A DATABASE QUERY BASED SOLUTION FOR CHEMICAL COMPOUND AND DRUG NAME RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞLAR ATA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2014

Approval of the thesis:

A DATABASE QUERY BASED SOLUTION FOR CHEMICAL COMPOUND AND DRUG NAME RECOGNITION

submitted by **ÇAĞLAR ATA** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Tolga Can _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Assoc. Prof. Dr. Halit Oğuztüzün _____
Computer Engineering Dept., METU

Assoc. Prof. Dr. Tolga Can _____
Computer Engineering Dept., METU

Asst. Prof. Dr. İsmail Şengör Altıngövde _____
Computer Engineering Dept., METU

Asst. Prof. Dr. Aybar Can Acar _____
Informatics Institute, METU

Dr. Ruken Çakıcı _____
Computer Engineering Dept., METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: ÇAĞLAR ATA

Signature :

ABSTRACT

A DATABASE QUERY BASED SOLUTION FOR CHEMICAL COMPOUND AND DRUG NAME RECOGNITION

Ata, Çağlar
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Tolga Can

September 2014, 42 pages

Searching structured information in unstructured free text is one of the most difficult challenges in computer science. Relevant information from documents has to be ready for use not only with accurate precision but also be ready in a fast manner. Although numerous studies on document searching has been published, only few of them specifically target chemical compound and drug names. Chemical compound and drug names have specific morphological properties. These unique morphological properties have to be examined before developing automatic text searching methods. These properties should also be integrated into chemical compound and drug name retrieval systems.

In this thesis, we focus on named entity recognition problem with a newly proposed method on chemical compound and drug name recognition model using queries on a very domain specific database. PubChem Power User Gateway (PUG) system is used

as the main database for this specific domain to demonstrate the method. Chemical compound and drug name grammar and morphological properties are used as base for constructing the model. These features are deeply examined and used for optimizing the queries and increase the recall with precision on finding relevant chemical compound and drug names in documents. This new proposed method also presents a unique chemical compound and drug name tokenizer designed for specifically tokenizing chemical words in an article. The proposed method is applied on significant amount of chemical compound and drug name containing documents. Results of our proposed method are compared against the state of the art methods that target the same problem.

Keywords: Chemical compound name, drug name, text information retrieval, database queries

ÖZ

VERİTABANI SORGULAMA TABANLI KİMYASAL BİLEŞİK VE İLAÇ İSMİ TANIMA METODU

Ata, Çağlar
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Tolga Can

Eylül 2014, 42 sayfa

Yapısal olmayan serbest metinlerde yapısal bilgi aramak bilgisayar bilimindeki en zor problemlerden biridir. Dokümanlardan uygun bilginin sadece kesin hassasiyet ile değil ayrıca hızlı şekilde kullanıma hazır hale gelmesi gerekmektedir. Her ne kadar sayısız çalışma doküman araştırma alanında yayımlanmışsa da, sadece içlerinden bir kaçı özellikle kimyasal bileşik ve ilaç isimlerini amaçlamıştır. Kimyasal bileşik ve ilaç isimleri doğalarında bazı özgün biçimsel özellikler taşımaktadır.

Bu tezde, metin bilgi bulup getirme problemi, çok belirli bir alan üzerindeki veritabanı sorgulamalarını kullanarak kimyasal bileşik ve ilaç isim çıkarma modeline dayalı yeni sunulan bir yöntem ile ele alınmıştır. PubChem Power User Gateway (PUG) sistemi bu metodu örneklemek için ana veritabanı olarak kullanılmıştır. Kimyasal bileşik ve ilaç isimlerinin dil bilgisi ve biçimsel özellikleri modeli oluşturmada temel olarak kullanılmıştır. Bu özellikler derin bir şekilde incelenmiş ve

dokümanlardaki kimyasal bileşik ve ilaç isimlerinin bulunmasında kullanılan sorguların iyileştirilmesi ile kesinlik ve hassasiyetin arttırılmasında yardımcı olarak kullanılmıştır. Yeni önerilen yöntem ayrıca verilen dokümanda özellikle kimyasal kelimeleri sınıflandırmak için tasarlanmış özgün bir kimyasal bileşik ve ilaç isim girdi sınıflayıcısı sunmaktadır. Önerilen metod kayda değer miktarda kimyasal bileşik ve ilaç adı içeren doküman üzerinde uygulanmıştır. Sunduğumuz yöntemin sonuçları bu arama problemi için özellikle tasarlanan en gelişkin yöntemler ile karşılaştırılmıştır.

Anahtar kelimeler: Kimyasal bileşik adı, ilaç adı, metin bilgi getirme, veritabanı sorguları

I dedicate this thesis to my lovely son, Çağan.

ACKNOWLEDGMENTS

I thank to my parents and my wife for their continuous support and encouragement. They always helped and motivated me when I am frustrated and depressed throughout the last three years.

I would like to express my sincerest gratitude to supervisor, Assoc. Prof. Dr. Tolga Can, for his generous guidance, great kindness and for his patience with me. Without his advice and encouragements throughout the last three years it would be very difficult to complete this research.

I also want to thank to all Middle East Technical University Computer Engineering Department faculty members as providing us a high quality standards in computer engineering education.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Extracting structured information in unstructured text such as web pages and documents is one of the most difficult challenges in computer science. Relevant information from documents has to be ready for use not only with accurate precision but also be ready in a fast manner. As scientific research depends on results in other domains, automated methods for information retrieval are needed for more effective information exchange. Because of this need, information systems that focus on extraction of information must work on not only the examined domain but also on other different domains. As information is spread through the world by internet at the speed of light and research projects has to be finished in short timelines, these systems must be as fast as possible dealing with huge amounts of data.

In text information retrieval systems, depending on the methodology, storing vital information is another problem to deal with. Databases has to deal with millions of entries and proposed methods has to be elegant to overcome performance issues. As fresh information is flooding continuously, systems that use database organization must be flexible to integrate with incoming new information. Chemical Abstract Service (CAS) Registry, which gives a registry number to a chemical substance when it enters the CAS Registry database, contains more than 60 million substances [1]. Also more than 15000 registry numbers are given to new chemical substances each day in CAS Registry database [2]. As universities and research facilities have

different kind of data centers and architectures, proposed methods must have the capability to be implemented in most of the common platforms.

There are several studies for document categorization and indexing but only few of them specialize on chemical compound and drug names [3]. Chemical compound names are complex in nature and a chemical compound may have more than one chemical name. Molecular formulas are sophisticated as well, and different combinations may address to the exact molecular formula [3]. It is difficult to derive regular expressions and rules to construct and validate molecular formulas. According to Corbett et al., it is difficult to parse documents which contain molecular formulas because 90 percent of these formulas do not have whitespace and 22 percent of these formulas are adjacent to or have hyphens and dashes [4]. These properties of chemical substances are explained in detail in Chapter 2.

Text tokenization is also another important vital area in text information retrieval. Methods that describe a full text searching system has to provide an effective text tokenizer. Otherwise the work on the tokenizer must be propagated to other parts of the system that will have a bad influence on the performance.

In this thesis, a method based on database queries to solve those issues discussed above is presented. We create a database with more than 145 million rows containing molecular formulas, compound and drug names with their synonyms. We created an English dictionary to use as an extensive stop word list and determine tokens which can be queried in the database. We also have a post processing phase with a small set of rules to merge consecutive chemical names. Our method implementation acquired 71% recall and 58% precision on a benchmark dataset containing 3500 articles provided by BioCreative IV (Critical Assessment of Information Extraction systems in Biology) Chemdner (chemical compound and drug name recognition) challenge with a total running time of 14 minutes [5]. The proposed method is described in detail and the experimental results are provided in the following chapters.

1.2 Organization

This thesis is organized as follows. In Chapter 2, literature and background information about chemical compound and drug names and methods similar to the method that is proposed in this thesis are provided in detail. Our approach to the mentioned problem is discussed in Chapter 3. Experiment results and comparison with other state of the art systems are given in Chapter 4. Conclusion is presented in Chapter 5.

CHAPTER 2

BACKGROUND INFORMATION

In this chapter, background information about this work is given. Also other related works in the literature are discussed.

2.1 Chemical Compounds and Drugs

Chemical compounds and drugs are pure chemical substances that consist of more than two chemical elements. Chemical compounds can be separated into simpler elements by chemical reactions [6]. They have well defined unique chemical structure. Atoms with a constant ratio are bond to each other in a spatial three dimensional arrangement to build a unique chemical compound.

Drugs are specific chemical compounds known to have biological effects on living creatures. Also pharmacology, which is a branch of medicine, is concerned with the effect of drugs, treatment, prevention, and diagnoses of diseases [7]. Drugs can be expressed as sets of chemical compounds; however, drug name recognition is a different problem, since, naming of drugs and chemical compounds are completely different. Naming of chemical compounds is described in the next section.

2.2 Chemical Compound and Drug Name Rules

Before proposing a new text information retrieval method, domain knowledge of searched text is vital. Chemical compounds and drugs have unique morphological

and grammar characteristics. These properties have huge impact on performance and recall / precision. In this section these properties are described.

Chemical compounds can be described in either common names, systematic words that obey chemical nomenclature or chemical formulas. In the following sections, chemical formulas, systematic nomenclature and daily words are described briefly.

2.2.1 Chemical Formulas

Chemists describe compounds using formulas. It is a way of expression about the ratio of atoms that forms compounds. Chemical formulas consist of chemical element symbols (O for oxygen, C for carbon ...), numbers (amount of atoms: O_2 – 2 oxygen atoms), plus (+) and minus (-) signs and other alphabetical symbols (dashes, parentheses brackets ...) [8]. A single line of letters are used to constitute a formula which may include sub or superscripts. A chemical formula is different from common chemical names as formulas do not contain words. Chemical formulas may give certain chemical characteristics, but they do not cover all the properties of the chemical substance as a structural formula does.

An example of chemical formula is given in Figure 1.1. A structural formula is a graphical way of expressing chemical compounds. It contains also spatial molecular structure information about how the atoms are arranged in three dimensional space. Also chemical bonds can be shown in this representation. An example of structural formula is given in Figure 1.2. In this thesis, only text searching problem is considered; therefore, structural formulas are not handled. In following chapters, by 'formula', we refer to 'chemical formula' by above definitions [9].

Butane: $CH_3CH_2CH_2CH_3$

Titin Protein: $C_{169723}H_{270464}N_{45688}O_{52}$

Figure 2.1 Examples of chemical formulas.

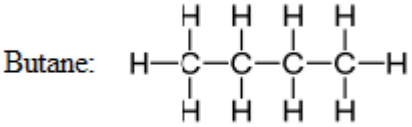

Figure 2.2 An example structural formula.

Molecular formulas which are subsets of chemical compound and drug names are complex, because one molecular formula can have several different variants. [3]. Isomers like glucose and fructose are a good example for this statement. These compounds have same chemical formula but different chemical structure and common name.

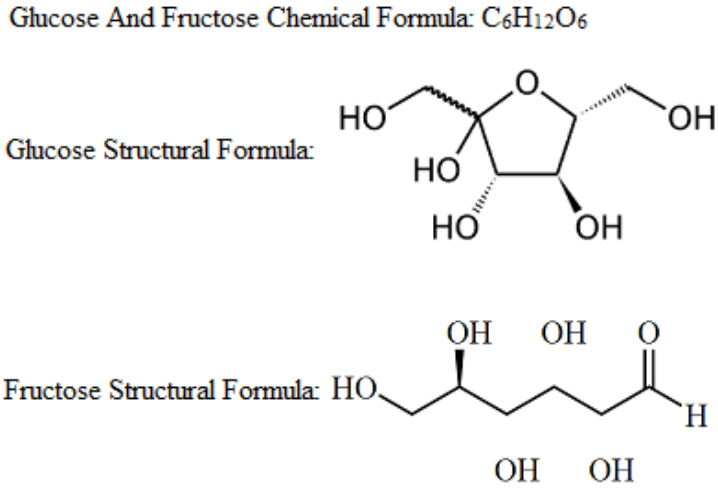

Figure 2.3 Example of substances referring to the same chemical formula.

2.2.2 Chemical Nomenclature

Chemical nomenclature is a set of rules that developed by the International Union of Pure and Applied Chemistry [10]. These rules are used to generate systematic names [11]. In chemical nomenclature, substances are divided into two groups as organic and inorganic compounds. All the IUPAC nomenclature of organic chemistry is described in "Blue Book" [12, 13]. IUPAC rules for inorganic compounds are contained in another publication known as 'Red Book' [14].

The main purpose of chemical nomenclature is to ensure that each chemical name have one single unique chemical substance and avoid ambiguity. It is a one-to-many mapping since a chemical substance can have multiple chemical names which obey chemical nomenclature.

Organic chemistry nomenclature is used to name organic chemical compounds. According to the Blue Book of IUPAC, any organic compound can have a name following the rules and from this name, a structural formula can be created [12, 13]. In organic chemistry nomenclature, all the rules are described in great detail. In the proposed method of this thesis, main categories of naming are taken into account. These are alkanes, alkynes, alcohols, halogens, ketones, aldehydes, carboxylic acids, ethers, amines, amides, and cyclic compounds. There is no common naming algorithm implemented in computer science for these conventions.

Inorganic chemistry nomenclature is used to name inorganic chemical compounds. According to the Red Book of IUPAC, all inorganic compounds can have a name from which an unambiguous formula can be prepared [14]. In this thesis, the proposed method covers ions and hydrates which are the main subcategories of inorganic compound nomenclature. Examples of chemical nomenclature are shown in Figure 2.4.

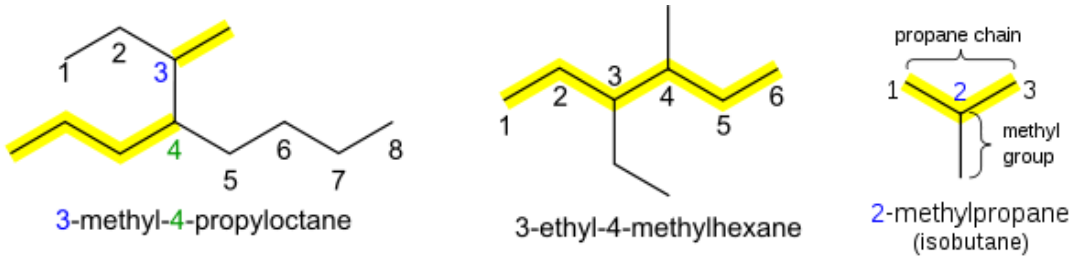

Figure 2.4 Chemical nomenclature examples.

2.2.3 Chemical Compound Common Names

Chemical common names are regular English words that are used to describe certain chemical compounds. These names can vary from daily words to domain specific new drugs.

Chemical compounds can have more than one common name. As an example, H_2O molecular formula can refer to water and aqua. All of the known drugs have also common names used for marketing purposes. In our proposed method, all the drug names known so far are taken into account when preparing the corpora.

2.2.4 Chemical Naming Usage Distribution in Documents

The distribution of different naming conventions in literature is approximately known: only 32% of chemicals mentioned in documents properly follow IUPAC standards, 34% of them are common names, 13% of them abbreviations, 20% of them are combinations of mentioned types of naming conventions [15]. According to Klinger et al. [15], any method which is to cover the chemical name text searching problem must handle all of the naming conventions in order to be of practical value.

2.3 Methods Used to Recognize Chemical Compounds in Documents

It is difficult to derive regular expressions and rules to construct and validate molecular formulas. According to Corbett et al., it is difficult to parse chemical name mentioning documents which contain molecular formulas, because 90 percent of these formulas do not contain whitespace and 22 percent of these formulas are adjacent to or have hyphens and dashes [4]. It is difficult to derive regular expressions and rules to construct and validate molecular formulas. According to Corbett et al., it is difficult to parse documents which contain molecular formulas because 90 percent of these formulas do not have whitespace and 22 percent of these formulas are adjacent to or have hyphens and dashes [4, 16]. Because of this fact, methods to find chemical substances in documents can be divided into three categories as dictionary based, context based and morphology based approaches [3]. These categories are discussed in the following sections.

2.3.1 Dictionary Based Methods

Dictionary based approaches are methods which find chemical names in documents by comparing names by a dictionary or a catalog. This approach is commonly called as matching or lookup. The most vital part of these methods is to have a comprehensive dictionary that covers nearly all of the search domain [3]. Dictionaries can be built manually as well as automatically by chemical databases that are available like Pharmspresso, Polysearch, DrugBank, UMLS, PubChem, ChEBI [3, 17, 18, 19, 20, 21, 22].

One of the drawbacks of dictionaries is their sizes. These type of chemical dictionaries can have millions of entities, while an ordinary gene name dictionary has tens of thousands entries [3]. As an example, the *jochem* joint chemical dictionary contains nearly two million entries [3]. These type of dictionaries need manual curation and maintenance; also statistical properties are used to help the curation phase in many cases [3, 23].

Dictionaries are very fast and accurate when querying chemical names written correctly. Otherwise, if the chemical name is written in a wrong manner, direct querying these kind of databases gives poor results. To overcome this issue, researchers used some algorithms like the Levensthein distance algorithm, which is a distance calculation algorithm in which the similarity of two words are calculated based on the number of different characters to be changed in one sentence to transform to the other one [24]. Levensthein distance can be formulated as follows: Given two strings *a*, *b*, and a function *l* to calculate the length of a string, the Levensthein distance of the first m and n characters ($L(m,n)$) of these strings can be calculated as:

$$L(l(a), l(b))$$
$$= \begin{cases} \max(l(a), l(b) & if\ \min\big(l(a), l(b)\big) = 0 \\ \min\begin{cases} L\big(l(a)-1, l(b)\big)+1 \\ L(l(a), l(b)-1)+1 \\ L(l(a)-1, l(b)-1)+1_{(a_{l(a)} \neq b_{l(b)})} \end{cases} & otherwise \end{cases}$$

$1_{(a_{l(a)} \neq b_{l(b)}}$ is a function and equal to zero if the last characters of *a* and *b* are the same characters and equal to one if they are different. The first equation under first minimum statement is deletion, the second one is insertion and the third one is match/mismatch [25].

Levensthein distance has some problems on large dictionaries because for each word in the dictionary a distance must be calculated for the query word, as a result this approach is computationally costly [3]. Also regular expressions can be used to search the dictionary. In this approach, a query word is compared with each word in the dictionary as regular expressions may provide a possible output. As discussed in earlier sections, this method is not feasible because 90 percent of chemical formulas do not have whitespace and 22 percent of the formulas are adjacent to or have hyphens and dashes in documents containing chemical names [4].

LeadMine is a dictionary based chemical name method which is designed specifically to find chemical names in documents [26]. LeadMine has a dictionary of chemical words over 2.94 million, which supports the previous claims on the size of these type of dictionaries [26]. OCMiner is another dictionary based approach, which uses a dictionary containing 14 million compounds [27].

2.3.2 Morphology Based Methods

Chemical nomenclatures as explained in previous sections is in fact grammars which use finite set of symbols that are chemical name segments, numbers and other alphabetical symbols [3]. Chemical name segments are special words that are used to construct chemical words. As an example ‘12-butyl-4,8-diethyl’ is a chemical and ‘butyl’ and ‘diethyl’ are chemical name segments. These segments are more likely to appear in chemical words rather than regular English words, so searching these kind of segments helps finding chemical names in documents [3]. There is an available dictionary of those segments provided by Chemical Abstract Services (CAS) [28].

This type of segmentation dictionaries were used by the SEG method. In this method, segmented parts are basically searched through provided segmentation dictionaries

and results of these methods were compared with two different types of approaches. Results show no extra benefit provided by the SEG method compared with other methods [29]. Also these type of techniques deal only with segmented parts and do not consider whole chemical words [3].

2.3.3 Context Based Methods

Methods which bases on document context basically try to harvest information from a given document by known statistical properties, using natural language processing methods or by manually created rules based on domain knowledge [3].

Context aware methods also aim to find patterns specific to the applied domain to gather relevant information. There are several known methods that can be applied to chemical name searching, and one of them outperforms the others. This technique, called Conditional Random Fields (CRFs) is introduced in 2001 and has been used as a popular tool especially for named entity recognition tasks [3, 30].

CRFs are one of the statistical modelling methods used in text mining, pattern recognition and machine learning. CRFs are undirected graphical models which aim to calculate probabilities on a given input. CRFs can be applied on parsing sequential data used in computer vision, text searching, biological patterns, named entity recognition systems or shallow parsing [30, 31, 32, 33, 34].

We can define a CRF as follows: Let X be a vector of observations and Y be a vector contains random variables, G=(V,E) is a graph having properties such that $Y=(Y_v)_{v\epsilon V}$, so that G indexes Y with its vertices. So, (X,Y) is called a conditional random field when the random variables Y_v which are conditioned on vector X following the markov property with the graph as:

$$p\,(Y_v \mid X, Y_w, w \neq v) = p\,(Y_v \mid X, Y_w, w \sim v)$$

where ~ is used as w and v are neighbors and *p* function calculates the probability. [30, 33, 35]. In summary graphical models are used by CRF to calculate the probability of P(Y | X) in which $Y = (y_1, y_2, ..., y_n)$ is a possible output of labeling given input vector $X = (x_1, x_2, ..., x_n)$. In context based approaches, X is given as the

tokenized strings of analyzed document [15]. In chemical name recognition using CRF based models, one token is labeled conditionally based on the previous token labels, so features extracted from the targeted token do not only effect that token but it also helps identifying the neighbor tokens. Each token is assigned with a set of features with the information extracted by CRF. These features can vary with respect to the length of the token, number of vowels in the token and other morphological properties like presence of numbers, dashes, parentheses or the segment of a token that exists in segment dictionaries [3]. Morphological features have been proved to be the most effective and discriminative ones in searching for gene and protein names in documents [36]. In Figure 2.5, effect of a morphological feature is shown if it is removed from the CRF feature set [36].

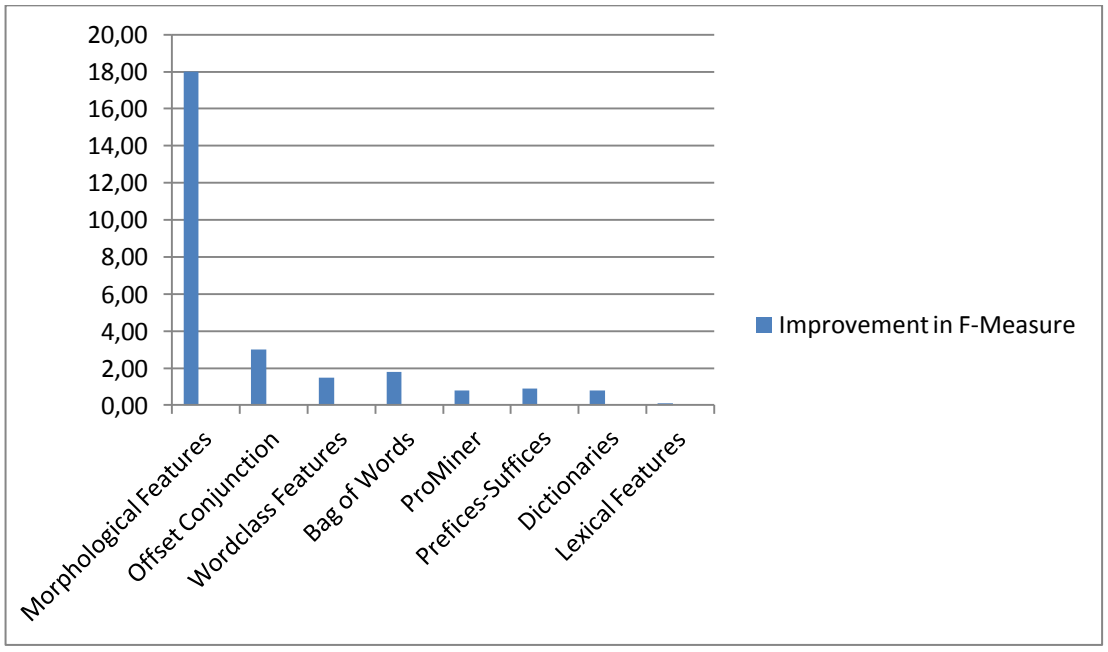

Figure 2.5 Influence of features if they are included to CRF feature set.

CHAPTER 3

PROPOSED METHOD

In this chapter, a dictionary based method which uses a comprehensive database to deal with those issues discussed for finding chemical and drug names from documents discussed in previous chapters. Our proposed method is in the category of dictionary based solutions. A database with more than 145 million rows containing molecular formulas, compound and drug names with their synonyms is used to find chemical names. We use an English dictionary to use as an extensive stop word list and determine tokens which can be queried in the database. There also exist a post processing phase with a small set of rules to merge consecutive chemical names. The proposed method is developed and tested on a benchmark dataset which is provided by BioCreative Conference IV Chemdner task and the proposed method acquired 71% recall and 58% precision on this set with a total running time of 14 minutes for 3500 articles [5, 37]. The proposed approach is described in detail and the experimental results are provided in the following chapters.

3.1 Constructing the Dictionary

3.1.1 Downloading Relevant Compound Data

We used the PubChem Power User Gateway (PUG) to build the database [38]. PubChem PUG supplies an interface to query compounds, documents, substances and formulas. This system provides information in various file formats. We downloaded XML files from PUG system containing formulas and chemical name

synonyms. Each substance in the PUG database has a unique PubChem Compound Identifier (CID) [39]. We downloaded more than 71 million compounds with their more than 76 million synonyms in less than a day with disc space usage of 15 GBs.

Parsing and storing these XML data is another challenge. We investigated using database management systems (DBMS) such as PostreSQL, MySQL and XML DBMSs such as BaseX and eXist-db; however, these do not have the capability to query huge tables and index large columns of tables efficiently and effectively [40, 41, 42, 43]. XML databases gave poor performance results over xml file sizes 150000 and traditional databases like MySQL and PostreSQL was not handling tables containing 70 million rows with ease. We decided to use traditional databases and chose DB2 Express-C as our DBMS which is free to deploy, distribute and develop [44]. This database is free of charge supporting up to 16 GBs of RAM and two CPU cores. This hardware limitation was not a problem for our approach, since we plan to build a full functional system which can easily be run even on standard desktops. In addition, DB2 has the capability to index strings containing more than 1000 characters, which is an intended property; because chemical compound names can be very long such as PUG system contains some synonyms which have lengths more than 900.

XML files were parsed by our parser module and we created one table for molecular formulas and one table for synonyms, each contains over 71 million rows. The synonym table contains synonyms of a chemical formula. From these two tables, search engines can find other naming of chemical and drug names when one of them is found from any context. The phases of constructing the database are shown on Figure 3.1.

Figure 3.1 Database preparation phases.

3.1.2 Preparing stop words to improve performance

The PUG system serves names in raw xml files. To be able to harvest relevant information and prepare the data to be queried by any system, the data must be analyzed, cleaned and enriched with domain knowledge. Our approach is to intersect prepared chemical database with one of the largest English dictionary available. We queried all of the words from NI-2Webster's New International Dictionary with the constructed database [45]. The output of this operation is an intersection of words. This intersected list is analyzed and the found chemicals in the list are remove from English dictionary. The intersected words that exist in the database are removed, since these words might be chemical. The chemical-free English dictionary which is an output of this process is used to filter and ignore non-chemical English words in an article even before directly querying them in our database. We used this dictionary as a list of stop words. We aimed to enhance performance of the system in terms of time by using this filter. In Figure 3.2 the filtering process is shown.

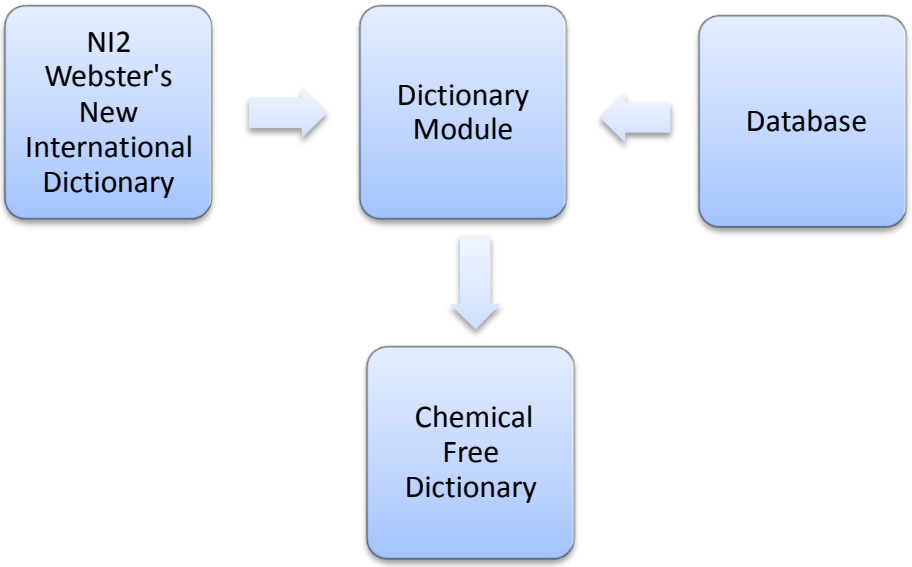

Figure 3.2 Construction of stop words.

3.2 Chemical and Drug Name Tokenizer

Tokenization is the method of splitting a text into words or other domain specific meaningful segments called tokens. It is heavily used in linguistics, text mining, and information retrieval areas of computer science. Tokenization is one of the first major steps of analyzing a document and it is vital to choose suitable tokenization method for a dataset [46].

Chemical and drug name properties are mentioned in detail in Chapter 2. Studies showed that methods to find chemical substances use morphological properties of chemical words as an important feature of their systems. To be able to tokenize a word containing those morphological properties is a domain-specific process. Most of the tokenizers available today are so generic or so specific to a certain non-chemical domain. To fill this gap, our approach proposes a tokenization technique based on the properties of chemical words. Chemical nomenclature allows chemical words to have numbers, parentheses, dashes, dots, semicolons including Greek symbols in word structure. A typical tokenizer would detect these symbols in other

meanings, resulting wrong tokenizing of words in sentences. To overcome this issue, we searched through development data set which contains 3500 articles containing chemicals and listed all the possible characters that chemical words contain. This list contains 18 different characters. List of these characters are presented In Table 3.1.

Table 3.1: List of symbols and characters handled by our tokenizer

Symbols	
(	{
)	}
,	-
;	‘
[	=
]	+
/	α
β	μ
θ	λ

Our approach depends on simply changing these symbols to a non-meaningful and non-English unique string of length 18. The length 18 is chosen according to a study about word lengths frequencies and distributions and only 413 words in English have length 18 and it is guaranteed that none of them is one of the unique strings that are used. In Figure 3.3 the distribution of English words by length is provided [47].

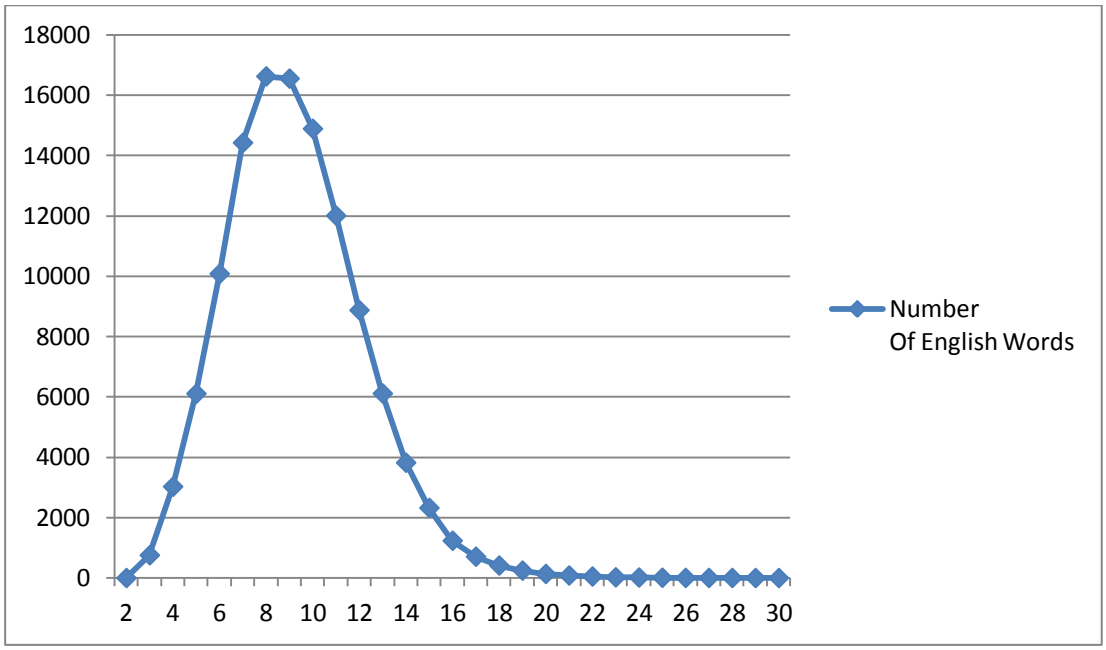

Figure 3.3 Distribution of English words with respect to their lengths.

After changing words, the targeted document is sent to an English parser. We used Stanford Tokenizer which is a commonly used and publicly available tokenizer [48]. After tokenizing the words, a post process scans all these symbols and puts them in appropriate places. Dots and semicolons are handled after this step to overcome ambiguous results.

To illustrate the issue, we convert '1,2-propadiene' which is chemical substance name to '1semicolonsemicolon2dashdashdashdashdapropadiene' and pass this word in sentence to the tokenizer and the post processing phase would give us '1,2-propadiene' as needed. Without this process, possible outcomes would be '1' as a token, ',' as another token and '2-propadiene' as the third token. Without the proposed approach several rules have to be specified to merge these three output tokens into a single token.

3.3 Query Processor and Rules

Our proposed method analyzes an input list of tokens and it outputs a list of tokens in which tokens are marked and merged as chemicals and non-chemicals.

Several chemical and drug names are analyzed to find the maximum word length of a chemical name. We figured out that almost all of the chemical names are at most 4 words length. We found out this value by analyzing the manually annotated articles provided by the BioCreative IV challenge [5, 37]. In our approach, a window size between 2 to 4 words is selected and we used these windows to slide on the tokens found by our tokenizer and each of these windows are queried in the database. Output of these querying are scored similar to the words that are individually queried, i.e., the whole window is marked as identified chemical names. A confidentially score of 1.0 is assigned to these windows if they are found directly in the database. The score value range is between 0 and 1.

If the sliding window is found in the database as a result of a similarity search query in which where condition contains "like" phrase, a score calculated with the Levensthein distance algorithm mentioned in the previous chapter is assigned. The performance problem of the Levensthein algorithm is resolved, since the tokens do not need to be checked with each word in the database. Instead the distance is calculated between the token and the result of the 'like' query which is a small finite set. Let S be the score function, t is the token, R is the result set given by the database and L is the Levensthein distance:

$$S\,(\mathrm{t},\mathrm{R}) = \frac{\mathrm{Length}(min(\,\mathrm{L}\,(\mathrm{ri}\,|\mathrm{ri}\in\mathrm{R}\,,\ \mathrm{t}\,)) - min(\,\mathrm{L}\,(\mathrm{ri}\,|\mathrm{ri}\in\mathrm{R}\,,\ \mathrm{t}\,)))}{\mathrm{Length}(min(\,\mathrm{L}\,(\mathrm{ri}\,|\mathrm{ri}\in\mathrm{R}\,,\ \mathrm{t}\,))}$$

Figure 3.4 Formula to calculate score of a token.

All the individual tokens of a window with no query results are queried to the database one by one. The individual tokens are marked with score of 1.0, if they are found directly in database. Otherwise if the length of the token is above 18, which is a threshold we determined by distribution of the length of English words, we again apply a *like* query in that case and calculate a score with the Levensthein distance algorithm with the formula given in Figure 3.4.

As mentioned in morphology-based approaches section, segment dictionaries can be helpful in finding chemical names in a document. Our approach splits a token into smaller segments based on morphological properties to contain dashes, parenthesis

and queries those segments if their size is above four. This idea is derived from a Bayesian classifier approach for analyzing text for chemical names [29]. The experiments show that the number 4 was the optimal value for the n-gram sliding window method [29]. The segments are queried in the database, if they are found then the whole token is assigned a score based on the length ratio of the segment and the token.

We only query the token in the database table containing molecular formulas, if the token contains all letters in the uppercase. We achieved this assumption after studying the annotated which are manually annotated and found out that nearly all of the chemical formulas in documents are written with all letters in uppercase.

The algorithm to query tokens using a window size of 4 is shown as pseudo code in Figure 3.5.

```
Input 1: List of parsed tokens, T= {t1, t2, t3, .. ,tn}
Main Function
For i=0 to i=n
    if(i+3<=n)
        if(queryToken(ti,ti+1,ti+2,ti+3))
            markAsChemical(ti,ti+1,ti+2,ti+3)
            i ← i+3 & continue
    if(i+2<=n)
        if(queryToken(ti,ti+1,ti+2))
            markAsChemical(ti,ti+1,ti+2)
            i ← i+2 & continue
    if(i+1<=n)
        if(queryToken(ti,ti+1))
            markAsChemical(ti,ti+1)
            i ← i+1 & continue
    if(queryToken(ti)
        markAsChemical(ti)
    else    continue

    if(i+3<=n)
        queryTokenAndMarkAsChemical(ti)
        queryTokenAndMarkAsChemical(ti+1)
        queryTokenAndMarkAsChemical(ti+2)
        queryTokenAndMarkAsChemical(ti+3)
        if(areChemical(ti,ti+1,ti+2,ti+3))
            merge(ti,ti+1,ti+2,ti+3)
            i ← i+3 & continue
        if(areChemical(ti,ti+1,ti+2))
            merge(ti,ti+1,ti+2)
            i ← i+2 & continue
        if(areChemical(ti,ti+1)
            merge(ti,ti+1)
            i ← i+1 & continue
    if(i+2<=n)
        queryTokenAndMarkAsChemical(ti)
        queryTokenAndMarkAsChemical(ti+1)
        queryTokenAndMarkAsChemical(ti+2)
        if(areChemical(ti,ti+1,ti+2))
            merge(ti,ti+1,ti+2)
            i ← i+2 & continue
            if(areChemical(ti,ti+1)
            merge(ti,ti+1)
            i ← i+1 & continue
    if(i+1<=n)
        queryTokenAndMarkAsChemical(ti)
        queryTokenAndMarkAsChemical(ti+1)
        if(areChemical(ti,ti+1)
            merge(ti,ti+1)
            i ← i+1 & continue
```

Figure 3.5 Algorithm of 4-sliding windows on text tokens.

The proposed method also has a post-processing phase on tokens to merge chemical names identified in the articles which are consecutive, since some of the new chemical words that are constituted of known chemical names are not in the dictionary. In the process of merging tokens, probabilities are multiplied.

3.4 Execution of the Whole System

Our proposed approach takes an article as input and reports a list of tokens in the article as either chemical names or non-chemical names. The indices of each

chemical name in the document is also reported. In addition, we assign a confidence score to each identified chemical name. Figures 3.6 shows the execution process.

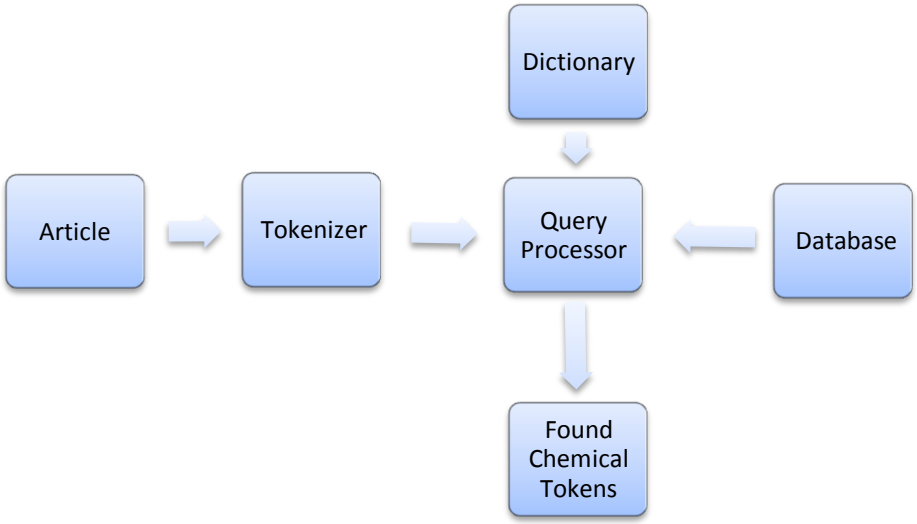

Figure 3.6 Execution process

The performance of the proposed algorithm on a benchmark dataset is discussed in next chapter. Results are compared with the state of the art systems and the scientific information gained from these experiments are presented.

CHAPTER 4

RESULTS AND DISCUSSION

The method proposed is implemented by the Java programming language to experiment on documents containing chemical names. One of the main problems stumbled upon to compare these kind of implementations is the lack of test and training data. To overcome this issue, we experimented our implementation on training, test and development data sets which are provided by BioCreative Conference IV Chemdner task [5]. Details of these datasets are discussed in data set section. In the experimental results section, the performance values and the recall and precision values are given. In the third section, the proposed method is compared with other methods which their results are published on Chemdner task on the same data set used by our method [5].

4.1 Data Sets

The organizers of BioCreative IV Chemdner task released training, development and test data sets for the participants. These data sets contains documents which are selected on various categories such as organic chemistry, multidisciplinary chemistry, endocrinology, chemical engineering, molecular biology, medicinal chemistry, applied chemistry, physical chemistry, polymer science, toxicology and pharmacology [5]. The top 100 journals which have at least 100 articles on these areas are picked to prepare the main corpus. Total of 10000 document are manually annotated and each of the chemical compound mentioned are found human annotators who have background in chemistry [5, 49]. Training data set and

development data set contained manually annotated 3500 document for each of them [49]. The mentions of the chemical compounds in the training and development data set are given in a list called Gold Standard annotation list, so the results of the methods can be compared with this list to compare the success of the methods in terms of precision and recall [5]. Test data set contains 3500 manually annotated documents and other 16500 chemical compound containing documents in total of 20000 documents. The aim of this mixture is to prevent manual intervention during the chemdner evaluation period. The methods aiming to solve problem of chemical name finding in documents are developed implemented by teams attending BioCreative IV Chemdner challenge on this test, training and development data set. To be able to compare with these methods, we implemented and experimented our proposed approach on three data sets.

4.2 Experimental Results

DB2 Express-C is chosen as a DBMS to store huge chemical name and synonym tables on a computer with 4 cores CPU with 8 GBs of memory. The free license of DB2 allows only 2 cores CPU and 4 GBs of memory [44]. The most challenging part of testing the system was to parse and analyze the test data set which contained 20000 articles. While running the implementation on test data set, the system used maximum of 3 GBs of memory; therefore, the method is capable to execute almost on any common system. Also our system has a unique table structure that needs indexes on columns having size of 1024. This is a huge length for column indexing, but the database system we have chosen helped solving this issue.

We ran our implementation on documents supplied by BioCreative Conference IV Chemdner task [5]. Our proposed approach acquired 71% recall and 58% precision for finding chemical names on the development set containing 3500 documents which are manually annotated by humans. On the training set our system resulted in 59% precision and 74 % recall. In the Chemdner task, the test set which contained 20000 documents has been published in September 2013. Our implementation was also tested on this huge test set which is manually annotated by humans. Our system

acquired 70% recall and 62% precision recall for finding compound names on this set. This result proves our system is independent of document domain and easily expandable. The correlation of this observation is shown in Figure 4.1. Since dictionary based methods do not need to be trained, no new training will be required with incoming documents. This is a major drawback in CRF based methods, because dependent on the development set there is always a risk of overfitting on development set.

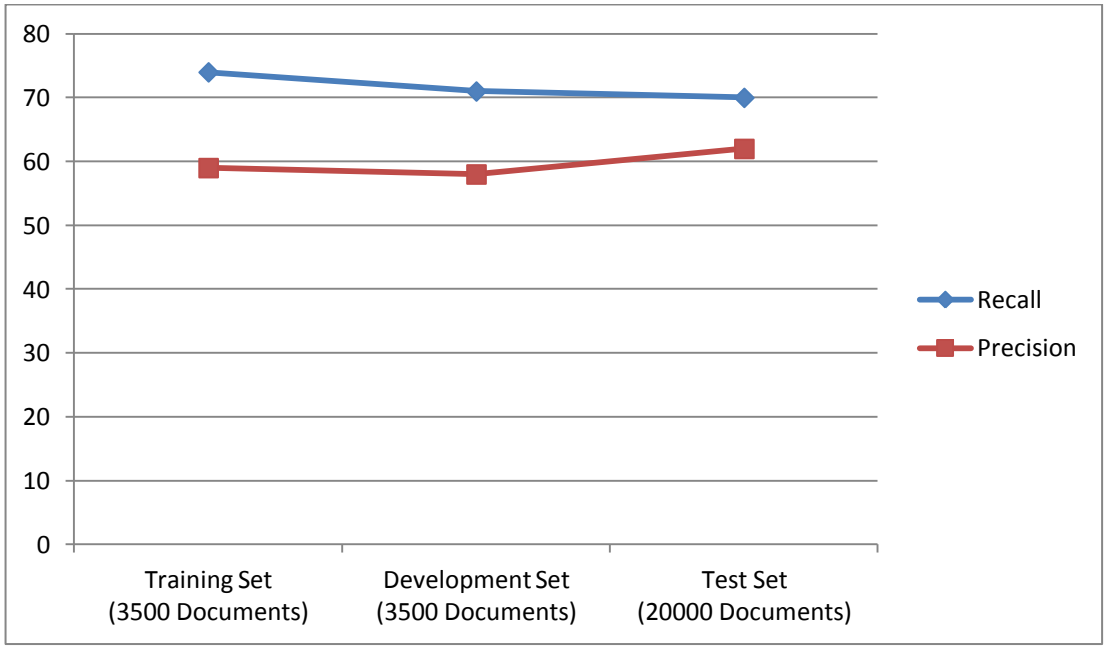

Figure 4.1 Recall and precision on different data sets.

We analyzed the outputs and figured out that that some of our false positive predictions are actually true positives. As an example, the term “thyroid hormone” in some of the articles of the document set is not annotated as a chemical substance, but in another document of the same set, it is annotated as a chemical substance. As another example in one of the articles in the document set, the human annotators did not flag “carboxylic acid” as a chemical substance, although it is a chemical compound. To give another example, the word *steroid* is checked as a chemical substance in some documents and not in some other documents. Enzymes are marked as chemical in some documents while not marked as a chemical in other ones.

Human curators disagree on several issues as mentioned above resulting in a significant decrease in precision and recall.

We have also found other examples in which our proposed method has the capability to find chemical substance names that human annotators did not found in the same document set. These inconsistencies reduced the precision to a certain extent; however, our database querying based solution will still be far from 80%-90% precision even when there is a good inter-rater agreement.

Most of the chemical names in an article can be found by our proposed method in an efficient and elegant way. The running time performance of implementation of the method is of considerable benefit; especially, considering that it can process 20000 articles and the total runtime is below 2 hours. 250 articles per minute can be processed by the on a laptop with using 4 cores of CPU and 3 GBs of memory at maximum.

Some of observations on our database are interesting and worth mentioning about. 33107771 of molecular formulas out of 71604307 do not have any synonyms, in which 46 percent of molecular formulas in the database. 12377292 of molecular formulas have multiple synonyms. The maximum number of synonyms that one molecular formula has is 1615, which the molecular formula with the common name "ethanol". The observations on the database are given in Table 4.1.

Table 4.1: Interesting properties of the chemical compound database

	Molecular Formula	Synonyms
Number rows in database	71602000	76104871
Number of molecular formulas which do not have synonyms	33107771	-
Number of molecular formulas which have more than one synonym	12377292	-

Table 4.1 (continued)

	Molecular Formula	Synonyms
Length of longest name or formula	44	900
Length of shortest name or formula	1	2
Maximum number of synonyms on one formula	-	1615

The effects of different types of querying phases in our query processor and rules module are analyzed by removing the phases one by one and comparing the output of the implementation on development data set. The most effective phase is to query the words directly to database and the least effective is searching the word with its segments above the size of 4 characters. The effects of removing phases on precision and recall are presented in Table 4.2.

Table 4.2: Effects of querying phases on precision and recall

Development Data Set	Recall	Precision
Approach with all phases	59 %	73 %
Without 4 window sliding method	58 %	69 %
Without searching the word with its segments above the size of 4 characters	62 %	67 %
Without single querying of a token	38 %	28 %

Table 4.2 (continued)

Development Data Set	Recall	Precision
Without post processing to merge consecutive chemical words	58 %	72 %

Also some further studies on our system revealed that our method is very suitable for parallelization because of its modular capabilities. By using several computers in a cluster, it has been shown to have the capability of processing more than a thousand articles per minute.

4.3 Comparison with Other Methods

In this section, the proposed method in this thesis is compared with the state of the art systems. Three CRF-based methods and three dictionary-based methods are analyzed. Because the implementation of the method is tested on the BioCreative IV Chemdner challenge development and test data sets, it is appropriate to compare it with methods that are also experimented on these data sets. The experimental results of the methods mentioned in this section are acquired from the results that are provided by Krallinger et al. [5].

The method which has the highest precision and recall on this data set is The WHU-BioNLP CHEMDNER system which mainly uses the CRF model. The experiment results of this method on the development set have 87% precision and 84% recall. The method has a similar preprocessing phase in tokenization of articles like our proposed method. WHU-BioNLP method uses a list of symbols that are used in the tokenizer. The method uses n-gram models on word searching and resulted best on 5-gram models. It mixes this approach with CRF to have better results. Although the method has 84% recall on the development set, it is increased hugely in test set with 89%. This seems to be confusing since the CRF methods needs more data to train itself for more new incoming test data and expected to behave worse on new huge

test datasets [50]. One of the best CRF methods for this problem is Chemistry-specific Features and Heuristics for Developing a CRF-based Chemical Named Entity Recogniser method. In this method a large feature set is used which contains n-grams of parsed words, token properties, ortographic properties and dictionaries. The feature sets is applied with the CRF model and the results on the development set showed 88% precision and 81% recall. It is an interesting observation that this model gave better results on the test set which is 91% precision and 85% recall. This was also an unexpected behavior [51]. The third CRF-based method is called Extended Feature Set for Chemical Named Entity Recognition and Indexing method. The method uses CRF-model with domain independent features. These features are based on surface forms, n-grams, pos tags and token patterns. The method achieved 89% precision and 80% recall on the development dataset and 89% precision and 86% recall on the test set [52]. There is nine other CRF-based models for this problem on BioCreative IV Conference Chemdner task. Recall and precision ranges of all CRF models on test data set are shown on Figure 4.2 [5]. The standard precision mean of the methods is 77.58% and the standard recall mean is 80.33%. The standard deviation of recall is 10.73 and the standard deviation of precision is 10.67.

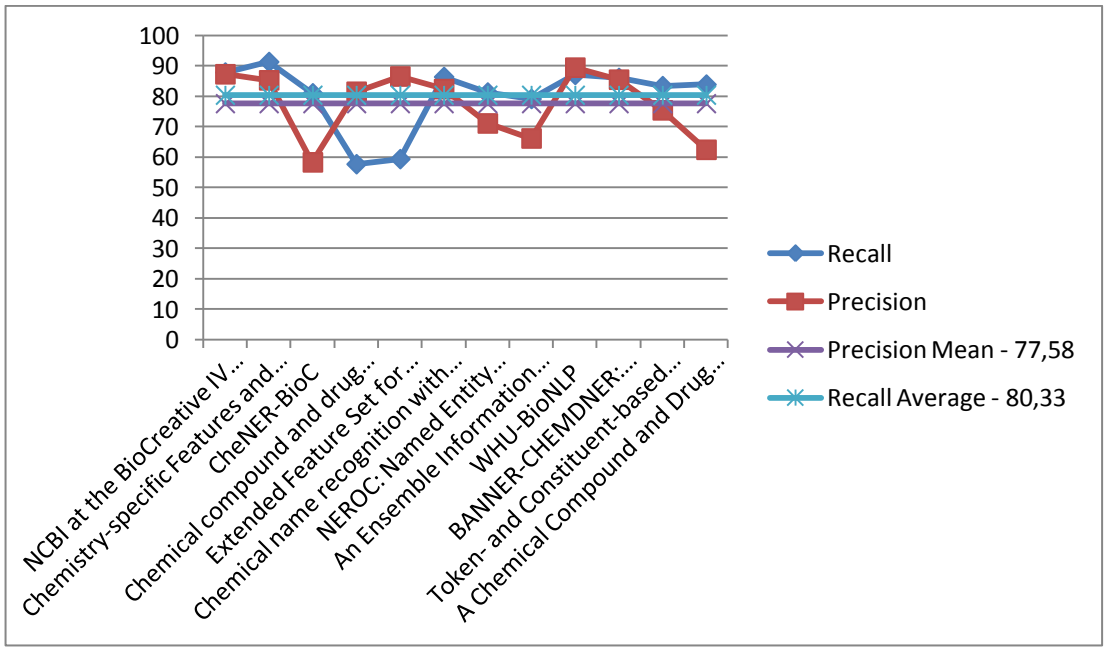

Figure 4.2 Recall and precision of CRF-based methods on the same dataset

LeadMine method is a dictionary-based method which is mentioned in previous chapters. This method resulted 87% precision and 83% recall on the development dataset, 88% precision and 83% recall on the test set [26]. LeadMine method has a more sophisticated pre-processing stage of preparing database, so the results of the method are better than results of our proposed approach. Another method called Chemical Named Entity Recognition with OCMiner which is also mentioned in previous chapters has obtained 82% precision and 72% recall on development set. The method has obtained 85% precision and 71% recall [27] on the test dataset. The third outstanding method is Combining Machine Learning with Dictionary Lookup method. This method combines CRF-model with dictionary lookup. The method uses features like word stemming, part of speech tags and other word specific morphological features [52]. The method obtained 82% precision and 72% recall on the development dataset. On the test dataset this method obtained 73% precision and 76% recall. There is three other dictionary-based models for this problem on BioCreative IV Chemdner task. Recall and precision ranges of all dictionary-based methods on the test dataset are shown on Figure 4.3 [5]. The standard precision mean of the methods is 70.07% and the standard recall mean is 74.13%. The standard deviation of recall is 11.21 and the standard deviation of precision is 10.67.

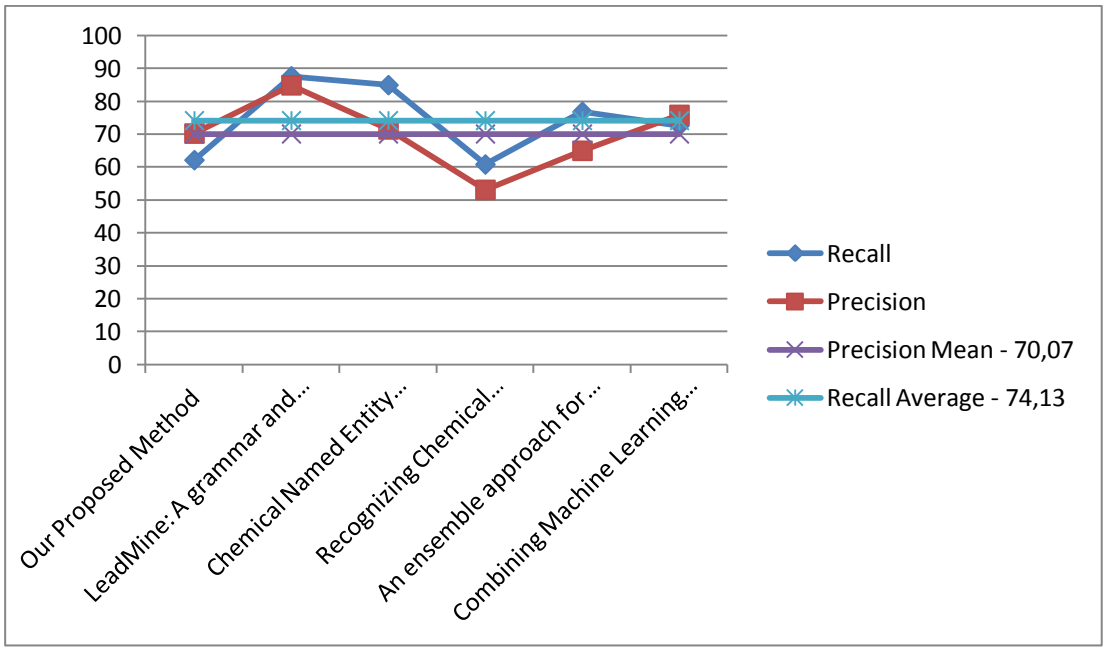

Figure 4.3 Recall and precision of dictionary-based methods on the same dataset

Analyzing these results shows that CRF-based methods are more convenient for this problem. Their recall and precision average is far higher than dictionary-based model results. The recall and precision difference of CRF-based methods are unstable on different dataset compared to dictionary-based models.

4.4 Discussion

Based on the experimental results, the CRF-based models are much more effective on the chemical name searching problem. But according to CAS Registry Fact Sheet [53], more than 15000 new substances are added to the CAS Registry every day. This huge value is a potential problem for not only dictionary-based but also to CRF-based models. Periodical training of CRF-based models is needed in order to handle new incoming substances. The training of CRF-based models is difficult because a manually annotated document input must be prepared. Also dictionary based models have to process these new substance names and need to add them to their dictionaries. Our proposed approach has benefits resolving these issues, because it is capable of constructing a database containing than more than 145 million of entries in less than two days.

CHAPTER 5

CONCLUSION

In conclusion, we presented a chemical and drug name recognition method which attains a very good running-time performance for chemical named entity recognition in articles. However, our approach may fall short in terms of precision compared to other competing methods in the BioCreative Challenge. Our analysis shows that our method can be improved in terms of precision. The PUG database, which is the most critical component of the proposed approach, needs to be preprocessed before creating the database.

Testing of these systems is another challenge, because creating test data for name mentioning in documents needs human interaction, which can cause ambiguous results. Human curators commonly annotate chemical names in different manners. It is impossible to create any model to handle erroneous inputs and it leads to decrease in precision and recall.

By 2012, MedLine has more than 20 million articles [54]. All these 20 million MedLine articles can be processed by our approach in 22 days on a regular laptop and our method has the ability to find the chemical names in these documents. In addition to this, our approach has the capability of providing synonyms of chemical substance in addition to finding them in documents. It is useful for search engines since search engine queries tries to give all the possible relevant result of any query.

To sum up, our proposed method for the chemical compound and drug name problem is an innovative and flexible approach. Its modular design allows

combination with different techniques and models, which allows our method for improvement. The database and dictionaries provided are comprehensive and can be used in other scientific areas. The querying methods are capable of finding chemical name probabilities which can offer an input to other searching models. Also integrating CRF-based techniques with our proposed approach can lead to better results. Our proposed method can be used as a token filter before passing the chemical names to CRF-based methods.

REFERENCES

1. CAS Registry and CAS Registry Number FAQs. http://www.cas.org/content/chemical-substances/faqs. Accessed: 2014-07-29.

2. CAS Registry Keeps Pace with Rapid Growth of Chemical Research, Registers 60 Millionth Substance. http://www.cas.org/news/media-releases/60-millionth-substance. Accessed: 2014-07-29.

3. M. Vazquez, M. Krallinger, F. Leitner, A. Valencia. Text Mining for Drugs and Chemical Compounds: Methods, Tools and Applications. *Molecular Informatics*, 30(6-7), pages 506-519, 2011.

4. P. Corbett, C. Batchelor, S. Teufel. Annotation of chemical named entities. *BioNLP 2007: Biological, translational, and clinical language processing*, pages 57-64, 2007.

5. M. Krallinger, F. Leitner, O. Rabal, M. Vazquez, J. Oyarzabal, A. Valencia. Overview of the chemical compound and drug name recognition (CHEMDNER) task. *BioCreative Challenge Evaluation Workshop*, Volume 2, 2013.

6. Chemical Compound. http://en.wikipedia.org/wiki/Chemical_compound . Accessed: 2014-07-29.

7. Drug. http://dictionary.reference.com/browse/drug . Accessed: 2014-07-29.

8. R. S. Petrucci, W. S. Harwood, F. G. Herring. General Chemistry: Principles and Modern Applications (8th edition). Book published by Prentice-Hall, 2002.

9. Chemical Formula. http://en.wikipedia.org/wiki/Chemical_formula. Accessed: 2014-07-29.

10. D. I. Cooke-Fox, G. H. Kirby, J. D. Rayner. Computer translation of IUPAC systematic organic chemical nomenclature. 3. Syntax analysis and semantic processing. *Journal of Chemical Information and Modeling*, 29 (2), pages 112 – 118, 1989.

11. Chemical Name. http://en.wikipedia.org/wiki/Chemical_name . Accessed: 2014-07-29.

12. J. Rigaudy, S.P. Klesney, Nomenclature of Organic Chemistry. Book published by Pergamon Press, 1979

13. R. Panico, W.H. Powell, J.C. Richer. A Guide to IUPAC Nomenclature of Organic Compounds. Book published by IUPAC/Blackwell Science, 2004.

14. N. G. Connelly, Nomenclature of Inorganic Chemistry IUPAC Recommendations 2005 ed. Book published by RSC Publishing, 2005.

15. R. Klinger, C. Kolarik, J. Fluck, M. Hofmann-Apitius, C.M. Friedrich. Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, 24(13), pages 268-i276, 2008.

16. C. Ata and T. Can. DBCHEM: A Database Query Based Solution for the Chemical Compound and Drug Name Recognition Task. *BioCreative Challenge Evaluation Workshop*, Volume 2, 2013.

17. Y. Garten, R.B. Altman. Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text. *BMC Bioinformatics*, Suppl 2:S6, 2009.

18. D. Cheng, C. Knox, N. Young, P. Stothard, S. Damaraju, D.S. Wishart. PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Res 36 (suppl. 2)*, 2008.

19. D.S. Wishart, C. Knox, A.C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, M. Hassanali. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res*, 36(Database issue):D901-6, 2008.

20. O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Oxford JournalsScience & Mathematics Nucleic Acids Res* Volume 32, Issue suppl 1, pages 267-270, 2004.

21. Y. Wang, J. Xiao, T.O. Suzek, J. Zhang, J. Wang, Z. Zhou, L. Han, K. Karapetyan, S. Dracheva, B. A. Shoemaker, E. Bolton, A. Gindulyte, S.H. Bryant. PubChem's BioAssay Database. *Nucleic Acids Res*, 40(Database issue), 2011.

22. P. de Matos, M. Ennis, M. Bidden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj, M. Ashburner, K. Degtyarenko. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res*, Volume 36, 2008.

23. W. J. Rogers, A. R. Aronson, Technical Report 2008. http://skr.nlm.nih.gov/papers/references/filtering07.pdf. Accessed: 2014-07-30.

24. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, vol. 10, no. 8, pages 707–710, 1966.

25. Levenshtein Distance. http://en.wikipedia.org/wiki/Levenshtein_distance. Accessed: 2014-07-30.

26. D.M. Lowe, R.A. Sayle. LeadMine: A grammar and dictionary driven approach to chemical entity recognition. *BioCreative Challenge Evaluation Workshop* vol. 2, 2013.

27. M. Irmer, C. Bobach, T. Bohme, U. Laube, A. Puschel, L. Weber. Chemical Named Entity Recognition with OCMiner. *BioCreative Challenge Evaluation Workshop* vol. 2, 2013.

28. Registry File Basic Name Segment Dictionary. Published by Chemical Abstracts Service, 1993.

29. W.J. Wilbur, G.F. Hazard Jr, G. Divita, J.G. Mork, A.R. Aronson, A.C. Browne. Analysis of biomedical text for chemical names: a comparison of three methods. *Proceedings of the AMIA Symposium*, pages 176 – 180, 1999.

30. J.D. Lafferty, A. McCallum, F.C.N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282-289, 2001.

31. B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107, 2004.

32. F. Sha, F. Pereira. Shallow Parsing with Conditional Random Fields. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Volume 1, 2003.

33. Conditional Random Field. http://en.wikipedia.org/wiki/Conditional_random_field. Accessed: 2014-07-30.

34. X. He, R.S. Zemel, M.A. Carreira-Perpinan. Multiscale Conditional Random Fields for Image Labeling. Computer Vision and Pattern Recognition, *Proceedings of the 2004 IEEE Computer Society Conference* on Volume 2, pages 695-702, 2004.

35. A.A. Markov. Theory of Algorithms, 1954.

36. R. Klinger, C.M. Friedrich, J. Fluck, M. Hofmann-Apitius. Named Entity Recognition with Combinations of Conditional Random Fields. *Proceedings*

of the Second BioCreative Challenge Evaluation Workshop, pages 89-92, 2007.

37. BioCreative Conference. http://www.biocreative.org/about . Accessed: 2014-08-02.

38. PubChem Power User Gateway (PUG): https://pubchem.ncbi.nlm.nih.gov/pug/pughelp.html. Accessed: 2014-07-31.

39. PubChem Compound Identifier (CID): https://pubchem.ncbi.nlm.nih.gov/search/help_search.html#Cid. Accessed: 2014-07-31.

40. MySQL Database. http://en.wikipedia.org/wiki/MySQL. Accessed: 2014-07-31.

41. BaseX Xml Database. http://en.wikipedia.org/wiki/BaseX. Accessed: 2014-07-31.

42. PostreSQL: http://en.wikipedia.org/wiki/PostgreSQL. Accessed: 2014-07-31.

43. eXist: http://en.wikipedia.org/wiki/EXist . Accessed: 2014-07-31.

44. DB2 Express-C: http://en.wikipedia.org/wiki/IBM_DB2_Express-C. Accessed: 2014-07-31.

45. NI2-Webster's New International Dictionary, Second Edition http://www.puzzlers.org/dokuwiki/doku.php?id=solving:wordlists:about:start. Accessed: 2014-07-31.

46. Tokenization. http://en.wikipedia.org/wiki/Tokenization. Accessed: 2014-08-01.

47. R.D. Smith, Rochester. Distinct word length frequencies: distributions and symbol entropies. *Glottometrics Journal* 23, 7-22, 2012.

48. Ronald M. Kaplan. A Method for Tokenizing Text. Inquiries into Words, Constraints and Contexts.

http://web.stanford.edu/group/cslipublications/cslipublications/koskenniemi-festschrift/ . Accessed: 2014-08-03.

49. BioCreative IV Chemdner Task Frequently Asked Questions. http://www.biocreative.org/tasks/biocreative-iv/chemdner-task-2-faq . Accessed: 2014-08-10.

50. Y.Lu, X. Yao, X. Wei, D. Ji. WHU-BioNLP CHEMDNER System with Mixed Conditional Random Fields and Word Clustering. *BioCreative Challenge Evaluation Workshop*, Volume 2, 2013.

51. R.T. Batista-Navarro, R. Rak, S. Ananiadou. Chemistry-specific Features and Heuristics for Developing a CRF-based Chemical Named Entity Recogniser. *BioCreative Challenge Evaluation Workshop*, Volume 2, 2013.

52. L. Li, R. Guo, S. Liu, P. Zhang, T. Zheng, D. Huang, H. Zhou. Combining Machine Learning with Dictionary Lookup for Chemical Compound and Drug Name Recognition Task. *BioCreative Challenge Evaluation Workshop*, Volume 2, 2013.

53. CAS REGISTRY Fact Sheet. http://www.cas.org/about-cas/cas-fact-sheets/registry-fact-sheet. Accessed: 2014-08-04.

54. MEDLINE Number of Citations to English Language Articles, Number of Citations Containing Abstracts. http://www.nlm.nih.gov/bsd/medline_lang_distr.html. Accessed: 2014-08-03.