

3.3 Text Processing

This section relates to the methods of sentence segmentation, tokenization and simplification used to process the input text employed in the relation extraction methods discussed in Chapter 4 and Chapter 5.

3.3.1 Sentence Segmentation and Tokenization

In the BioCreative V dataset, the relation instances can be considered on both the abstract and sentence levels [79]. Our methods performed the relation extraction task on the sentence-level. As already discussed in Section 2.1.1.1, the irregularities in the biological names, decimal values, inline citations, and abbreviations amongst other problems make this a non-trivial task and may require the use of external NER toolkits to improve efficiency and accuracy. During sentence segmentation, the input data are documents containing only titles and abstracts. Normally, a period (.) is used to identify the sentence boundaries in a paragraph. However, considering the single sentence from the article with PubMed ID (PMID) 20067456: "Total ASEX scores were significantly lower, i.e. better, among men who received bupropion than placebo, at 15.5 (4.3) vs 21.5 (4.7) (P= 0.002).", it can be seen that splitting simply at periods will result in multiple sentences, thereby, making sentence segmentation a non-trivial task.

Additionally, the process of tokenization also can become a non-trivial one due to issues such as entity names with white space (composite tokens) and the inconsistent use of hyphenation in the entity names. For example, "Lithium-induced" should return two tokens whereas, "alpha-methyldopa" should be treated as a single token. The irregularities with biological terminologies (e.g. 1,3-bis-(2-chloroethyl)-1-nitrosourea) also make this process more complicated. It is worth mentioning that the errors

generated in the tokenization step are transmitted through the subsequent text pre-processing steps, thereby decreasing the performances of the subsequent tasks [80].

3.3.2 Sentence Simplification

In this section, we discuss the methods utilized for simplifying the sentences after sentence segmentation and tokenization stages. The sentences used in this step contain at least one entity mention. Due to the irregularities associated with the entity names, they are replaced by placeholders, which are unique names. The entity names considered during this process are entirely those extracted through the gold standard annotation provided in the BioCreative V CDR corpus.

The corpus consists of both simple and composite entity mentions. A simple entity mention consists of either a single token or multiple tokens separated by white space (e.g. Suxamethonium, blurred vision). However, a composite entity mention consists of multiple tokens mainly combined using white space (e.g. cholestatic hepatitis), conjunctive operator (e.g. pleural and pericardial effusion), disjunctive operator (e.g. endometrial hyperplasia or cancer) or comma in conjunction with an operator (e.g. a decrease in MAP, HR, SV, and CO). The composite entity mentions may contain embedded entities, therefore, a set of rules are defined in our system in order to efficiently and accurately extract all the embedded entities in the composite mentions and replace them with the placeholders. The handling of a composite entity mention is non-trivial. For example, consider the excerpt “THE FIELD: **Fluoropyrimidines**, in particular **5-fluorouracil** (**5-FU**), have been the mainstay of treatment for several solid **tumors**, including **colorectal, breast** and **head and neck cancers**, for > 40 years.” from the article with PMID 20722491. The bold words are entity mentions according to the gold standard annotation. The composite entity mentions “colorectal, breast and head and neck cancers”, with the composite concept identifier

D015179|D001943|D006258 that represents the three embedded entities “colorectal cancers”, “breast cancers”, and “head and neck cancers” respectively.

The placeholders used to replace the entity names are a tag (ARG) with a numeric value (0, 1, 2 …) (e.g. ARG0, ARG1). These placeholders are used to replace the entity mentions in their order of appearance in order to keep a proper reference for each entity. After the entity mentions have been replaced by the placeholders, in the next phase of the sentence simplification, all the parentheses not containing any placeholders were eliminated (tokens and parentheses) from the text. After the sentence simplification process, simplified sentences are generated with the aim of improving the generalization ability of the dependency parser employed. After simplification, the sample sentence given above becomes “THE FIELD: **ARG0**, in particular **ARG1** (**ARG2**), have been the mainstay of treatment for several solid **ARG3**, including **ARG4, ARG5** and **ARG6**, for > 40 years.”

The simplified sentences are then passed to the relation instance construction step, where the candidate relation instances are extracted. The average lengths of the original and simplified sentences are reported in Table 3.1, where it can be seen that the average length of the simplified sentence is lesser than the original sentence.

3.4 Relation Instance Construction

In this step, we extract the candidate CID relation instances from the input sentence. Given a pair of entities expressed as either <chemical, disease> or <disease, chemical>, a CID relation can exist in either intra-sentence or inter-sentence mention level. As reported in this thesis, the intra-sentence level describes the case where a chemical and disease mention occur in the same sentence. Whereas, the inter-sentence

level describes the case where the mentions occur within neighbouring sentences. Two sets of filtering rules and four-sentence structural forms are employed during the CID relation instance construction.

In order to construct the sentence structural forms, a triplet is built based on the candidate relation instances (e.g. <ARG_a, REL, ARG_b> where ARG_a and ARG_b are placeholders representing the candidate relation instances and *a* and *b* being numeric values showing the order of appearance of an entity (argument) in the sentence. $a < b$ for every triplet). The variable REL in the triplet signifies a word that can be extracted from the sentence based on any of the structural forms to provide a clue of a possible relation between the instances. Therefore, the variable REL is called a clue word. The extracted clue words are either nouns or verbs that exist before, between or after the entity pair in a candidate relation instance. In the structural sentence forms listed below, token* represents zero or more tokens [65].

Form 1: ARG_a token* REL (verb) token* ARG_b

Example: ARG_a-inducing effect of ARG_b.

Form 2: ARG_a token* REL (noun/verb) token * ARG_b

Example: ARG_a may have an adverse effect on ARG_b.

Form 3: REL (noun) token * ARG_a token* ARG_b

Example: interaction between ARG_a and ARG_b.

Form 4: ARG_a token* ARG_b token * REL (noun/verb)

Example: ARG_a following ARG_b administration.

These rules were applied to determine if the CID candidate instances have a higher probability of being a true CID relation. The process of sentence construction at the

intra- and inter-sentence levels is discussed in detail in Sections 3.4.1. The two set of rules are each dependent on the sentence levels been considered as follows:

Rule 1: For the candidate relation instances (triple) in the intra-sentence mention level:

i. The total number of tokens existing between the entity mentions in the candidate relation instance must not be greater than 10.
ii. If there exist multiple candidate relation instances describing the same CID relation, the instance with the closest distance (token-wise) between the entity mentions is selected.

Rule 2: For the candidate relation instances (triple) in the inter-sentence mention level:

i. Any candidate relation instance considered at this level must not exist in the intra-sentence level,
ii. Since composite sentences are used to construct a candidate relation instance at this level, the number of combined sentences must not be greater than three (i.e. two or three),
iii. As in the intra-sentence level, if there exist multiple candidate relation instances at this level describing the same CID relation, the instance with the closest distance (token-wise) between the entity mentions is selected.

3.4.1 Relation Instances on the Intra-sentence and Inter-sentence Levels

The intra-sentence mention level describes the case where the entity mentions of a candidate relation instance are extracted from the same sentence, whereas, the inter-sentence mention level describes the case where the entity mentions of a candidate relation instance that are extracted from neighbouring sentences. In the inter-sentence level, the neighbouring sentences considered are merged to form a composite sentence consisting of a single sentence boundary.

In order to describe the existence of the candidate relation instances in an input text, let us consider an excerpt of the title and abstract of a document (PMID 2234245) as shown in Figure 3.2. The abstract is segmented into sentences and as discussed in Section 3.3.2, only the sentences with at least one entity mention are considered in our approach, therefore, the sentence S3 is eliminated. The sentences with just one entity mention are used since they can be a member of a candidate relation instance that spans a number of neighbouring sentences. According to the BioCreative V corpus annotation, the document given in Figure 3.2 consists of three CID relations between the entity mentions <D003676, D014786> <D003676, D012164>, and <D003676, D006319>. The first relation occurs in the same sentence, Title, between "desferrioxamine$_{D003676}$" and "Ocular$_{D014786}$ (Ocular toxicity)". This case is referred to as the intra-sentence level. Whereas, the latter two relations between "desferrioxamine$_{D003676}$" and "pigmentary retinal deposits$_{D012164}$", and "desferrioxamine$_{D003676}$" and "neurosensorial hearing loss$_{D006319}$" occur over multiple neighbouring sentences, between S4 and S6, and S5 and S6 respectively. These cases are referred to as the inter-sentence level.

Title Ocular$_{D014786}$ and auditory toxicity$_{D006311}$ in hemodialyzed patients receiving desferrioxamine$_{D003676}$.

S1 During an 18-month period of study 41 hemodialyzed patients receiving desferrioxamine$_{D003676}$ (10-40 mg/kg BW/3 times weekly) for the first time were monitored for detection of audiovisual toxicity$_{D014786|D006311}$.

S2 6 patients presented clinical symptoms of visual$_{D006311}$ or auditory toxicity$_{D014786}$.

S3 Moreover, detailed ophthalmologic and audiologic studies disclosed abnormalities in 7 more asymptomatic patients.

S4 Visual toxicity$_{D014786}$ was of retinal origin and was characterized by a tritan-type dyschromatopsy, sometimes associated with a loss of visual acuity$_{D014786}$ and pigmentary retinal deposits$_{D012164}$.

S5 Auditory toxicity$_{D006311}$ was characterized by a mid- to high-frequency neurosensorial hearing loss$_{D006319}$ and the lesion was of the cochlear type.

S6 Desferrioxamine$_{D003676}$ withdrawal resulted in a complete recovery of visual function in 1 patient and partial recovery in 3, and a complete reversal of hearing loss in 3 patients and partial recovery in 3.

Figure 3.2. A sample document (PMID 223424) showing CID relations. The chemical entities are highlighted in dash lines and the disease entities are highlighted in solid lines. The CID relations are between the entities <D003676, D014786> <D003676, D012164>, and <D003676, D006319>.

The construction of a candidate CID relation instance depends on the two set of defined rules as described above. Rule 1 is employed if the candidate instances are on the intra-sentence level, otherwise, Rule 2 is employed.

3.4.2 Relation Instances on the Joint Level

In this thesis, we develop a case called the "joint level" where the candidate relation instances from both the intra- and inter-sentence levels are combined after they are extracted before the classification. Since the relation instances from the intra- and inter-sentence levels are non-overlapping, their combination only leads to the generation of the complete relation instances in the dataset. The construction of the candidate relation instances on the three sentence levels generates three subsets of given datasets namely; intra-sentence level dataset, inter-sentence level dataset and joint level dataset. Table 3.2 reports the statistics on candidate relation instances extracted across the three datasets from the BioCreative V corpus. The positive instances are those entity pairs that have been annotated by the corpus to possess a true CID relation between them, while the negative instances are the entity pairs not annotated as such. This table shows that the similar distribution of the positive and the negative instances across the datasets are similar.

Table 3.2. Candidate relation instances from the BioCreative V corpus

Datasets	**Intra-sentence**		**Inter-sentence**		**Joint level**		**Total**
	Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	
Training	277	524	761	3102	1038	3626	4664
Development	244	622	768	3409	1012	4031	5043
Test	315	549	751	3426	1066	3975	5041

3.5 Feature Extraction

Feature extraction or selection is the process of extracting a subset of features from the complete set through the means of functional mapping [81]. Feature extraction can also be seen as the process of generating all possible transformations from the original set of features in order to find an optimum subset, which with the lowest possible dimensionality can aid in preserving class separability within the space [82]. In the recognition field, Duda et al. simply refer to the term "feature extraction" as a process of extracting features from data [83]. The process of feature extraction covers a number of disciplines such as data mining [84], machine learning [12, 14, 85, 86, 87, 88], and pattern recognition [89, 90]. Feature extraction helps in reducing computational complexity and dimensionality as well as solving the problem of overfitting [81]. Overfitting is experienced when a classifier model can correctly classify data points that are very closely related to the training data but then performs poorly with the data which are not closely related to the training data [91]. Little knowledge exists on describing features that would be relevant for a classification task, therefore, many subsets of candidate features from the original feature set are introduced and this creates a case of having the existence of redundant or irrelevant features in a given subset [92]. The relevant features are neither redundant nor irrelevant to the task; the redundant features add no new information to improve the task, and the irrelevant features are not directly associated with the target concept, however, they affect the learning process [92]. Therefore, selecting an optimum subset from the original feature set is a tough task. For example, let $\mathbf{Y}$ be the original feature space having a cardinality of p, and $\overline{\mathbf{Y}}$ is the selected feature sub-space with a cardinality of $\bar{p}$. The criterion for the selected feature sub-space $\overline{\mathbf{Y}} \subseteq \mathbf{Y}$ is $K(\overline{\mathbf{Y}})$. Without a loss of generality, an

assumption can be made that a higher value of K will indicate a better feature space. Therefore, difficulty arises in selecting a feature sub-space $\overline{\mathbf{Y}} \subseteq \mathbf{Y}$ such that

$$K(\overline{\mathbf{Y}}) = \max_{W \subseteq \mathbf{Y}, |\mathbf{W}| = \bar{p}} K(\mathbf{W}) \tag{3.7}$$

For an exhaustive approach, all the possible combinations of $\binom{p}{\bar{p}}$ must be considered. Due to the exponential increase in the number of possible combinations, this exhaustive search is impractical and unfeasible for large p values. It is usually intractable to compute the best feature subset [93], and many problems associated with feature selection have shown to be nondeterministic polynomial-time hard (NP-hard) [94], thereby describing the process as remaining a trial-and-error skill-dependent task [9].

3.5.1 Features Used

As discussed in Section 3.4.1, in the intra-sentence level, a relation instance is extracted from in a single sentence, whereas, in the inter-sentence level, a relation instance is extracted from a composite sentence. Both single sentences and composite sentences are used in the same way during feature extraction. The types of feature extracted in this thesis have been used successfully in the relation extraction tasks [7, 9, 10, 12, 95]. The feature sets are grouped into three categories: (1) contextual, (2) dependency, and (3) statistical features.

3.5.1.1 Contextual Features

The contextual features employed in this thesis consist of the names of the entities in a candidate relation instance and a clue, word which describes any relationship present. The contextual features are described in Table 3.3.

Table 3.3. Description of the contextual features

No.	Description	Format
1	Chemical mentions	String
2	Disease mentions	String
3	Relation clue words (REL)	String

3.5.1.2 Dependency Features

The dependency features have been employed successfully to provide effective information in order to determine CID relations between entity mentions in other systems such as [8, 12, 95, 96]. They extracted from a dependency parse tree developed using the SpaCy[1] dependency parser. The parsed sentences are the simplified sentences generated after text processing. Table 3.4 gives the information about the dependency features employed in this thesis.

Table 3.4. Description of the dependency features

No.	Description	Format
1	POS tags along the path from the root node to the first entity in the candidate relation instance (ARG1).	String
2	The POS tags along the path from the root node to the second entity in the candidate relation instance (ARG2).	String
3	The node distance from the root node to ARG1	String
4	The node distance from the root node to ARG2	String

3.5.1.3 Statistical Features

The statistical features describe the attributes of the entity mentions or tokens present in the considered sentences in either Boolean (binary representation of 1 or 0) or

[1] Spacy parser: https://spacy.io/docs/usage/

numeric (frequency of occurrence) formats. For the Boolean representation, 1 denotes "true" while 0 denotes "false". The statistical features are presented in Table 3.5. The phrase "around an entity" refers to the defined context window containing four words on each side of a given entity mention.

Table 3.5.Description of the statistical features

No.	Description	Format
1	Number of REL extracted using the four defined sentence structure forms.	Numeric
2	The number of verbs in the considered sentence	Numeric
3	The number of verbs between ARG1 and ARG2	Numeric
4	The number of tokens between ARG1 and ARG2	Numeric
5	The number of tokens between ARG1 and REL	Numeric
6	The number of tokens between REL and ARG2	Numeric
7	The number of chemical mentions in the sentence	Numeric
8	The number of disease mentions in the sentence	Numeric
9	The number of both the chemical and disease mentions in the sentence	Numeric
10	Does the title contain ARG1	Boolean
11	Does the title contain ARG2	Boolean
12	Does the title contain both ARG1 and ARG2	Boolean
13	Are the words "increase" or "decrease" around the chemical mention	Boolean
14	Are the words "increase" or "decrease" around the disease mention	Boolean
15	Are nouns representing persons like 'infant', 'adult', or 'patient', around the chemical mention	Boolean
16	Are nouns representing persons like 'infant', 'adult', or 'patient', around the disease mention	Boolean
17	Are keywords like 'mg/kg', 'mmol/l', or 'mg/dl' around the chemical	Boolean
18	The absence of a token between ARG1 and ARG2	Boolean
19	The presence of only a single token between ARG1 and ARG2	Boolean
20	The existence of a verb between ARG1 and ARG2	Boolean

3.6 Evaluation Methods

The relation extraction systems described in Chapter 4 and Chapter 5 are evaluated using Recall (R), Precision (P), and the F-score (F1) metrics. In order to calculate these metrics, the variables true positives (TP), false positives (FP) and false negatives (FN) are used. These variables are numbers returned by the system after classification. For the CID relation extraction task, TP denotes the number of the CID relation instances that are correctly predicted, FP denotes the number of relation instances that are incorrectly predicted as CID and FN denotes the number of CID relation instances that are unidentified as such by the classifier. F1 is used to evaluate the systems reported in this thesis and it is computed using recall and precision.

$$Recall\ (R)\ =\ \frac{TP}{(TP+FN)} \tag{3.8}$$

$$Precision\ (P)\ =\ \frac{TP}{(TP+FP)} \tag{3.9}$$

$$F-score\ (F1) = \frac{2RP}{(P+R)} \tag{3.10}$$

Chapter 4

CID RELATION EXTRACTION TASK USING GENETIC ALGORITHM WITH TWO VOTING METHODS FOR CLASSIFIER SUBSET SELECTION

Relation extraction has become an important forerunner for text-mining problems. In this chapter, we report a novel framework to facilitate the development of a multi-classifier system for CID relation extraction task. The system utilizes the GA optimization technique. The classifier ensembles are represented as chromosomes where each bit represents the participation of a classifier in the ensemble as reported in [97]. The genetic framework employed in this work contains three important design features: (1) Each chromosome contains an extra bit called the voting bit to determine the voting algorithm used for the combination of the classifiers in the ensemble, (2) Two different selection algorithms and two types of crossovers are used randomly during the evolution process, and (3) Two decision-making under uncertainty techniques are used as the voting methods. The third feature is an important contribution to the current work.

The system developed using this framework aims at producing a good combination of classifiers by utilizing the diversity of the classifiers in an ensemble. During the validation of our system, nine (9) experimental settings were employed. All the settings produced good results comparable to the state-of-the-art systems, thereby, justifying our approach. Although candidate relation instances are generated to form

three sentence levels as discussed in Section 3.4, only the dataset subset on the joint level is applied in this framework.

4.1 Background

There has been an increase in the scientific effort dedicated to improving knowledge discovery in biomedical texts. The relation extraction task has seen multiple systems developed to propose unique approaches and produce better performances. The single classier-based systems are predominant for classification tasks [11, 12, 14, 98], however, an MCS has also been considered [36, 99, 100]. In the classification tasks of multiple non-trivial pattern recognition problems, the MCS reportedly provides better performances [101, 102, 103]. The performance of an MCS is determined by the diversity or complementarity of the base classifiers [104].

One of the core modules of an MCS is the Classifier Subset Selection (CSS). In the CSS module, a subset of classifiers is selected in an ensemble from the base classifiers such that the performance of the ensemble is better than that of the ensemble of all the base classifiers and the best individual classifier [100, 104, 105, 106]. The research on classifier diversity and classifier selection has been aimed at investigating the reasons behind the different performance levels associated with the CSS approaches [100, 103, 105, 107, 108].

In general, in order to achieve good performance, MCSs are constructed with well performing and diverse base classifiers. Therefore, the overall performance of an MCS heavily depends on the selection of the base classifiers and their individual performances [109, 110]. The selection and combination of features for training influences the performance of the individual base classifiers [105].

In this framework, the relation extraction method described employs a multidimensional classifier selection approach through GA. GA, which is an optimization technique, provides a variety of options to deal with the complexity between the search algorithm used and the solution found [105]. Complementarity or diversity among the base classifiers in an MCS can be improved through the variations of the parameters of the classifiers, the use of different subsets of training dataset, and feature subsets [104, 111]. We implemented the novel use of two decision-making under uncertainty techniques as our voting methods for the classifier combination and the use of two randomly selected classifier selection techniques. Diversity within the base classifiers is increased through the use of different classifiers and classifiers tuned to different parameter settings in the base classifier, and the use of different feature subsets for training. Additionally, we added some variation during evolution by using two different randomly selected selection techniques and two types of crossovers.

4.2 Methods

This chapter continues in the direction of Chapter 3, where we discussed the experimental setup used for our relation extraction methods. The dataset used in this framework which is the joint level dataset, the base classifiers, the stages of text processing (sentence segmentation and sentence simplification), the construction of candidate relation instances, and feature extraction have been discussed in details in Section 3.1. Based on the extracted features described in Section 3.5.1, we developed four feature sets based on the three different feature categories as described in Table 4.1. For example, Set A contains the three feature categories.

Table 4.1. Feature sets used for training

Sets	Contextual	Dependency	Statistical
A	X	X	X
B	X	X	
C	X		X
D		X	X

4.2.1 Classifiers

The base classifiers employed include the SVM, three implementations of the Decision Trees (the J48, random forest (R4) and random tree (R3)), and two implementations of the Bayesian Classifiers (Naïve Bayes and Bayes Network). Table 4.2 presents the complete detail on the base classifiers and their individual parameter settings.

Table 4.2. Base classifiers and their parameter settings

S.NO	Classifiers	Settings
1.	Bayes Network	Search Algorithms: 1) HillClimber (as BN Hill) • Initial network: True • Number of parents per nodes: 2 • Score type: Bayes 2) K2 (as BN K2): similar to BN Hill but it is restricted by an order on variables • Initial network: True • Number of parents per nodes: 2 • Score type: Bayes 3) TAN (as BN TAN): determines the maximum weight spanning tree and returns a Naive Bayes network augmented with a tree. • Use Markov Blanket correction: false • Score type: Bayes
2.	J48	4) J48 • Confidence factor: 0.25 • Batch size: 100 • Minimum number of instances per leaf: 2
3.	Naïve Bayes	5) NB • Kernel estimator: inactive 6) NBK • Kernel estimator: active
4.	Random tree (R3)	7) R3 • Batch size: 100 • Number of randomly chosen attributes: 0 • Minimum proportion variance: 0.001 • Seed: 1
5.	Random forest (R4)	8) R4 ▪ Batch size: 100 ▪ Maximum depth: 0 ▪ Number of randomly chosen attributes: 0 ▪ Number of iteration: 100
6.	SVM	9) SVM1 ▪ Complexity: 0.6 ▪ Kernel: polynomial 10) SVM2 ▪ Complexity: 0.0 ▪ Kernel: polynomial

The base classifiers are trained using the feature sets A, B, C and D. The initial performances of the base classifiers are determined when evaluated using the feature set A. The other three feature sets, Sets B, C and D, which are subsets of Set A are used for training in order to increase the diversity among the base classifiers. This process led to different levels of classification success.

The six different ML classifiers shown in Table 4.2 are used in either different parameter settings or implementations to produce an initial number of 10 base classifiers. An introduction of the four different feature sets for training the initial base classifiers increased the number of the base classifiers to 40, as each of the classifiers was trained separately on the four feature sets. Therefore, a total of 40 classifiers were used as our base classifiers. The base classifiers are trained using the joint level training data and the outputs of the base classifiers from the joint level development dataset are used during the evolution process of our system. The optimum ensemble generated after the evolutionary process is used to evaluate the performance of our system on the joint level test dataset. The statistics of the joint level datasets are reported in Table 3.2.

4.2.2 Genetic Algorithm Framework

In the MCS settings for the GA framework, a subset of classifiers is represented by a string of binary values called chromosomes, with ‘1’ or ‘0’ at a location *i* denoting the presence or absence of classifier *i*. A group of chromosomes is termed a population and the population evolves in every generation through the application of selection, crossover and mutation processes. These processes are employed to generate possibly better chromosomes in every generation while aiming for an eventual convergence towards an optimal solution.

During selection, some of the chromosomes are randomly selected for reproduction. This selection is performed mainly using the fitness level of the chromosomes, such that the fittest ones have a greater probability for reproduction. The chromosomes selected for reproduction are called parents and the products of their reproduction are called offspring. In our approach, the two selection methods employed are the Roulette Wheel and the Tournament selection methods.

i. **Roulette Wheel Selection:** The parents are selected based on their relative fitness within the population. Therefore, the chromosomes with better fitness have more chances to be selected. The probability $prob_i$ of selecting an individual *i* is given by,

$$prob_i = \sum_{i=j}^{N_j} f_i \tag{4.1}$$

where f_i is the fitness of the individual and *N* is the population size.

ii. **Tournament Selection:** This method selects the chromosome with the highest fitness from a randomly selected subset of the population. The size of the subset controls the selection pressure as a bigger subset size causes an increase in the selection pressure.

The processes of crossover and mutation are performed after the selection of the parents. These two operations are performed to increase the variety of individuals in the population, thus, increase the chances of avoiding a convergence towards the local optimum [54]. When the termination condition is met or after a predefined number of generations, the fittest chromosome in the population is considered as the optimal MCS solution. Figure 4.1 describes the flowchart of our GA system.

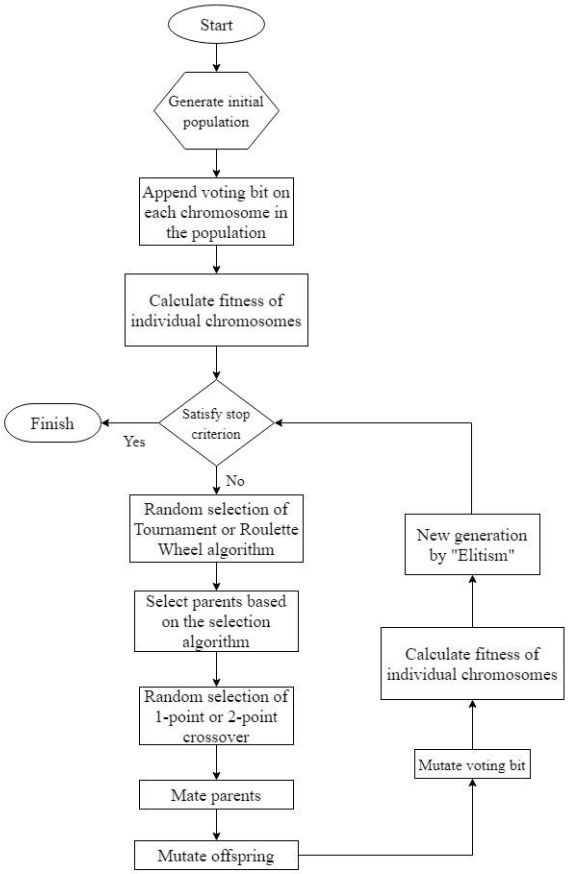

Figure 4.1. Flowchart of Genetic Algorithm

For a population of size N, C_i (where $1 \leq i \leq N$) are the chromosomes representing classifier ensembles where each chromosome contains M bits such that the first $M-1$ bits represented by 0 or 1 in location i denotes the absence or presence of a classifier respectively and the last bit shows the voting method used in the ensemble as shown in Figure 4.2. The choice of voting method depends on the voting bit where the bits 0 and 1 represents “Minimax Regret” and “Hurwicz Criterion” methods respectively.

For chromosome C_1 in Figure 4.2, the classifiers 1, 2, … , 38 and 40 are selected in the classifier ensemble and the voting method used is Minimax Regret algorithm.

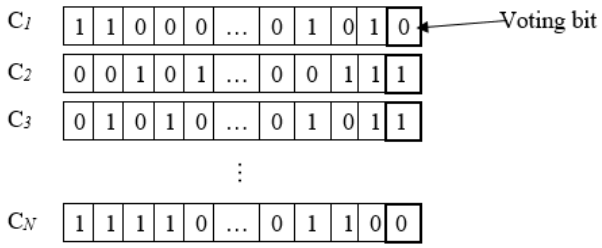

Figure 4.2. Description of population and voting bit

The population size in this study is $N = 100$ chromosomes, each represented by binary strings of length $M = 41$ including the voting bit. The number of generations for the GA evolution is set at 100. For every generation, the selection of the pair of chromosomes to partake in reproduction take place through a Tournament or Roulette Wheel selection as described in Figure 4.1. The pair of chromosomes selected is then passed through a process of crossover and mutation at a rate of 0.9 and 0.01 respectively. For a crossover, the system randomly chooses between a 1-point or 2-point crossover based on a split decision. After the crossover, the offspring are considered for mutation. The voting bit in an individual chromosome is subjected to mutation at a rate of 0.2. The fitness, which is the F-score, of each chromosome in the population is calculated by combining the classifier ensemble and the chromosomes are ranked according to their fitness. We employed elitism where 5% of the best individuals from the previous generation are propagated to the new generation as long as they are not already present in the new generation. This is to avoid the fittest chromosomes from quickly taking over the entire population.

There exist different voting methods used to compute the performance of a classifier ensemble including the simple majority, the weighted majority, percentage majority etc. However, in this thesis, we introduce the novel use of two decision-making under conditions of uncertainty techniques to calculate the fitness of chromosomes. These two voting methods are employed as they overcome the limitations of the conventional methods where decisions are made from a single opinion of either the strength, weight or percentage of the alternatives considered. They consider multiple opinions from all the alternatives considered before making a decision.

i. **Hurwicz Criterion (HC):** is a pessimistic approach suggested by Leonid Hurwicz in 1951. It selects the maximum and minimum payoff from each alternative and tries to find a middle ground between the extremes of the optimist and pessimist criteria. It also employs a measure of assigning a given percentage weight to optimism and the balance to pessimism in a bid to avoid an assumption of total optimism or pessimism. This percentage weight is called the coefficient of realism (α) and the balance is called the coefficient of pessimism (1 - α) where $0 \leq \alpha \leq 1$. In our implementation of this method, due to the pessimism about the actual outcome, we set α at 0.6, which is slightly in favour of the optimistic alternative. The subsets of classifiers are grouped into the two alternatives. The best and worst F-score from both alternatives are selected and used to calculate an HC weighted average for both alternatives Yes (A_Y) and No (A_N) as follows:

$$HC\ (A_Y) = \alpha\ (A_{Y\ max}) + (1-\alpha)\ (A_{Y\ min}) \tag{4.2}$$

$$HC\ (A_N) = \alpha\ (A_{N\ max}) + (1-\alpha)\ (A_{N\ min}) \tag{4.3}$$

The best $HC\ (A_j)$ such that HC= max ($HC\ (A_Y)$, $HC\ (A_N)$) where j signifies one of the two alternatives is chosen as the decision of the ensemble. However, in cases of a tie in the decision-making, we apply the reverse of the process to the same value of α, such that:

$$HC\ (A_j) = (1-\alpha)\ (A_{j\ max}) + \alpha\ (A_{j\ min}) \quad (4.4)$$

ii. **Minimax Regret (MR):** seeks to minimize the maximum regret and it is useful in executing a risk-neutral decision-making. The subset of classifiers are grouped into two alternatives of "Yes" and "No" and the best and worst F-scores from both alternatives are selected as $A_{j\ max}$ and $A_{j\ min}$ where j signifies one of the two alternatives. This method selects the alternative with the least opportunity loss using the formula:

$$MR = \min[\max\ ((A_{Y,N\ max}) - A_{j\ max}\ , (A_{Y,N\ min}) - A_{j\ min})] \quad (4.5)$$

where $A_{Y,N\ max}$ or $A_{Y,N\ min}$ represents the maximum or minimum from both alternatives. In this method, in cases of a tie, we break the tie by the use of a coin toss.

The major components of the CSS used in this thesis are shown in Figure 4.3. In the "Data" component, the entity mentions and extracted features are labelled and represented as token instances for sampling. In "Feature Categories", the instances used as features for training the classifiers are grouped into four (4) different sets as shown in Table 4.1. In the "Classifiers" component, the individual classifiers are trained separately using these different feature categories and their results are used during the CSS process in the "Classifier Subset Selection". In the CSS process, GA is employed and after a series of evolution and a number of generations, a solution is found as the best classifier ensemble.

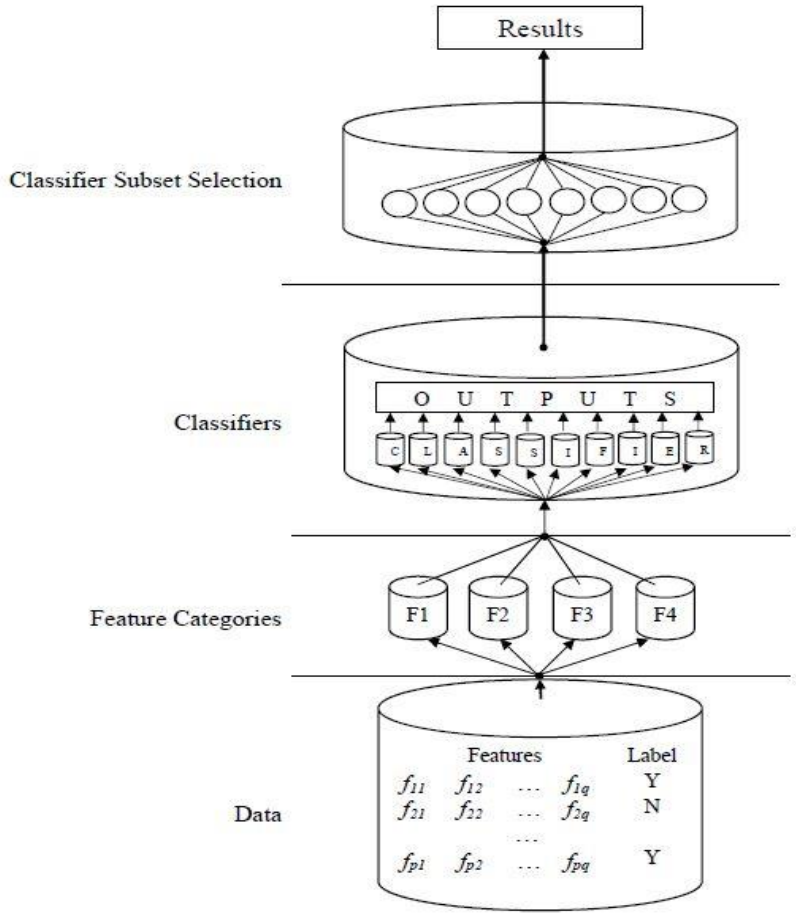

Figure 4.3. Description of the CSS components.

We performed the experiments using the 9 different settings in order to determine which of the settings generates the best classifier ensemble adaptable to the test sample in the last stage of our experiment. The initial population of chromosomes is generated randomly. However, the same set of randomly generated chromosomes is used as the initial population throughout the experiment for proper comparison on the evolution and results of the different settings employed. This initial population consists of unique chromosomes. Table 4.3 shows the settings used in this experiment.

Table 4.3. Experimental settings

S.NO	Settings	Selection Algorithm		Voting Method	
		Roulette Wheel	Tournament	Hurwicz Criterion	Minimax Regret
1	RTHM	X	X	X	X
2	RTH	X	X	X	
3	RTM	X	X		X
4	RHM	X		X	X
5	RH	X		X	
6	RM	X			X
7	THM		X	X	X
8	TH		X	X	
9	TM		X		X

The names used to describe each setting are coined using the first character of the selection algorithm and voting method employed in the setting. For example, the setting RTHM denotes the use of the roulette wheel and tournament selection algorithms along with Hurwicz Criterion and Minimax Regret voting methods. Additionally, in Settings 1-3, where the two selection algorithms are available, one of them is chosen to select all pairs of chromosomes to be considered for reproduction in each generation based on a coin toss. The main objective for using different selection methods is to create more variations in the chromosome selection process for chromosomes selected for reproduction. Additionally, in Settings 1, 4 and 7 where the two voting methods are employed for classifier combination, the choice of voting method used depends on the voting bit on the individual classifier. Since the fitness of a given chromosome can be affected by the quality of the voting method used, this seeks to achieve the best possible combination solution over time during the evolution. The classifier results obtained from the development dataset is used throughout the GA evolution process.

4.3 Results Evaluation

Table 4.4 presents the performances of the individual classifiers on the BioCreative V development dataset when the four different feature sets described in Table 4.1 are applied.

Table 4.4. Results obtained from the individual classifiers using the development set

S.NO	Classifier	Feature subsets	P (%)	R (%)	F1 (%)
1	BN Hill	A	62.25	52.47	56.94
2		B	48.84	6.23	11.05
3		C	65.12	48.52	55.61
4		D	61.39	52.47	56.58
5	BN K2	A	62.31	52.27	56.85
6		B	48.84	6.23	11.05
7		C	65.12	48.52	55.61
8		D	61.69	52.67	56.82
9	BN TAN	A	78.95	40.02	53.12
10		B	47.45	6.42	11.31
11		C	87.68	35.18	50.21
12		D	79.11	40.42	53.5
13	J48	A	72.73	50.59	59.67
14		B	54.82	26.98	36.16
15		C	83.55	37.65	51.91
16		D	71.19	49.8	58.6
17	NB	A	74.49	36.07	48.6
18		B	41.26	14.23	21.16
19		C	73.76	32.21	44.84
20		D	80.75	31.92	45.75
21	NBK	A	74.25	43.87	55.15
22		B	49.59	17.98	26.39
23		C	76.16	37.25	50.03
24		D	69.79	47.04	56.2
25	R3	A	40.92	34.98	37.72
26		B	35.81	26.68	30.58
27		C	42.37	42.79	42.58
28		D	53.95	51.98	52.95
29	R4	A	94.67	35.08	51.19
30		B	79.17	9.39	16.79
31		C	93.55	28.66	43.88
32		D	89.54	46.54	61.25
33	SVM1	A	57.67	49.8	53.45
34		B	43.18	35.38	38.89
35		C	47.66	34.19	39.82
36		D	68.84	37.55	48.59
37	SVM2	A	72.28	42	53.13
38		B	66.05	28.06	39.39
39		C	71.35	37.15	48.86
40		D	68.5	44.27	53.78

In order to evaluate the performance of the ensemble produced by GA, a classifier ensemble containing all the classifiers is combined using the HC and MR voting methods. This full classifier ensemble produced F-scores of 48.78% and 45.39% when combined using the HC and MR respectively. After 100 generations, the two fittest chromosomes produced in each of the 9 settings are presented in Table 4.5.

Table 4.5. The fittest chromosomes from the 9 settings on the development dataset

S.NO	Settings	Chromosomes	P (%)	R (%)	F1 (%)
1	RTHM	00000100010000000000000000001101000001000	78.94	54.45	64.45
		00000100010000010000000000001101000001000	68.57	59.29	63.59
2	RTH	00000100110001000100000011001101000010001	76.38	55.93	64.57
		00000100110001000100010011001101000010001	76.38	55.93	64.57
3	RTM	01000000010001000000110010001101000010000	47.99	65.02	62.58
		01000000010001000000110000001101000011000	61.99	62.06	62.58
4	RHM	00000100001000000100010001001101000011000	71.53	58.10	64.12
		00000100001000000100000001001101000011000	71.36	58.10	64.05
5	RH	01000100110001000100000010000001000010001	76.42	56.03	64.66
		01000100110001000100010011000001000011001	76.42	56.03	64.66
6	RM	01000000010010001010010000000101010011000	50.97	67.29	60.68
		01000000010010001010010000000101010001100	50.71	67.19	60.20
7	THM	01000100010101000100000001001101000010000	71.09	58.79	64.52
		01000100010101000100000001001101000010001	76.08	55.93	64.47
8	TH	01000100110001000100010010000111000011001	75.68	55.34	63.93
		01000100110001000100010010000111010011001	75.68	55.34	63.93
9	TM	01000000110001000100010001000001000010000	71.7	59.58	64.66
		01000000110001000100010001000001000011000	69.44	60.18	64.66

Table 4.5 shows that the fittest classifier ensembles (chromosomes) are produced in Setting 5 (RH) where the roulette wheel selection and HC voting method are employed and in Setting 9 (TM) where the Tournament selection and MR voting method are employed.

The fittest classifier ensembles reported in Table 4.5 are employed for validation using the joint level test dataset. The validation is performed in order to determine the performance of our evolution system and the ability of the classifier ensemble to handle generalization task on the test dataset and the results are reported in Table 4.6.

Table 4.6. Results obtained by applying the fittest classifier ensembles on the test dataset

S.No	Settings	Chromosomes	P (%)	R (%)	F1 (%)
1	RTHM	00000100010000000000000000001101000001000	88.93	48.97	63.16
		00000100010000010000000000001101000001000	73.91	57.13	64.45
2	RTH	00000100110001000100000011001101000010001	84.79	50.19	63.06
		00000100110001000100010011001101000010001	84.79	50.19	63.06
3	RTM	01000000010001000000110010001101000010000	49.03	63.7	55.41
		01000000010001000000110000001101000011000	63.05	62.1	62.57
4	RHM	00000100001000000100010001001101000011000	75.79	53.75	62.9
		00000100001000000100000001001101000011000	85.12	49.91	62.92
5	RH	01000100110001000100000010000001000010001	84.54	50.28	63.06
		01000100110001000100010011000001000011001	53.09	59.66	56.18
6	RM	01000000010010001010010000000101010011000	52.38	68.11	59.22
		01000000010010001010010000000101010001100	49.97	71.29	58.76
7	THM	01000100010101000100000001001101000010000	75.65	54.22	63.17
		01000100010101000100000001001101000010001	84.54	50.28	63.06
8	TH	01000100110001000100010010000111000011001	51.11	62.57	56.26
		01000100110001000100010010000111010011001	44.49	67.45	53.62
9	TM	01000000110001000100010001000001000010000	75.93	53.85	63.01
		01000000110001000100010001000001000011000	65.58	60.23	62.79
*	FullM	11111111111111111111111111111111111111110	31.67	84.9	46.13
*	FullH	11111111111111111111111111111111111111111	32.61	81.52	46.58

The best performing classifier ensemble was generated from the RTHM setting when the MR voting method was employed. It produced an F-score of 64.45%.The settings (*) shows that the complete base classifiers are employed and combined using the MR (FullM) and the HC (FullH) voting methods. The combinations produced F-scores of

46.13% and 46.58% for MR and HC respectively. The individual performances of the base classifiers on the test dataset are reported in Table 4.7.

Table 4.7. Results obtained from the individual classifiers using the test set

S.NO	Classifier	Feature subsets	P (%)	R (%)	F1 (%)
1	BN Hill	A	52.53	62.64	57.14
2		B	7.13	61.29	12.77
3		C	49.34	67.61	57.05
4		D	52.53	62.29	57.0
5	BN K2	A	52.25	63.22	57.21
6		B	7.13	61.29	12.77
7		C	49.44	67.48	57.07
8		D	52.44	62.67	57.1
9	BN TAN	A	40.9	79.85	54.09
10		B	7.6	57.86	13.44
11		C	36.21	90.4	51.71
12		D	41.18	80.11	54.4
13	J48	A	52.25	76.83	62.2
14		B	28.71	61.32	39.11
15		C	39.02	86.85	53.85
16		D	53.38	75.17	62.43
17	NB	A	38.74	79.73	52.14
18		B	14.92	47.46	22.7
19		C	34.24	77.49	47.49
20		D	35.37	82.14	49.45
21	NBK	A	46.06	76.72	57.56
22		B	23.83	61.8	34.4
23		C	39.12	81.45	52.85
24		D	47.37	70.73	56.74
25	R3	A	39.02	42.15	40.52
26		B	25.89	33.99	29.39
27		C	44.28	42.29	43.26
28		D	52.44	51.71	52.07
29	R4	A	35.74	97.69	52.33
30		B	10.6	81.88	18.77
31		C	30.68	95.61	46.45
32		D	48.41	88.81	62.66
33	SVM1	A	50.47	56.87	53.48
34		B	39.4	46.56	42.68
35		C	39.49	51.28	44.62
36		D	38.27	71.2	49.78
37	SVM2	A	9.38	38.31	15.07
38		B	0.38	100	0.76
39		C	39.59	71.77	51.03
40		D	10.23	23.8	14.31

4.4 Analysis and Discussion

4.4.1 Analysis

The classifier ensembles in Table 4.6 reveals that they compose of different classifier algorithms trained on different feature sets. For example, the second classifier ensemble "000001000100000100000000000011010000001000" from RTHM with MR employed as the voting method shows that the classifiers are selected from three different feature sets which are Set A, B and D and comprises of BN K2, BN TAN, R4 and SVM2 classifier algorithms. The individual classifiers in the ensemble produced average performances on the test dataset in terms of recall and F-score as reported in Table 4.7. However, by employing the classifier combination method, this classifier ensemble produced the best F-score of 64.45% on the test dataset as shown in Table 4.6. The improved performance of the classifier ensembles from the average performances of the individual classifiers is due to the voting methods employed. These voting methods help to improve the complementarity in the classifiers and maximizes the strengths of the best performing classifiers in the ensemble. Furthermore, unlike the conventional methods, these voting methods handle the diversity of classifiers in an ensemble better in order to make a more accurate decision.

Although these voting methods show good decision-making ability and efficiency in classifier combination, they also have some drawbacks. Consider the first classifier ensemble from setting RTH presented in Table 4.6, the chromosome "000001001100010001000000110011010000010001", shows that the classifiers are selected from feature sets A, B and D and comprises of 7 different classifiers (BN K2, BN TAN, J48, NB, R3, R4 and SVM2). The ensemble, when applied to the test dataset, is used to discuss the limitations of combining the two voting methods employed in

our approach. This classifier ensemble is a collection of classifiers 6, 9, 10, 14, 18, 25, 26, 29, 30, 32 and 37 reported in Table 4.7.

From the test dataset, consider the abstract excerpts from the documents with PubMed ID: 23433219 and 24100257 respectively.

Excerpt 1: "…for **methamphetamine**-induced psychosis and other Axis I **psychiatric disorders**."
Excerpt 2: "…Extensive literature search revealed multiple cases of **coronary artery vasospasm** secondary to **zolmitriptan**."

The classifier ensemble aims at deciding whether there are CID relations between the entity mentions in bold or not. In excerpt 1 there exist no true CID relation between the chemical "methamphetamine" and the disease "psychiatric disorders" mentions with concept identifier D008694 and D001523 respectively. However, based on the HC and MR methods, the decision to this relation instance differ between the two methods. This scenario is explained by applying Equations (4.2), (4.3), and (4.5). From the classifier ensemble, only classifier 25 with an F-score of 37.72% predicted "Yes", while the others predicted "No". The best and worst F-scores of the classifiers that predicted "No" are 61.25% and 11.05% respectively. Applying Equations (4.2) and (4.3) for the HC method where $\alpha = 0.6$ gives;

$$H_{Yes} = 0.6\ (37.72) + 0.4\ (37.72) = 37.72$$

$$H_{No} = 0.6\ (61.25) + 0.4\ (11.05) = 41.17$$

Since H_{No} is better than H_{Yes}, the decision from *HC* is "No".

However, when applying Equations (4.5), the MR method gives:

MR_Y = max [max (37.72, 61.25) – 37.72), (max (37.72, 11.05) – 37.72)] = 23.53

MR_N = max [max (37.72, 61.25) – 61.25), (max (37.72, 11.05) – 11.05)] = 26.67

MR = min (MR_Y, MR_N) = 23.53 = MR_Y

The decision from *MR* is "Yes" since MR_Y gives a lesser opportunity cost compared to MR_N.

Furthermore, in excerpt 2, there exist a true CID relation exists between the chemical "zolmitriptan" and the disease "coronary artery vasospasm" mentions with concept identifier C089750 and D003329 respectively. Only classifier 14 predicted "Yes" to these entities having a relationship. When calculating the decisions of the HC method using Equations (4.2), (4.3) and the MR method using Equation (4.5), HC predicted "No", while MR predicted "Yes". These examples show the limitations of HC handling the scenarios where an alternative has only a single option (classifier) and MR handling the scenarios where an alternative produces both the best and the worst performances. We handled these limitations during the evolution by allowing the classifiers chosen in a chromosome to be combined using one of the two voting methods over the course of the evolution. The voting bit on the chromosome is mutated with a probability of 0.2 and this helps to improve the performance of the ensembles generated during the evolution.

4.4.2 Comparison of Results

In Table 4.8, we compare the best performing ensemble produced by our GA framework with those of other state-of-the-art systems that used the BioCreative V corpus test dataset. Zheng et al. [9] used the Convolutional Neural Network (CNN) and integrated the Long-Short Term Memory Units (LSTM) to extract high-level semantic relation representations between the chemical and disease mention and achieved an F-score of 54.30%. Zhou et al. [112] used the shortest dependency path

tree to capture the most direct syntactic and semantic relationship between chemical and disease and achieved an F-score of 55.05%. Alam et al. [23] extracted features from the CTD [113] together with other linguistic features. Their system produced an F-score of 56.60%. Xu et al. [114] used numerous drug-side-effect resources in order to extract KB features like the ngram word features used to train SVM classifier. During the training of the classifier, they made good use of relation labels of chemical and disease pairs present in CTD, which is the main source for the corpus generation. They achieved an F-score of 57.03%.

Table 4.8. Performance comparison with other systems

System	**Description**	**F-score (%)**
Zheng et al. [9]	CNN + LSTM	54.30
Zhou et al. [112]	Tree kernel, Three parsers	55.05
Alam et al. [23]	Knowledge approach	56.60
Xu et al. [114]	SVM + KB features	57.03
Gu et al. [8]	CNN + ME	60.20
Lowe et al. [115]	Heuristic rules	60.80
Peng et al. [10]	SVM + KB	63.10
Our system	MCS using GA	64.45

Gu et al. [8] used an ML-based system that also used the CNN and linguistic features to extract CID relations with Maximum Entropy (ME) models and achieved an F-score of 60.20%. Lowe et al. [115] achieved good results (an F-score of 60.80%), but the computational cost, as well as the huge amount of time their system requires for this task, makes their system limited. Peng et al. [10] used a set of linguistic knowledge and statistic features to achieve an F-score of 63.1%. However, they trained their system with an additional 500 BioCreative development dataset and 18,410 CTD-Pfizer documents from [64] to improve the performance of their system to 71.83%.

4.5 Conclusion

In this section, we highlighted the efficient application of a novel approach for classifiers combination in our MCS framework. This was the application of two decision-making under uncertainty techniques as voting methods (Minimax Regret and Hurwicz Criterion) in order to overcome the major drawbacks associated with the implementation of the conventional classifier combination methods in GA. The implementation of these voting algorithms for classifier combination in our system produced good results that are comparable to the current state-of-the-art systems in CID relation extraction. These two techniques consider multiple opinions about every alternative before carefully making a decision, unlike the conventional ones that normally make decisions from a single opinion of either the strength, weight or percentage of the alternatives. Due to the critical nature of the decisions to be made, their decision-making is pessimistic as they try to avoid making costly decisions at every point. These methods also showed that the literature requirement for the individual classifiers considered in the MCSs to be well performing could be enhanced by increasing the diversity and the complementarity of the classifiers to make up the MCS, however, without importantly sacrificing efficiency and result.

Despite the success of this approach, there is a need for further improving the system. For instance, this can be achieved by increasing the pool of classifiers to determine the effect a larger pool can have and also, to apply a control function to the voting methods in other to help them overcome their limitations and to further improve their performances.

Chapter 5

RELSCAN^{+}: IMPROVING CHEMICAL DISEASE RELATION EXTRACTION THROUGH THE COMBINATION OF MULTIPLE MENTION LEVELS

In this chapter, we introduce a two-classifier ML-based relation extraction architecture using the same setup described in the previous chapters. In this architecture, the relation instances on all three-sentence levels, that are referred to as 'Intra-sentence level', 'Inter-sentence level' and 'Joint level', are employed. The 'Intra-sentence level' refers to the mention of the chemical and disease entities in the same sentence, the 'Inter-sentence level' refers to the mention of the chemical and disease entities in neighbouring spanning sentences, while the 'Joint level' is the combination of the intra- and inter-sentence levels. The features used in this system are the three feature categories discussed in Section 3.5.

The biomedical relation extraction system discussed in this chapter is made of a three-phase architecture. In Phase 1, the input sentence undergoes text processing and then the construction of relation instances at the intra- and inter-sentence levels, which are subsequently merged to form the joint level as discussed in Section 3.4. In Phase 2, features are specifically extracted for each relation instance at the three levels. At each of these levels, three classifier models that consist of the combination of two ML classifiers, SVM and J48 decision tree were trained using the training dataset and then applied on the test dataset to classify the CID relation instances. Phase 3 consists of

two steps; in Step 1, the classifier outputs from both the intra- and inter-sentence levels are merged and in Step 2, the results from Step 1 are combined with the results from the classifier trained at joint level using a prediction probability-based voting algorithm to determine the final result. Using the BioCreative V corpus for validation, we obtained an F-score of 64.2% and 65.32% for the development and test dataset respectively. Based on the validation on test dataset, this leads to a 0.87% increase from the 64.45% F-score of the system in Chapter 4.

5.1 Background

Relation extraction in other domains [12, 17] as well as in the CID task [10, 23, 98, 116] is normally treated as a classification problem and most of the proposed approaches employed ML methods. During the CDR BioCreative V challenge, the participating teams employed several ML techniques such as logistic regression [116], maximum entropy [6, 117], Support Vector Machine (SVM) [66], LIBSVM [114] and Naïve Bayes [4].

The CID task has been generally expressed as a binary classification task that predicts the presence of an induction relation between a chemical and disease pair in an article [6, 7, 8, 10, 118]. Although the BioCreative V corpus provides CID relations only at the document level, some CID systems limit their CID relation extraction tasks to intra-sentence level [115, 119]. However, this leads to the loss of some inter-sentence level CID relations as they account for one-third of the total CID relations present in the corpus [63]. This has motivated some systems to perform relation extraction on a document-level in order to extract the CID relations on both the intra- and inter-sentence levels [9, 10, 98]. Furthermore, some systems perform the CID relation

extraction separately on both levels and then merge the results to get the full CID relation [6, 7, 8, 9, 120].

The reported system in this chapter utilizes the constructed candidate relation instances on the intra- and inter-sentence level datasets in two ways. Firstly, they are individually employed during classification and the outputs of the classifiers from the two levels are then combined to generate the complete relation instances and secondly, they are combined to form the joint level dataset before classification. These processes are discussed in details in Section 5.2.

5.2 Method

5.2.1 RelSCAN^{+}

The architecture of this relation extraction system consists of three phases. In phase 1 as shown in Figure 5.1, the text processing module transforms the input data into sentences. This is followed by the construction of candidate relation instances using the predefined entities in the input data. In Phase 2, features are extracted for all candidate relation instances. Subsequently, a label (YES or NO) is added to each candidate relation instance indicating the existence of a true relationship between the two entities paired according to the gold standard data. In Phase 3, in order to obtain the final CID predictions from the relSCAN^{+}, we merge the outputs from the intra- and inter-sentence levels and then combine them with the outputs from the joint level obtained in Phase 2 by using a voting algorithm.

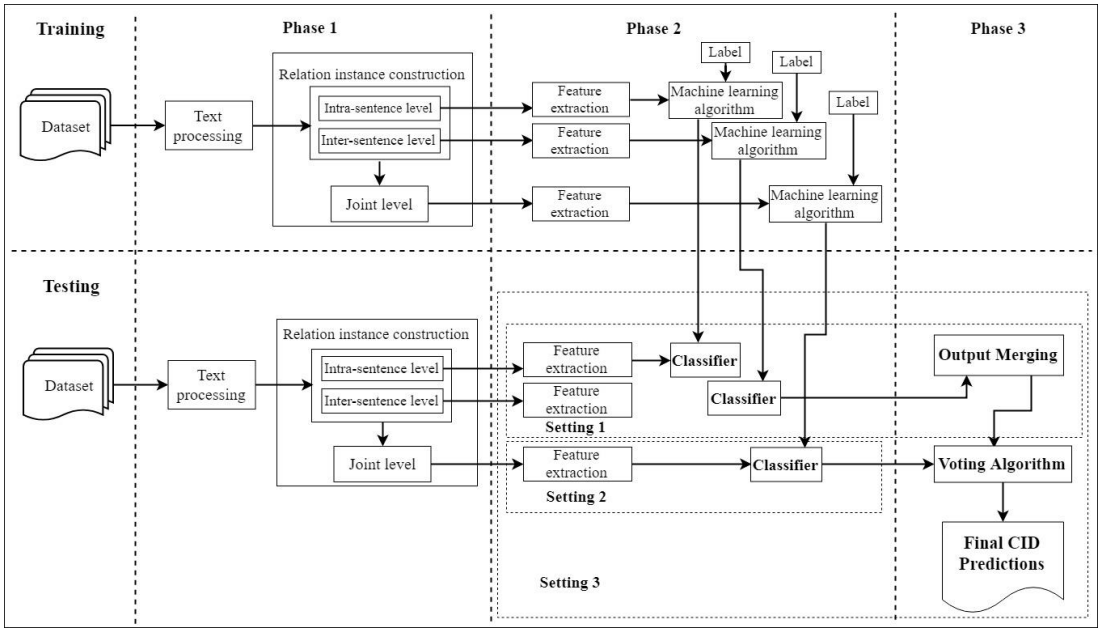

Figure 5.1. RelSCAN^{+} architecture

In testing stage, Setting 1 shows the architecture used for presenting results from merging the outputs from the intra- and inter-sentence levels; Setting 2 shows the architecture used for presenting the results of the joint level and Setting 3 is the section of the architecture that combines results from both Settings 1 and 2 using a voting algorithm.

5.2.1.1 Phase 1

The input to Phase 1 is documents, each consisting of only a title and an abstract. In the text processing step, we segment the abstracts and titles into sentences and then replace all entity mentions with placeholders. After the text processing stage, the relation instances are constructed at two different mention levels, which are the intra- and inter-sentence levels. During the construction of the relation instances, the number of sentences used to generate a given relation instance is different at the two sentence levels. For constructing relation instances at the intra-sentence level, only a single sentence that contains the two entity mentions is used. However, at the inter-sentence level, since the entity mentions may span multiple sentences, two to three neighbouring

sentences are used to generate a given relation instance. These multiple sentences used to generate a relation instance in the inter-sentence level are then combined to form a composite sentence. Figure 5.2 presents an example to illustrate the construction of the relation instances across the intra-sentence, inter-sentence and joint level datasets.

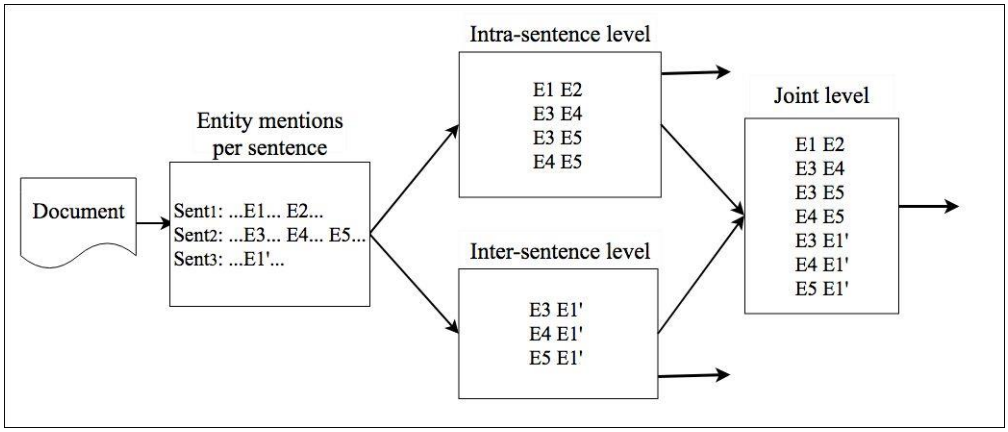

Figure 5.2. An illustration of the construction of candidate relation instances at different levels.

As represented in Figure 5.2, Sent1, Sent2 and Sent3 denote the sentences extracted from a document. E1, E2… denote the entity mentions present in a given sentence and E1` denotes the appearance of the entity E1 in another sentence Sent3, in the modules of the Intra- and Inter-sentence levels. The paired entities are the candidate relation instances extracted at those levels. Finally, the combination of these two sentence levels produces the joint level. Thus, we create three datasets where the candidate relation instances at the intra- and inter-sentence levels are non-overlapping sets and the joint level, which is the union of the first two datasets. The pair of chemical and disease mentions is unordered, indicating that their order of appearance in the text does not affect the possibility of a CID relation between them. The three different datasets are individually passed to Phase 2. The details on the text processing, construction of

the candidate relation instances and feature extraction have been discussed in Chapter 3.

5.2.1.2 Phase 2

Each of the three datasets (intra-sentence, inter-sentence and joint levels) that consist of the candidate relation instances and their extracted feature sets are employed for training using a combination of two different ML algorithms namely SVM and J48 decision tree. Thus, three classifier models are formed as shown in Figure 5.1. The robustness of the learning method and computational efficiency can be improved if the features extracted per relation instance are distinguishing enough [12]. Therefore, even though the feature types used for all three mention levels are exactly the same, the features extracted per relation instance are particular to the individual entity and the collective relation information of both entities present in the sentences considered. In the inter-sentence case, the composite sentences as discussed in Section 3.4 are used for feature extraction in the same manner as the single sentences in the intra-sentence level. Details of the ML algorithms or base classifiers and features employed in this system are discussed in Section 3.2 and 3.5 respectively.

5.2.1.3 Phase 3

Firstly, the outputs of the classifiers from the intra- and inter-sentence levels are merged to form the dataset that has the same set of candidate relations as the joint level dataset. Since there are no overlapping of the candidate relation instances in both mention levels, this operation is used to produce the complete relation instances in the dataset. Aside from the merging of the outputs from both sentence levels, no post-processing or filtering of the results is performed. The results obtained after the merging process are then combined with the results from the joint level by using a voting algorithm to produce the final CID prediction of our system. Figure 5.3 presents

a graphical description of the processes involved in Phase 3 using the same example given in Figure 5.2.

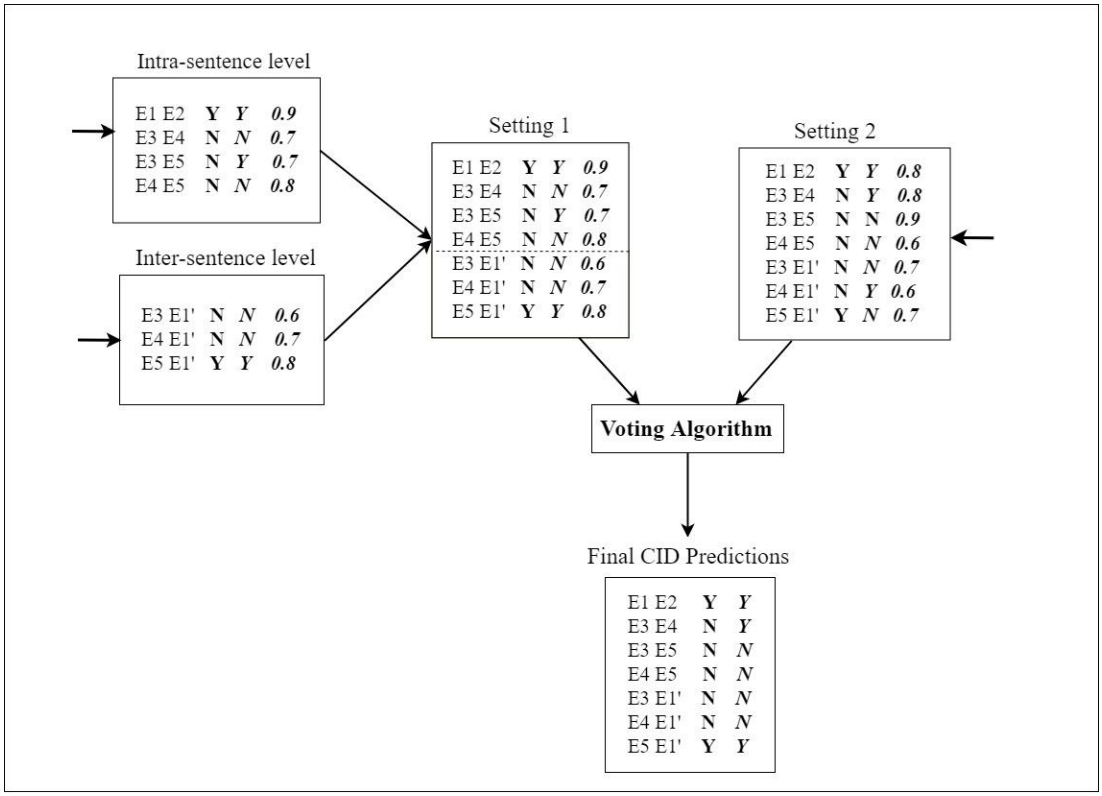

Figure 5.3. Generation of the final CID predictions

In Figure 5.3, the documents labelled Intra-sentence level, Inter-sentence level and Setting 2 are the classifier outputs of the three classifiers as shown in Figure 5.1. In all documents given in Figure 5.3, the third columns represent the actual labels that show the presence or absence of a true CID relation between the candidate relation instances and the fourth columns represent the classification prediction labels. For the Intra-sentence level, Inter-sentence level, Setting 1 and Setting 2 documents, the additional fifth column shows the prediction probabilities of the classifiers employed. The Final CID predictions document is formed by the voting algorithm that combines the outputs from Settings 1 and 2 using the prediction probability per relation instance.

5.2.1.3.1 Voting algorithm

The voting algorithm used in this system is a type of decision-making technique, which is based on the prediction probability generated from a classification output. The voting algorithm considers every instance from the output separately and it uses a simple but effective approach for finding the maximum prediction probability based on the confidence of the decision made for each instance between the two settings. During the combination of the results from both settings, the voting algorithm is only applied when the classification predictions for a given relation instance from both settings are different. After the combination process, the final set of CID predictions by relSCAN^{+} is generated and evaluated. Algorithm 1 describes the voting algorithm process during the combination of Settings 1 and 2.

Algorithm 1. Algorithm for the voting algorithm

P_{Set1}, P_{Set2}: predictions of Settings 1 and 2 respectively
Pr_{Set1}, Pr_{Set2}: prediction probabilities of Settings 1 and 2 respectively
$F1_{Set1}$, $F1_{Set2}$: F-scores of Settings 1 and 2 respectively
Dp: the prediction decision of the system
N: the number of candidate relation instances.

For each relation instance *k=1* to *N*
  If $P_{Set1}(k) == P_{Set2}(k)$
    $Dp(k) = P_{Set1}(k)$
  Else
    If $Pr_{Set1}(k) >= Pr_{Set2}(k)$
      If $Pr_{Set1}(k) > Pr_{Set2}(k)$
        $Dp(k) = P_{Set1}(k)$
      Else
        If $F1_{Set1}(k) > F1_{Set2}(k)$
          $Dp(k) = P_{Set1}(k)$
        Else
          $Dp(k) = P_{Set2}(k)$
        End if
      End if
    Else
      $Dp(k) = P_{Set2}(k)$
    End if
  End if
End for

5.2.1.4 Classifiers Used

The two ML classifiers (SVM and J48) used in this system have been previously discussed in Section 3.2, however, their parameter settings and implementation as to this system are reported in this section. The implementations of these classifiers in the Waikato Environment for Knowledge Analysis (WEKA[2]) toolset is employed for training the classifiers. We combine both the SVM and J48 decision tree classifiers by using the 'class vote' option (weka.classifiers.meta.Vote.classifiers) and the 'average of probabilities' combination rule. The SVM classifier is used with default polynomial kernel and the complexity parameter C is tuned to 0.6 by using CVParameterSelection function. The J48 is used in its default settings with a confidence factor of 0.25, batch size of 100 and the minimum number of instances per leaf set at 2.

5.3 Results and Discussion

5.3.1 Results

The numbers of candidate relation instances in the three datasets have been previously presented in Table 3.2. This table shows a similar distribution of the positive and the negative instances across the datasets. The positive instances are the entity pairs annotated by the corpus as having a CID relation, whereas the negative instances are the entity pairs not annotated as such.

The system was trained on the training dataset and evaluated on the development and test datasets of the BioCreative V corpus. For the performances of the ML algorithms when they are used separately and combined as Setting 1 on the intra- and inter-sentence levels on both datasets are shown in Table 5.1. Note that the set of relation

[2] WEKA: https://www.cs.waikato.ac.nz/~ml/weka/

instances in the intra- and inter-sentence level datasets are non-overlapping. Furthermore, their combination contains the complete set of relations in the dataset.

Table 5.1. Results for Setting 1 on the development and test datasets

Classifier	**Dataset**	**Development**			**Test**		
		P	**R**	**F1**	**P**	**R**	**F1**
SVM	Intra-sentence level	51.5	43.8	47.5	47.1	41.8	44.3
	Inter-sentence level	80.7	80.3	80.5	90.5	79.0	84.4
	Intra + Inter sentence levels	59.4	52.6	55.8	63.3	56.5	59.7
J48	Intra-sentence level	64.4	41.1	50.2	61.9	39.5	48.3
	Inter-sentence level	100	93.4	96.6	100	92.1	95.9
	Intra + Inter sentence levels	75.7	53.8	62.9	76.2	55.1	63.9
SVM + J48 (Setting 1)	Intra-sentence level	66.2	42.1	51.4	63.7	41.3	50.1
	Inter-sentence level	97.5	95.1	96.3	98.6	92.7	95.6
	Intra + Inter sentence levels	**76.5**	**54.8**	**63.9**	**76.9**	**56.5**	**65.1**

The results reported in Table 5.1 show that at the intra-sentence level SVM produce a better recall compared to the J48 decision tree on both the development and test dataset. However, J48 produced a better recall on the inter-sentence level and a better precision on the sentence levels individually and when they are combined. The J48 decision tree in general outperformed SVM on both the development and test datasets, however, their combination produced an improved performance compared to when they are used individually.

Additionally, on both datasets, the results obtained on the inter-sentence levels highly outperform those on the intra-sentence levels. Some systems that performed the CID relation extraction task on both the intra- and inter-sentence levels [7, 8, 118, 121]

have reported the performance of the intra-sentence level to greatly outperform that of the inter-sentence level. Gu et al. attributed this to the complex structure of the sentences on the inter-sentence level limiting the extraction of traditional features [8]. In this system, we developed an approach that handles the sentences on the inter-sentence level properly, with exceptional performance on this level. However, on the intra-sentence level, the system did not produce the same performance. One of the reasons for this is that the system is able to extract more productive features on the inter-sentence level as compared to the intra-sentence level thereby producing a better classification result for the inter-sentence level. Another reason for this is attributed to the limited number of CID relation instances that span over multiple sentences as shown in Table 3.2, which leads to a smaller sample size in the inter-sentence level that in turn reduces the chances of overfitting and overgeneralization that may lead to errors during classification. The result for Setting 1 is obtained from evaluating the result produced after merging the classifier outputs from the intra- and inter-sentence levels.

Table 5.2 presents the performances of the SVM and J48 classifiers individually and when they are combined in the classifier model to generate the results for Setting 2 on both the development and train datasets. As in Setting 1, the J48 decision tree outperforms SVM on both datasets, and the performance when they are combined is better than their individual performances.

Table 5.2. Results for Setting 2 on the development and test datasets

Classifier	**Dataset**	**Development**			**Test**		
		P	**R**	**F1**	**P**	**R**	**F1**
SVM	Joint level	57.7	49.8	53.4	56.9	50.5	53.5
J48	Joint level	72.7	50.6	59.7	76.8	52.3	62.2
SVM + J48 (Setting 2)	**Joint level**	**72.7**	**52.1**	**60.7**	**74.0**	**55.2**	**63.2**

The performance of Setting 3 (relSCAN^{+}) for the development dataset when Settings 1 and 2 are combined is presented in Table 5.3. The performance in Setting 1 is better than Setting 2, however, relSCAN^{+}, which combines both Settings 1 and 2 improves the precision and the F-score despite the fact that it produces a slight decrease in the recall as reported in Table 5.3.

Table 5.3. Results from relSCAN^{+} on the development dataset

Architecture	**TP**	**FP**	**FN**	**P**	**R**	**F1**
Setting 1	555	171	457	76.5	54.8	63.9
Setting 2	527	198	485	72.7	52.1	60.7
relSCAN+	**546**	**143**	**466**	**79.25**	**53.95**	**64.2**

In Table 5.4, the performance of relSCAN^{+} on the test dataset is presented. Based on the reported results, Setting 1 outperforms Setting 2. RelSCAN^{+} causes a slight decrease of 0.31% in the recall, however, it produces a decrease in the number of FP by 6.63% and improves the precision and F-score by 1.09% and 0.22% respectively.

Table 5.4. Results from relSCAN^{+} on the test dataset

Architecture	**TP**	**FP**	**FN**	**P**	**R**	**F1**
Setting 1	602	207	464	76.9	56.5	65.1
Setting 2	588	207	478	74.0	55.2	63.2
relSCAN+	**599**	**169**	**467**	**77.99**	**56.19**	**65.32**

In combining Settings 1 and 2 only 5.87% of the total relation instances utilized the voting algorithm and for these cases, the decision was made by Setting 1 52.36% of the times and in general, a correct decision is made 59.80% of the times.

5.3.2 Discussion

5.3.2.1 Impacts of features

Table 5.5 compares the effects of the three different feature categories for both Settings 1 and 2 on the BioCreative V test dataset. In order to determine the impacts of the different feature categories, we applied different sets of the feature categories in turns and retrained the model. In both settings, the first row shows that when all the three feature categories are employed, the best performances were achieved at 65.1% and 63.1% F-scores for Settings 1 and 2 respectively.

Table 5.5. Impacts of the features on the test dataset

Feature Sets			**Setting 1**		**Setting 2**	
Contextual	**Dependency**	**Statistical**	**F1 (%)**	**F1 change (%)**	**F1 (%)**	**F1 change (%)**
X	X	X	65.1	-	63.2	-
X	X		42.3	-22.8	42.4	-20.8
X		X	57.6	-7.5	53.6	-9.6
	X	X	64.6	-0.5	62.8	-0.4
X			23.0	-42.1	17.9	-45.3
	X		31.7	-33.4	31.4	-31.8
		X	56.1	-9.0	48.5	-14.7

Table 5.5 shows that the effects of the feature categories for both Settings 1 and 2 have identical patterns. In both settings, the most significant drop in performance (Setting 1: -42.1% and Setting 2: -45.3%) occurred when the only the contextual features are used. The second biggest drop in performance (Setting 1: -22.8% and Setting 2: -20.8%) occurred when the statistical features were removed. When the three feature categories were used individually, the statistical features produced the least drop in performance (Setting 1: -9.0% and Setting 2: -14.7%). Additionally, the statistical features proved to complement the other two feature categories better as its combination with the contextual (Setting 1: -7.5% and Setting 2: -9.6%) and

dependency (Setting 1: -0.5% and Setting 2: -0.4%) features produced the two least drop in performances. This shows that although the statistical features produced the smallest decrease in performance when the feature categories were used individually, the classification performances at both settings are improved when it is combined with any other feature categories. Nonetheless, the combination of all feature categories produced the best classification performances in both settings.

5.3.2.2 Comparison with other systems

A comparison of relSCAN^{+} with the other state-of-the-art systems on the BioCreative V test dataset can be seen in Table 5.6. All the systems reported are evaluated using the gold standard annotated entities.

Table 5.6. Comparison with related work

Systems	**P (%)**	**R (%)**	**F1 (%)**
Xu et al. [120]	60.86	53.10	56.71
Panyam et al. [118]	53.2	69.7	60.3
Zhou et al. [119]	55.56	68.39	61.31
Zhou et al. [121]	60.19	58.16	61.35
relSCAN^{+}	**77.99**	**56.19**	**65.32**

Xu et al. [120] performed the CID relation extraction task by employing a CRF-based named entity recognition approach for biological entity names into their ML-based system. Their system produced an F-score of 56.71%. However, in order to improve the performance of their system, they extracted extra domain knowledge features from the knowledge-based biomedical database CTD [64]. This enhanced their system's performance by producing an improved F-score of 67.16%. RelSCAN^{+} does not utilize any external knowledge; however, it produced results comparable to [120] after their system applied the external information. Panyam et al. [118], utilized the all path graph

(APG) kernel which has the ability to work with arbitrary graph structures to attain an F-score of 65.1% for the intra-sentence level, 45.7% for the inter-sentence level and 60.3% for the full CID relation extraction task. Compared to [118], our system utilized more extensive feature categories hence why it vastly outperforms theirs. Zhou et al. [119], performed their CID relation extraction task on only the intra-sentence level where they integrated three models: feature-based, kernel-based and neural network models into their system. These models were combined to form a uniform framework that produced an F-score of 61.31%. Unlike the system proposed by Onye et al [119], relSCAN^{+} utilized a voting algorithm in its feature-based and classifier ensemble system but achieves a better result of 65.32% F-score. Zhou et al. [121], performed the CID relation extraction task on both the intra- and inter-sentence levels. Their system utilized the CNN model, which employed a sequence-based and a dependency-based model at the intra-sentence level and just a sequence-based model at the inter-sentence level. The results of these models are merged to produce an F-score of 59.16%. Their system further applied some post-processing rules on the merged results to achieve an F-score of 61.35%. Compared to [121], the worst performing component (Setting 2) of relSCAN^{+}, where we merged the two-sentence levels before classification without any post-processing produced a better result than theirs.

The main findings from relSCAN^{+} can be summarized as:

i. The inter-sentence level substantially outperforms the intra-sentence level on the CID relation extraction task,

ii. The combination of the candidate relation instances from both the intra- and inter-sentence levels after classification produced a better performance compared to when they are combined before classification (Setting 1: 65.1% vs Setting 2: 63.2% F-scores), however,

iii. The use of a maximum prediction probability-based voting algorithm to combine the results from Settings 1 and 2 further improved the performance of relSCAN^{+} on the CID relation extraction task from 65.1% F-score for Setting 1 to 65.32% F-score,

iv. To the best of our knowledge, RelSCAN^{+} outperforms all CID relation extraction architectures, which do not utilize additional resources aside from the corpus itself.

5.3.2.3 Error analysis

We performed error analysis to detect the reasons for the FN and FP in the results from relSCAN^{+} on the test dataset as shown in Table 5.4.

i. **Incorrect classification in Setting 1:** The majority of the false classifications occurs at the intra-sentence level producing 95% and 98% of the total FN and FP respectively. This may be attributed to the extractable information at both levels. At the intra-sentence level, features are extracted from a single sentence, which limited the extraction of sufficient informative and distinct features whereas at the inter-sentence level two to three sentences could be employed, thereby increasing the amount of informative and distinct features available for extraction.

ii. **Incorrect classification in Setting 2:** In the joint level, the number of FN and FP increased compared to Setting 1 by 14.36% and 3.02% respectively which resulted in a drop in the system's recall. This may be due to the increase in the complicated structure of the relation instances from the two different sentence levels degrading generalization performance of the classifier used in the system.

iii. **Voting algorithm misclassification:** The number of FN and FP detected during the tiebreaking constituted 11.13% and 39.64% of the total FN and FP respectively detected when combining Settings 1 and 2. The reason for this is mainly due to the limitation of the voting algorithm employed as its decision-making ability is simply based on identifying and selecting the maximum prediction probability between the two settings.

5.4 Conclusion

This system implements an ML-based classifier ensemble system that automatically extracts CID relations from three mention levels: intra-sentence, inter-sentence and joint levels. This study shows that the combination of the inter- and intra-sentence level relations after classification (Setting 1) produces a better performance compared to when they are combined before classification (Setting 2: joint level). In relSCAN^{+}, in order to determine the final CID predictions, we merged the outputs of the two settings using a maximum prediction probability-based voting algorithm. Thus, the precision and F-score were improved compared to the better results achieved in Setting 1.

RelSCAN^{+} does not utilize any external data and relies on features extracted solely from the given dataset. The evaluation benchmark on the BioCreative V corpus has shown that relSCAN^{+} performs better than the current systems, which do not require any additional knowledge from outside sources during the CID relation extraction.

Despite the success of relSCAN^{+}, it can still be improved. Firstly, we aim to find a balance in which we can develop an improved set of features that would be more suited to the intra-sentence case whilst not weakening the performances of the inter-sentence

case and the overall system. Secondly, we aim to utilize a more adaptive and flexible decision-making voting algorithm that is not limited to prediction probability but has the ability to compare multiple variables per relation instance in both Settings 1 and 2 during the combination process.

Chapter 6

CONCLUSION AND FUTURE WORK

This thesis is based on the observation that relations between chemicals and diseases may be described using one sentence that mentions both entities, a disease and a chemical, explicitly or in some cases two or more neighbouring sentences that mention the disease and/or chemical and the challenges of extracting these relation mentions across multiple sentence levels. Given the task of extracting CID relations from abstracts, all candidate relations mentioned in a single sentence (intra-sentence level) or in multiple neighbouring sentences (inter-sentence level) must be considered since both levels are expected to contain more informative and in many cases distinctive information. The final decision of relation extraction should be based on both sentence levels. This thesis implements two ML systems based on novel approaches for the CID relation extraction task. This chapter contains a summary of the contributions of the thesis and the directions for future work.

6.1 Thesis Contributions

The main contributions of this thesis are the two novel relation extraction approaches implemented to extract and predict CID relation instances across multiple mention levels and the improvement to the relation extraction tasks in terms of performance. The contributions are discussed in more details through the following steps:

1. We described a GA optimization system for biomedical relation extraction, which uses a novel approach of employing two decision-making under uncertainty techniques for MCS. The two decision-making under uncertainty

techniques (Minimax Regret and Hurwicz Criterion) are employed in our system to overcome the limitations of the conventional classifier selection techniques. In GA, the development of a base classifier traditionally requires the use of only high performing individual classifiers as one way to guarantee the generation of classifier ensembles through the evolution process that a better performance than the ensemble with all the base classifiers or the best performing individual classifier. In contrast, our system implemented the use of both high and average performing classifiers in the base classifier. Additionally, in order to introduce more variations in the evolution process with an aim of avoiding the local optimum, the system utilized the random selection of a classifier selection technique per generation and a type of crossover per chromosome. The selection techniques implemented are the Roulette Wheel and Tournament selections, and the types of crossover used are the 1-point or 2-point crossover. Mutation is performed firstly after crossover, where the chromosomes are considered for mutation at a rate of 0.01 and then on the voting bit which is considered for mutation at a rate of 0.2.

2. We implemented an ML-based system that uses a voting algorithm to predict relations across multiple sentence mention levels. In relSCAN^{+}, the candidate relation instances over three sentences mention levels (intra-sentence, inter-sentence and joint levels) are used to predict the final CID relations. This is an improvement from [122]. In this system, relSCAN^{+}, the relation instances are firstly created on both the intra- and inter-sentence levels, then they are merged to create the joint level. In the past, CID relation extraction has been performed on either the intra-sentence or inter-sentence levels or both, however, to the best of our knowledge, no system has applied the CID relation extraction task

using the intra- and inter-sentence mention levels in multiple combinations. In relSCAN^{+}, the relation instances on the intra- and inter-sentence levels are classified individually, then their outputs are merged (reported as Setting 1 in Chapter 5) [122], and then, the relation instances on the intra- and inter-sentence levels are merged (joint level) before classification (Setting 2).

Although both settings reported impressive results, in order to generate the final CID prediction of the system, relSCAN^{+}, a maximum prediction probability-based voting algorithm is employed to combine the two outputs from the two settings. The implementation of the voting algorithm was efficient in improving the system [122] and it outperforms all CID relation extraction architectures, which do not utilize additional knowledge.

6.2 Future Work

In addition to the two relation extraction methods presented in this thesis, our study opens up several opportunities for future work.

1. Incorporation of external knowledge dictionary into our system. The performance of our system, relSCAN^{+}, on the intra-sentence compared to the outstanding performance on the inter-sentence level shows that more work needs to be done on this level to improve the performance. We aim to implement a form of transfer learning by incorporating an external database with prior knowledge about chemicals and diseases such as [123] for chemical concept identification and Peregrine [124] for disease concept identification. The external database is aimed at improving the level of the extracted features from the intra-sentence level and not to extract features particular to this level. This is because we employed the same set of features

at both levels, as we had to combine the different sentence levels to form the joint level. In the joint level, classification would be a problem if: (1) the number of features is not the same for the combined sentence levels and (2) the attributes or types of features are not matching.

2. Implementation of more robust voting algorithms for decision-making. The voting algorithms utilized in both of GA-based system and relSCAN^{+} showed that there is room for improvement with them. The Hurwicz Criterion and Minimax Regret voting methods showed that although they considered the decision of all the alternatives involved, they showed that they will most likely tend to lean towards a certain class based on their combined performance and in some rare cases as discussed in Section 4.4.1, lean towards the decision of the most impressive class alternative even if it consists of just one classifier. The maximum probability-based voting algorithm implemented in relSCAN^{+} which makes a decision per relation instance solely on just the probability of predictions between the combined outputs showed that the performance of the system can be improved if the voting algorithm is expanded to make decisions on multiple variables per relation instance.

REFERENCES

[1] Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., & Salakoski, T. (2008). Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics, 9*(Suppl 3), S6. doi:10.1186/1471-2105-9-S3-S6.

[2] Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., & Declerck, T. (2013). The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions. *J Biomed Inform, 46*(5), 914-20. doi:10.1016/j.jbi.2013.07.011.

[3] Segura-Bedmar, I., Martinez, I., & de Pablo-Sanchez, C. (2010). Extracting drug-drug interactions from biomedical texts. *BMC Bioinformatics, 11*(Suppl 5), P9.

[4] Li, J., Sun, Y., Johnson, R., Sciaky, D., Wei, C. H., Leaman, R., . . . Lu, Z. (2015). Annotating chemicals, diseases and their interactions in biomedical literature. *Proceedings of the fifth BioCreative challenge evaluation workshop*, (pp. 173-82). Sevilla. Retrieved from http://www.biocreative.org/resources/biocreative-v/proceedings-biocreative5/.

[5] Ananiadou, S., Pyysalo, S., Tsujii, J., & Kell, D. B. (2010). Event extraction for systems biology by text mining the literature. *Trends Biotechnol, 28*(7), 381-90. doi:10.1016/j.tibtech.2010.04.005.

[6] Gu, J., Qian, L., & Zhou, G. (2015). Chemical-induced disease relation extraction with lexical features. *In Proceedings of the fifth BioCreative Challenge*

Evaluation Workshop. BioCreative Organizing Committee., (pp. 220-25). Sevilla. Retrieved from http://www.biocreative.org/resources/biocreative-v/proceedings-biocreative5/.

[7] Gu, J., Qian, L., & Zhou, G. (2016). Chemical-induced disease relation extraction with various linguistic features. *Database,* baw042. doi:10.1093/database/baw042.

[8] Gu, J., Sun, F., Qian, L., & Zhou, G. (2017). Chemical-induced disease relation extraction via convolutional neural network. *Database, 2017*(1). doi:10.1093/database/bax024.

[9] Zheng, W., Lin, H., Li, Z., Liu, X., Li, Z., & Xu, B. (2018). An effective neural model extracting document level chemical-induced disease relations from biomedical literature. *J Biomed Inform, 83,* 1-9. doi:10.1016/j.jbi.2018.05.001.

[10] Peng, Y., Wei, C. H., & Lu, Z. (2016). Improving chemical disease relation extraction with rich features and weakly labeled data. *J Cheminformatics, 8,* 53. doi:10.1186/s13321-016-0165-z.

[11] Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., & Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics, 9*(Suppl 11), S2. doi:10.1186/1471-2105-9-S11-S2.

[12] Bui, Q. C., Katrenko, S., & Sloot, P. M. (2011). A hybrid approach to extract protein-protein interactions. *Bioinformatics*, 259-65. doi:10.1093/bioinformatics/btq620.

[13] Kim, S., Yoon, J., Yang, J., & Park, S. (2010). Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics, 11*, 107. doi:10.1186/1471-2105-11-107.

[14] Miwa, M., Sætre, R., Miyao, Y., & Tsujii, J. (2009). A rich feature vector for protein-protein interaction extraction from multiple corpora. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. 1*, pp. 121-30. Singapore: Association for Computational Linguistics.

[15] Vlachos, A., & Craven, M. (2012). Biomedical event extraction from abstracts and full papers using search-based structured prediction. *BMC Bioinformatics, 13*(Suppl 11), 1–11.

[16] Culotta, A., McCallum, A., & Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (pp. 296-303). Association for Computational Linguistics.

[17] Muzaffar, A. W., Farooque, A., & Usman, Q. (2015). A relation extraction framework for biomedical text using hybrid feature set. *Comput Math Methods Med, 2015*. doi:10.1155/2015/910423.

[18] Abacha, A. B., & Zweigenbaum, P. (2011). A hybrid approach for the extraction of semantic relations from medline abstracts. In G. A (Ed.), *International conference on intelligent text processing and computational linguistics* (pp. 139-50). Berlin: Springer.

[19] Jensen, L. J., Saric, J., & Bork, P. (2006). Literature mining for the biologist : from information retrieval to biological discovery. *Nat Rev Genet, 7*(2), 119-29.

[20] Chapman, W. W., & Cohen, K. B. (2009). Current issues in biomedical text mining and natural language processing. *J Biomed Inform, 42*(5), 757-9. doi:10.1016/j.jbi.2009.09.001.

[21] Zweigenbaum, P., Demner-Fushman, D., Yu, H., & Cohen, K. B. (2007). Frontiers of biomedical text mining: current progress. *Brief Bioinform, 8*(5), 358-75.

[22] Erhardt, R. A., Schneider, R., & Blaschke, C. (2006). Status of text-mining techniques applied to biomedical text. *Drug Discov Today, 11*(7-8), 315-25.

[23] Alam, F., Corazza, A., Lavelli, A., & Zanoli, R. (2016). A knowledge-poor approach to chemical-disease relation extraction. *Database (Oxford), 2016,* baw071. doi:10.1093/database/baw071.

[24] Kang, N., Singh, B., Bui, C., Afzal, Z., van Mulligen, E., & Kors, J. (2014). Knowledge-based extraction of adverse drug events from biomedical text. *BMC Bioinformatics, 16,* 64. doi: 10.1186/1471-2105-15-64.

[25] Xu, R., & Wang, Q. (2014). Automatic construction of a large-scale and accurate drug-side-effect association knowledge base from biomedical literature. *J Biomed Inform, 51,* 191-9. doi:10.1016/j.jbi.2014.05.013.

[26] Ohta, T., Pyysalo, S., Kim, J. D., & Tsujii, J. I. (2010). A Re-Evaluation of Biomedical Named Entity–Term Relations. *J Bioinform Comput Biol, 8*(5), 917-28.

[27] Leaman, R., & Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. *Pac Symp Biocomput, 13,* 652-63. doi:10.1142/9789812776136_006.

[28] Bikel, D. M., Schwartz, R., & Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Mach Learn, 34*(1-3), 211-31.

[29] Voyant, C., Notton, G., Kalogirou, S., Nivet, M. L., Paoli, C., Motte, F., & Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting: A review. *Renewable Energy, 105*, 569-82.

[30] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

[31] Bui, Q. C. (2012). *Relation extraction methods for biomedical literature*. Amsterdam: University of Amsterdam. Retrieved March 13, 2017, from http://dare.uva.nl/personal/pure/en/publications/relation-extraction-methods-for-biomedical-literature(c4e2a79a-cb1e-405e-a58d-ba8ad674759b).html

[32] Miwa, M., Sætre, R., Miyao, Y., & Tsujii, J. (2009). Protein–protein interaction extraction by leveraging multiple kernels and parsers. *Int J Med Inform, 78*(12), e39-e46.

[33] Ditterrich, T. G. (1997). Machine learning research: four current direction. *AI Mag., 18*(4), 97-136.

[34] Gams, M., Bohanec, M., & Cestnik, B. (1994). A schema for using multiple knowledge. *In Proceedings of the workshop on Computational learning theory and natural learning systems (vol. 2)* (pp. 157-70). MIT Press.

[35] Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics, 22*(3), 418-35.

[36] Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms.* John Wiley & Sons.

[37] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI, 14(2),* pp. 1137-45.

[38] Roli, F., Giacinto, G., & Vernazza, G. (2001). Methods for designing multiple classifier systems. *International Workshop on Multiple Classifier Systems* (pp. 78-87). Berlin, Heidelberg: Springer.

[39] Sharkey, A. J., Sharkey, N. E., Gerecke, U., & Chandroth, G. O. (2000). The "test and select" approach to ensemble combination. *International Workshop on Multiple Classifier Systems* (pp. 30-44). Berlin, Heidelberg: Springer.

[40] Ruta, D., & Gabrys, B. (2001). Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. *International Workshop on Multiple Classifier Systems* (pp. 399-408). Berlin: Springer.

[41] Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Syst Appl, 62*, 91-103. doi:10.1016/j.eswa.2016.06.004.

[42] Emary, E., Zawbaa, H. M., Ghany, K. K., Hassanien, A. E., & Parv, B. (2015). Firefly optimization algorithm for feature selection. *Proceedings of the 7th Balkan Conference on Informatics Conference*, *26*.

[43] Zhang, L., Mistry, K., Lim, C. P., & Neoh, S. C. (2018). Feature selection using firefly optimization for classification and regression models. *Decis Support Syst, 106*, 64-85. doi:10.1016/j.dss.2017.12.001.

[44] Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Syst, 84*, 144-61. doi:10.1016/j.knosys.2015.04.007.

[45] Ibrahim, N. M., & Zainal, A. (2017). A Feature Selection Technique for Cloud IDS Using Ant Colony Optimization and Decision Tree. *Adv Sci Lett, 23*(9), 9163-9.

[46] Ruta, D. &. Gabrys, B. (2005). Classifier selection for majority voting. *Inform Fusion*, *6*(1), 63-81. doi:10.1016/j.inffus.2004.04.008.

[47] Mitchell, M. (1998). *An introduction to genetic algorithms.* MIT press.

[48] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* Michigan: The University of Michigan Press.

[49] Davis, L. (1991). *Handbook of genetic algorithms.* New York: Van Nostrand Reinhold.

[50] Cho, S. B. (1999). Pattern recognition with neural networks combined by genetic algorithm. *Fuzzy Set Syst, 103*(2), 339-347. doi:10.1016/S0165-0114(98)00232-2.

[51] Handels, H., Roß, T., Kreusch, J., Wolff, H. H., & Poeppl, S. J. (1999). Feature selection for optimized skin tumor recognition using genetic algorithms. *Artif Intell Med, 16*(3), 283-97. doi:10.1016/S0933-3657(99)00005-6

[52] Kuncheva, L. I., & Jain, L. C. (2000). Designing classifier fusion systems by genetic algorithms. *IEEE Transactions on Evolutionary computation, 4*(4), 327-36.

[53] Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing, 19*(9-10), 699-707.

[54] Rocha, M., & Neves, J. (1999). Preventing premature convergence to local optima in genetic algorithms via random offspring generation. *In International*

Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 127-36). Berlin: Springer.

[55] Mitchell, M. (1995). Genetic algorithms: An overview. *Complexity, 1*(1), 31-39.

[56] Haupt, R. L. (1998). *Practical genetic algorithms* (Vol. 2). New York: Wiley.

[57] Bunescu, R., Ge, R., Kate, R. J., Marcotte, E. M., Mooney, R. J., Ramani, A. K., & Wong, Y. W. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artif Intell Med*, *33*(2), 139-55.

[58] Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., & Salakoski, T. (2007). BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics, 8,* 50. doi:10.1186/1471-2105-8-50.

[59] Erkan, G., Ozgur, A., & Radev, D. R. (2007). Semi-supervised classification for extracting protein interaction sentences using dependency parsing. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, (pp. 228–37). Prague.

[60] Ding, J., Berleant, D., Nettleton, D., & Wurtele, E. (2002). Mining MEDLINE: abstracts, sentences, or phrases? *Pac Symp Biocomput, 7*, pp. 326-37.

[61] Nédellec, C. (2005). Learning language in logic-genic interaction extraction challenge. *Proceedings of the 4th Learning Language in Logic Workshop (LLL05), 7*, pp. 31–37.

[62] *WBI corpus repository*. (2011). Retrieved March 5, 2018, from http://corpora.informatik.hu-berlin.de.

[63] Wei, C. H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., & Wiegers, T. (2016). Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database, 2016*, baw032. doi:10.1093/database/baw032.

[64] Davis, A. P., Wiegers, T. C., Roberts, P. M., King, B. L., Lay, J. M., & Lennon-Hopkins, K. (2013). A CTD–Pfizer collaboration: manual curation of 88 000 scientific articles text mined for drug–disease and drug–phenotype interactions. *Database (Oxford)*, bat080. doi:10.1093/database/bat080.

[65] Onye, S. C., Akkeleş, A., & Dimililer, N. (2016). Chemical-disease Relation Extraction with SVM and Enhanced Internal Features. *3rd International Conference on Data Mining, Electronics and Information Techonlogy (DMEIT'16)*, (p. 39). Istanbul, TR.

[66] Le, H. Q., Tran, M. V., Dang, T. H., & Collier, N. (2015). The UET-CAM system in the BioCreAtIvE V CDR task. *In Fifth BioCreative challenge evaluation workshop*, (pp. 208-13). Sevilla.

[67] Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (pp. 1785–94). Lisbon.

[68] Quinlan, J. (2014). *C4.5: programs for machine learning.* Elsevier.

[69] Tjen-Sien, L., Wei-Yin, L., & Yu-Shan, S. (2000). A comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Mach Learn*, 203-28. doi:10.1023/A:1007608224229.

[70] Quinlan, J. (1993). *C4.5: Programs for machine learning.* San Francisco: Morgan Kaufmann Publishers Inc.

[71] Kaur, G., & Chhabra, A. (2014). Improved J48 classification algorithm for the prediction of diabetes. *Int J Comput Appl T, 98*(22).

[72] Podgorelec, V., Kokol, P., Stiglic, B., & Rozman, I. (2002). Decision trees: an overview and their use in medicine. *J Med Syst, 26*(5).

[73] Landwehr, N., Hall, M., & Frank, E. (2005). Logistic model trees. *Mach. Learn.*, 59(1-2), 161-205. doi: 10.1007/s10994-005-0466-3.

[74] Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1), 5-32. doi: 10.1023/A:1010933404324.

[75] Chan, K. Y., & Loh, W. Y. (2004). LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics, 13*(4), 826–52, doi:10.1198/106186004X13064.

[76] Frank, E., Wang, Y., Inglis, S., Holmes, G., & Witten, I. H. (1998). Using model trees for classification. *Mach. Learn., 32*(1), 63-76.

[77] John, G. H. (1995). Estimating continuous distributions in Bayesian classifiers. *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338-45). Morgan Kaufmann Publishers Inc.

[78] Markov, Z., & Russell, I. (2007). *Probabilistic reasoning with naïve bayes and Bayesian networks.* Central Connecticut State University, Computer Science. New Britain. Retrieved August 10, 2018, from https://pdfs.semanticscholar.org/39b3/17ce5eb9ea7d14a0a3e2755dbea105328efa.pdf

[79] Wei, C. H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., & Li, J. (2015). Overview of the BioCreative V chemical disease relation (CDR) task. *In*

Proceedings of the fifth BioCreative challenge evaluation workshop, (pp. 154-66). Sevilla. Retrieved from http://www.biocreative.org/resources/biocreative-v/proceedings-biocreative5/.

[80] Barrett, N., & Weber-Jahnke, J. (2011). Building a biomedical tokenizer using the token lattice design pattern and the adapted Viterbi algorithm. *BMC bioinformatics, 12*(3), S1. doi:10.1186/1471-2105-12-S3-S1.

[81] Pechenizkiy, M. (2005). Feature extraction for supervised learning in knowledge discovery systems. *Jyväskylä studies in computing, 56*.

[82] Aladjem, M. E. (1994). Multiclass discriminant mappings. *Signal Processing, 35*(1), 1-18.

[83] Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: Wiley.

[84] Chen, M.-S., Han, J., & Yu, P. (1996). Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering, 8*(6). doi:10.1109/69.553155.

[85] Kim, S., Shin, S.-Y., Lee, I.-H., Kim, S.-J., Sriram, R., & Zhang, B.-T. (2008). PIE: an online prediction system for protein-protein interactions from text. . *Nucleic acids research* . doi:36:W411-5.

[86] Kim, M.-Y. (2008). Detection of gene interactions based on syntactic relations. *J Biomed Biotechnol*. doi:2008:371710.

[87] Sætre, R., Sagae, K., & Tsujii, J. (2007). Syntactic features for protein-protein interaction. *In The 2nd International Symposium on Languages in Biology and Medicine LBM*, (p. 319).

[88] Van Landeghem, S., Saeys, Y., De Baets, B., & Van de Peer, Y. (2008). Extracting Protein-Protein Interactions from Text using Rich Feature Vectors and Feature Selection. *In Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*. (pp. 77–84). Turku: Turku Centre for Computer Sciences (TUCS).

[89] Jain, A. K., & Chandrasekaran, B. (1982). 39 Dimensionality and sample size considerations in pattern recognition practice. *Hand Book of Statistics, 2*, 835-55. doi:https://doi.org/10.1016/S0169-7161(82)02042-2.

[90] Wu, C. (2007). *Advanced feature extraction algorithms for automatic fingerprint recognition systems.* Buffalo: University of New York. Retrieved September 29, 2018, from https://cedar.buffalo.edu/~govind/chaohong_thesis.pdf

[91] Pham, H. N. (2010). *The Impact of Overfitting and Overgeneralization on the Classification Accuracy in Data Mining*. Computer Science. Louisiana State

University. Retrieved September 30, 2018, from http://digitalcommons.lsu.edu/gradschool_dissertations/3335

[92] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. In C. C. Aggarwal (Ed.), *Data classification: Algorithms and applications* (pp. 37-64). CRC Press.

[93] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence, 97*(1-2), 273-324.

[94] Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans Knowl Data Eng, 17*(4), 491-502.

[95] Katrenko, S., & Adriaans, P. (2007). Learning Relations from Biomedical Corpora Using Dependency Trees. In K. Tuyls, R. Westra, Y. Saeys, & A. Nowé (Eds.), *Knowledge Discovery and Emergent Complexity in Bioinformatics (KDECB)* (Vol. 4366, pp. 61-80). Berlin: Springer. doi:10.1007/978-3-540-71037-0_5.

[96] Zhang, Y., Lin, H., Yang, Z., Wang, J., Zhang, S., & Sun, Y. (2018). A hybrid model based on neural networks for biomedical relation extraction. *J Biomed Inform, 81*, 83-92. doi:10.1016/j.jbi.2018.03.011.

[97] Dimililer, N., Varoğlu, E., & Altınçay, H. (2007). Vote-Based Classifier Selection for Biomedical NER Using Genetic Algorithms. In J. Martí, J. M.

Benedí, A. M. Mendonça, & J. Serrat (Ed.), *Pattern Recognition and Image Analysis. IbPRIA 2007. Lecture Notes in Computer Science*. 4478, pp. 202-9. Berlin, Heidelberg: Springer.

[98] Pons, E., Becker, B. F., Akhondi, S. A., Afzal, Z., Van Mulligen, E. M., & Kors, J. A. (2016). Extraction of chemical-induced diseases using prior knowledge and textual information. *Database, 2016*, baw046. doi:10.1093/database/baw046.

[99] Giot, R., & Rosenberger, C. (2012). Genetic programming for multibiometrics. *Expert Systems with Applications, 39*(2), 1837-47.

[100] Ruta, D., & Gabrys, B. (2000). An overview of classifier fusion methods. *Computing and Information Systems, 7*(1), 1-10.

[101] Arasu, A., Götz, M., & Kaushik, R. (2010). On active learning of record matching packages. *In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, (pp. 783-94).

[102] Gabrys, B. (2002). Combining neuro-fuzzy classifiers for improved generalisation and reliability. *Proceedings of the International Joint Conference on Neural Networks (IJCNN2002), A Part of the WCCI2002 Congress*, *3*, pp. 2410-15. Honolulu.

[103] Kuncheva, L. (2000). Fuzzy Classifier Design. *Springer Science & Business Media, 49*.

[104] Hao, H., Liu, C. L., & Sako, H. (2003). Comparison of genetic algorithm and sequential search methods for classifier subset selection. *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on IEEE*, (pp. 765-69).

[105] Gabrys, B., & Ruta, D. (2006). Genetic algorithms in classifier fusion. *Appl Soft Comput, 6*(4), 337-47. doi:10.1016/j.asoc.2005.11.001.

[106] Beitia, I. M. (2015). *Contributions on distance-based algorithms, multi-classifier construction and pairwise classification.* Doctoral dissertation, Universidad del País Vasco-Euskal Herriko Unibertsitatea, San Sebastián. Retrieved May 25, 2018, from https://addi.ehu.es/bitstream/handle/10810/15943/TESIS_I%C3%91IGO_MENDIALDUA_BEITIA.pdf?sequence=7

[107] Bellare, K., Iyengar, S., Parameswaran, A., & Rastogi, V. (2013). Active sampling for entity matching with guarantees. *ACM Transactions on Knowledge Discovery from Data (TKDD), 7*(3), 12.

[108] Bennett, P. N., & Carvalho, V. R. (2010). Online stratified sampling: evaluating classifiers at web-scale. *In Proceedings of the 19th ACM international conference on Information and knowledge management. ACM.*, (pp. 1581-84).

[109] McDowell, L. K., Gupta, K. M., & Aha, D. W. (2009). Cautious collective classification. *J Mach Learn Res*, 2777-836.

[110] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Mag.*, *29*(3), 93.

[111] Zenobi, G., & Cunningham, P. (2001). Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error. *In European Conference on Machine Learning* (pp. 576-87). Berlin: Springer, , Heidelbe.

[112] Zhou, H. W., Deng, H. J., & He, J. (2015). Chemical-disease relations extraction based on the shortest dependency path tree. *In Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, (pp. 214-19). Sevilla.

[113] Davis, A. P., Murphy, C. G., Saraceni-Richards, C. A., Rosenstein, M. C., Wiegers, T. C., & Mattingly, C. J. (2008). Comparative Toxicogenomics Database: a knowledgebase and discovery tool for chemical–gene–disease networks. *Nucleic Acids Res*, *37*(Database issue), D786-D92. doi:10.1093/nar/gkn580.

[114] Xu, J., Wu, Y., Zhang, Y., Wang, J., Liu, R., & Wei, Q. (2015). UTH-CCB@ BioCreative V CDR task: identifying chemical-induced disease relations in biomedical text. *In Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, (pp. 254–59). Sevilla. Retrieved from http://www.biocreative.org/resources/biocreative-v/proceedings-biocreative5/.

[115] Lowe, D. M., OBoyle, N. M., & Sayle, R. A. (2016). Efficient chemical-disease identification and relationship extraction using Wikipedia to improve recall. *Database, 2016*, baw039. doi:10.1093/database/baw039.

[116] Jiang, Z., Jin, L. K., Li, L. S., Qin, M., Qu, C., Zheng, J., & Huang, D. (2015). A CRD-WEL system for chemical-disease relations extraction. *In The fifth BioCreative challenge evaluation workshop*, (pp. 317–26). Sevilla. Retrieved from http://www.biocreative.org/resources/biocreative-v/proceedings-biocreative5/.

[117] Ellendorff, T. R., Clematide, S., van der Lek, A., Furrer, L., & Rinaldi, F. (2015). Ontogene term and relation recognition for CDR. *In: The fifth BioCreative challenge evaluation workshop*, (pp. 305–310). Sevilla. Retrieved from http://www.biocreative.org/resources/biocreative-v/proceedings-biocreative5/.

[118] Panyam, N. C., Verspoor, K., Cohn, T., & Ramamohanarao, K. (2018). Exploiting graph kernels for high performance biomedical relation extraction. *J Biomed Semantics, 9*(1), 7. doi:10.1186/s13326-017-0168-3.

[119] Zhou, H., Deng, H., Chen, L., Yang, Y., Jia, C., & Huang, D. (2016). Exploiting syntactic and semantics information for chemical-disease relation extraction. *Database (Oxford), 2016*, baw048. doi:10.1093/database/baw048.

[120] Xu, J., Wu, Y., Zhang, Y., Wang, J., Lee, H. J., & Xu, H. (2016). CD-REST: a system for extracting chemical-induced disease relation in literature. *Database (Oxford), 2016*, baw036. doi:10.1093/database/baw036.

[121] Zhou, H., Yang, Y., Liu, Z., Liu, Z., & Men, Y. (2017). Integrating Word Sequences and Dependency Structures for Chemical-Disease Relation Extraction. In M. Sun, X. Wang, B. Chang, & D. Xiong (Eds.), *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data* (pp. 97-109). Springer, Cham. doi:https://doi.org/10.1007/978-3-319-69005-6_9.

[122] Onye, S. C., Akkeleş, A., & Dimililer, N. (2018). relSCAN - a system for extracting chemical-induced disease relation from biomedical literature. *J Biomed Inform, 87*, 79-87.

[123] Leaman, R., Wei, C. H., & Lu, Z. (2015). tmChem: a high performance approach for chemical named entity recognition and normalization. *J Cheminform, 7*(Suppl 1 Text mining for chemistry and the CHEMDNER track), S3. doi:10.1186/1758-2946-7-S1-S3.

[124] Schuemie, M. J., Jelier, R., & Kors, J. A. (2007). Peregrine: Lightweight gene name normalization by dictionary lookup. *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, (pp. 131-33).