DBCHEM: A Database Query Based Solution for the Chemical Compound and Drug Name Recognition Task

Caglar Ata\textsuperscript{1} and Tolga Can\textsuperscript{2}

Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

Abstract. We propose a method, named DBCHEM, based on database queries for the chemical compound and drug name recognition task of the BioCreative IV challenge. We prepared a database with 145 million entries containing compound and drug names, their synonyms, and molecular formulas. PubChem Power User Gateway (PUG) system is used to construct the database. Candidate chemical and drug names are identified by using an English dictionary as a list of stop words. All candidates are queried in the compound database. We integrated a small number of heuristic rules into this query based approach. DBCHEM attained 58\% precision and 71\% recall on the development set with a total running time of 14 minutes for 3500 articles.

Keywords. PubChem; Database Management Systems; Database query; Compound and drug names

1 Introduction

Today, one of the challenges in computer science is extraction of structured information in free text, such as research publications and web pages. Researchers need to find relevant information in a fast and correct way within these huge amounts of data. There have been numerous studies in document indexing and categorization but there are few which specifically focus on chemical and drug names [1]. Chemical and drug names are complex in nature and ambiguous as several different names can represent the same chemical compound. Molecular formulas are complex also as too many different variants can refer to the same molecular formula [1]. Rules and regular expressions for constructing and validating molecular formulas are difficult to derive. Parsing documents containing molecular formulas is also difficult since researches show that 90\% percent of formulas contain no whitespace and 22\% of these have or adjacent to hyphens or dashes [2].
We present a solution based on database queries to overcome those issues discussed above. We use a database with 145 million entries containing compound and drug names, their synonyms, and molecular formulas. We use an extensive English dictionary as a stop word list and identify tokens which can be queried in the compound database. We also use a small number of rules for post processing the query results. DBCHEM attained 58% precision and 71% recall on the development set with a total running time of 14 minutes for 3500 articles. The details of the method and the results are given in the following sections.

2 Materials and Methods

The Chemical Compound Database
PubChem Power User Gateway (PUG) system provides a web based interface to query chemical names, formulas, assays, and substances. We wrote a script to download molecular formulas and chemical names from PUG as XML files. We downloaded PubChem Compound Identifiers (CID) starting from 1 to 71604307 and containing 76104871 synonyms for these compounds in less than 24 hours on a 15 GBs of disc space. We parsed these XML files and created two tables, one for synonyms and one for molecular formulas, each one containing more than 70 million entries.

We used DB2 Express-C as a database management system (DBMS) which is free to develop, deploy and distribute [3], to store these huge tables on an HP Pavilion dv7 laptop with Intel i5 CPU with 8 GBs memory, allowing DB2 to use maximum 2 cores and 4 GBs of memory at most. We investigated using XML DBMSs and open source DBMSs such as MySQL, eXist-db, and BaseX; however, these DBMSs were not capable of storing and querying huge tables and large column indexes effectively. The total system (including the database) used less than 3 GBs of memory while parsing the 20000 articles in the test dataset; therefore, DBCHEM is able run almost on any system containing portable PCs. In addition, DB2 was capable of indexing strings containing 1024 characters, which is a desired property, since chemical names can be very long in practice. (PubChem contains synonyms as long as 900 characters in length). Some interesting statistics on this constructed database are presented in the Results and Discussion section.

Tokenization of Articles
We used the Stanford Penn TreeBank Tokenizer (PTBTokenizer) to tokenize each article. PTBTokenizer tokenizes traditional English sentences
very good, but did not perform well on articles containing complex chemical names. PTBTokenizer parses complex molecular formulas into several tokens, since molecular formulas may contain parseable characters such as dashes and colons. Therefore, we analyzed several chemical names from different articles and implemented a preprocessing phase before submitting the article to PTBTokenizer. Our algorithm replaces each of the characters in a character set containing 48 characters (dashes, colons, and semicolons without whitespaces at ends, curly brackets, etc.) with a unique string of 18 characters. To illustrate, we convert \textit{1,2-propadiene} to \textit{1semicolonsemicolon2dashdashdashdashdapropadiene} and pass this string to PTBTokenizer. We convert these strings back to their original form after PTBTokenizer identifies the tokens.

\textit{Preparing a Chemical-Free English Word Dictionary}

We prepared a chemical-free English word dictionary by querying all words in NI-Webster’s New International Dictionary [4] in our database. The intersected word list is examined and all the chemicals in this intersected list are removed from the English-word dictionary. All of the English words which are found in the chemical database are also removed as they may be valid chemical names. We used this chemical-free English word dictionary as list of stop words to filter-out English words in an article before querying the identified tokens in our database. The main purpose of this filter is to improve running time performance of the system.

\textit{The Query Algorithm}

We analyzed several chemical and drug names and found that most of the chemical and drug names are at most 4 words length. We picked a window size of 2 to 4 words and by sliding this window over the tokens identified in the article, we queried each window in the database. Query results are evaluated similar to the individually queried tokens found in the chemical database, i.e., they are marked as identified chemical names. If the whole window exists in the database as is, we assign a score of 1.0 to these tokens (scores are between 0.0 and 1.0). If they are found in the database as a result of a “like” query, a score between 0.8-0.9 is assigned based on the specificity of the “like” query. If the token contains all letters in the uppercase, we query the token in the molecular formula database as well. After analyzing the whole article by windows of size 2 to 4, we query the individual tokens of a window in the database, for windows with no query results. We also have a post-processing phase to merge consecutive chemical names identified in the article abstract.
3 Results and Discussion

Our system attained 58% precision and 71% recall for the CEM (Chemical entity mention recognition sub-task) [5] and 58% precision and 73% recall for the CDI (Chemical document indexing sub-task) [5] on the development set provided. The system is able to process 250 articles per minute on a laptop with Intel i5 processor using less than 3 GBs of memory in total.

A close up analysis on the results shows that some of our false positive predictions are actually true positives. For example, the term “thyroid hormone” in article 23258742 of the training set is not marked as a chemical name, but in article 23594789 of the same training set, it is marked as a chemical entity. In article number 23177256 in the training set, the gold standard annotations did not mark “carboxylic acid” as a chemical, although it is a chemical entity. As another example, the word steroid is marked as a chemical in some documents and not in other documents. We have also identified some examples in which DBCHEM is capable of finding chemical mentions that are not found by human annotators. These inconsistencies decrease the reported precision to a certain extent; however, we do not claim that a good inter-rater agreement would increase the precision significantly to 80%-90% levels.

There are also some interesting observations that worth mentioning about the PubChem chemical name/synonym and molecular formula databases we constructed. Out of 71604307 molecular formulas, 33107771 of them do not have any names, which is 46% of the whole molecular formula database. 32% of the named molecular formulas have more than one name.

In conclusion, DBCHEM is able to find most of the chemical names in an article abstract in a simple and efficient way. The running time performance of DBCHEM is of considerable benefit; especially, considering the postponement requests on the prediction deadline made by machine learning based solutions DBCHEM can process 20000 articles in less than 2 hours. As of 2012, MedLine contains over 20 million articles [6]. Our system is able to process all MedLine articles in 22 days on a regular laptop for both the CEM and CDI tasks.

4 Acknowledgments

We thank Osman Karslioglu for helpful discussions on standards and rules in chemical compound naming.
REFERENCES