NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

MASTER'S PROGRAM
"DATA SCIENCE AND INFORMATION TECHNOLOGIES (DSIT)"
SPECIALIZATION: BIOINFORMATICS - BIOMEDICAL DATA SCIENCE

MSc THESIS

Chemical text and association rule mining to facilitate the study of metabolic processes in hyperthermophilic microorganisms.

Ourania E. Theologi

Supervisor: **Theodoros Dalamagas,** Research Director, ATHENA RC

Examination Comittee: **Theodore Dalamagas,** Research Director, ATHENA RC
Evangelos Pafilis, Assistant Researcher, HCMR
Ioannis Emiris, President, ATHENA RC & Professor, NKUA

ATHENS

OCTOBER 2021

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΙΔΡΥΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
“DATA SCIENCE AND INFORMATION TECHNOLOGIES (DSIT)”
SPECIALIZATION: BIOINFORMATICS - BIOMEDICAL DATA SCIENCE

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τεχνικές εξόρυξης κειμένου και κανόνων συσχέτισης χημικών οντοτήτων για την υποστήριξη μελέτης των μεταβολικών διεργασιών υπερθερμόφιλων μικροοργανισμών.

Ουρανία Ε. Θεολογή

Επιβλέπων: **Θεόδωρος Δαλαμάγκας,** Διευθυντής Ερευνών, ΕΚ ΑΘΗΝΑ

Τριμελής Επιτροπή: **Θεόδωρος Δαλαμάγκας,** Διευθυντής Ερευνών, ΕΚ ΑΘΗΝΑ
Ευάγγελος Παφίλης, Ερευνητής, ΕΛΚΕΘΕ
Ιωάννης Εμίρης, Πρόεδρος, ΕΚ ΑΘΗΝΑ & Καθηγητής, ΕΚΠΑ

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2021

MSc THESIS

Chemical text and association rule mining to facilitate the study of metabolic processes in hyperthermophilic microorganisms.

Ourania E. Theologi
S.N.: DS2190007

SUPERVISOR: **Theodoros Dalamagas,** Research Director, ATHENA RC

EXAMINATION COMITTEE: **Theodore Dalamagas,** Research Director, ATHENA RC
Evangelos Pafilis, Assistant Researcher, HCMR
Ioannis Emiris, President, ATHENA RC & Professor, NKUA

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τεχνικές εξόρυξης κειμένου και κανόνων συσχέτισης χημικών οντοτήτων για την υποστήριξη μελέτης των μεταβολικών διεργασιών υπερθερμόφιλων μικροοργανισμών.

Ουρανία Ε. Θεολογή
Α.Μ.: DS2190007

ΕΠΙΒΛΕΠΩΝ: **Θεόδωρος Δαλαμάγκας,** Διευθυντής Ερευνών, ΕΚ ΑΘΗΝΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Θεόδωρος Δαλαμάγκας,** Διευθυντής Ερευνών, ΕΚ ΑΘΗΝΑ
Ευάγγελος Παφίλης, Ερευνητής, ΕΛΚΕΘΕ
Ιωάννης Εμίρης, Πρόεδρος, ΕΚ ΑΘΗΝΑ & Καθηγητής, ΕΚΠΑ

ABSTRACT

The study of the metabolism of hyperthermophilic microorganisms is of great industrial importance since these organisms' enzymes catalyze reactions at elevated temperatures. Various hyperthermophiles of archaea (e.g., *Thermococcus, Pyrococcus*, etc.) and bacteria (e.g., *Caldicellulosiruptor, Thermotoga*, etc.) are established as laboratory models; their metabolic pathways undergo genetic engineering aiming at optimal production of fuels and chemicals at elevated temperatures in industrial scale.

This thesis aims to ease the literature study of the biological processes (and related metabolites) of microorganisms of interest. The key contribution of the thesis is a framework to (a) identify mentions of chemical entities (metabolites, substrates, enzyme cofactors, etc.) in paper abstracts, and (b) retrieve co-occurrence associations between microorganisms and chemical entities, and among chemical entities.

First, a chemical Named Entity Recogniser (NER), built upon the spaCy [50] (`https://spacy.io/`) library and trained on the CHEMDNER corpus [36], was developed in order to extract mentions of chemicals. A set of 9 hyperthermophile species (*Thermococcus kodakarensis, Pyrococcus furiosus, Metallosphaera sedula, Thermotoga maritima, Caldicellulosiruptor bescii, Sulfolobus solfataricus, Thermus thermophilus, Thermoanaerobacter mathranii, Caldicellulosiruptor hydrothermalis*) were used to demonstrate the chemical NER's functionality and applicability using literature (i.e PubMed abstracts) retrieved via the ORGANISMS web application (`http://organisms.jensenlab.org`).

Then, the chemical entities extracted from the aforementioned literature were further analysed for the appearance of frequent patterns and the co-occurrences among them by generating association rules between frequent itemsets (association rule mining).

As a result of our work, the association of carbohydrates to *C. bescii* and of copper and *T .thermophilus* to heme were suggested from this study. Manual curation of the literature showed that indeed Carbohydrate metabolism has been extensively studied in *C. bescii* for the production of ethanol from lignocellulosic biomass. Similarly, extensive studies of *T. thermophilus* report that the heme–copper oxygen reductases are able to catalyze the reduction of nitric oxide to nitrous oxide under reducing anaerobic conditions.

Beyond the hyperthermophilic microorganisms study, the presented methods could be applied to any microorganism specific abstract collection. The chemical NER facilitates the identification of chemical entities in text. Furthermore, association rule mining provides co-occurrence associations between microorganisms and chemical entities and between chemical entities in a selected abstract collection.

SUBJECT AREA: Text mining

KEYWORDS: extremely thermophilic microorganisms, chemical named entity recognition, association rule mining, microbial metabolic processes

ΠΕΡΙΛΗΨΗ

Η μελέτη του μεταβολισμού των υπερθερμόφιλων μικροοργανισμών έχει μεγάλη σημασία για τη βιομηχανία διότι τα ένζυμα αυτών των μικροοργανισμών καταλύουν αντιδράσεις σε υψηλές θερμοκρασίες. Διάφορα είδη υπερθερμόφιλων μικροοργανισμών από γένη αρχαίων (π.χ., *Thermococcus, Pyrococcus*, κλπ.) και βακτηρίων (π.χ., *Caldicellulosiruptor, Thermotoga*, κλπ.) αποτελούν εργαστηριακά μοντέλα, των οποίων τα μεταβολικά μονοπάτια έχουν τροποποιηθεί γενετικά με στόχο την παραγωγή καυσίμων και χημικών σε υψηλές θερμοκρασίες σε βιομηχανική κλίμακα.

Η παρούσα εργασία έχει ως στόχο να διευκολύνει μελέτη των βιολογικών διεργασιών (και των μεταβολιτών που σχετίζονται με αυτές) των υπερθερμόφιλων μικροοργανισμών. Η πορεία που ακολουθήθηκε ήταν να ανιχνευτούν χημικά σε περιλήψεις από επιστημονικά άρθρα που αναφέρουν τους μικροοργανισμούς που μας ενδιαφέρουν και στη συνέχεια να μελετηθούν οι συσχετίσεις συναναφορας μεταξύ των μικροοργανισμών αυτών και των χημικών στοιχείων, καθώς επίσης και οι συσχετίσεις συνύπαρξης μεταξύ των χημικών.

Δημιουργήσαμε ένα εργαλείο αναγνώρισης χημικών οντοτήτων με τη χρήση του πακέτου spaCy (https://spacy.io/) της Python, το οποίο εκπαιδεύσαμε με τη χρήση της συλλογής CHEMDNER, ώστε να εντοπίζει αναφορές ονομάτων μικρών χημικών μορίων σε επιστημονικά κείμενα. Επιλέξαμε να μελετήσουμε εννέα είδη υπερθερμόφιλων μικροοργανισμών (*Thermococcus kodakarensis, Pyrococcus furiosus, Metallosphaera sedula, Thermotoga maritima, Caldicellulosiruptor bescii, Sulfolobus solfataricus, Thermus thermophilus, Thermoanaerobacter mathranii, Caldicellulosiruptor hydrothermalis*). Τα επιστημονικά κείμενα (περιλήψεις άρθρων) πάνω στα οποία έγινε η ανίχνευση χημικών με το μοντέλο που δημιουργήσαμε προήλθαν με τη βοήθεια της εφαρμογής ORGANISMS (http://organisms.jensenlab.org). Τα χημικά στοιχεία που εντοπίστηκαν να αναφέρονται, αναλύθηκαν περεταίρω για την παρουσία συχνών μοτίβων και για την παρουσία ταυτόχρονης συνύπαρξης των μοτίβων αυτών με στόχο να εντοπιστούν οι κανόνες συσχετίσεων μεταξύ των συχνών μοτίβων (εξόρυξη κανόνων συσχετίσεων).

Η συσχέτιση των υδατανθράκων με το είδος *C. bescii* και το χαλκό και η συσχέτιση του είδους *T .thermophilus* με την αίμη προτάθηκε από αυτή τη μελέτη. Περαιτέρω μελέτη της βιβλιογραφίας έδειξε ότι πράγματι ο μεταβολισμός των υδατανθράκων έχει μελετηθεί εκτενώς στο είδος *C. bescii* με στόχο την παραγωγή αιθανόλης από λιγνοκυτταρινική βιομάζα. Ομοίως, πολυάριθμα άρθρα για το είδος *T. thermophilus* αναφέρουν ότι οι αναγωγάσες οξέος -χαλκού οξυγόνου καταλύουν τη αναγωγή του οξειδίου του αζώτου σε οξείδιο του αζώτου υπό μειωμένες αναερόβιες συνθήκες.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εξαγωγή δεδομένων από κείμενο

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: υπερθερμόφιλοι μικροοργανισμοί, αναγνώριση ονομασίας χημικής οντότητας, εξόριξη κανόνων σύνδεσης, μεταβολικές διεργασίες

Στη Μαριάνθη

ACKNOWLEDGEMENTS

This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No 241 (PREGO project).

Το έργο χρηματοδοτήθηκε από το Ελληνικό Ίδρυμα Έρευνας και Καινοτομίας (ΕΛ.ΙΔ.Ε.Κ.) και από τη Γενική Γραμματεία Έρευνας και Τεχνολογίας (ΓΓΕΤ), με αρ. Σύμβασης Έργου 241 (πρόγραμμα PREGO).

ΕΛ.Ι.Δ.Ε.Κ.
Ελληνικό Ίδρυμα Έρευνας & Καινοτομίας

ΓΓΕΤ
ΓΕΝΙΚΗ ΓΡΑΜΜΑΤΕΙΑ
ΕΡΕΥΝΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

Για τη διεκπεραίωση της παρούσας Διπλωματικής Εργασίας, θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές Θοδωρή Δαλαμάγκα και Ευάγγελο Παφίλη καθώς επίσης και τους διδακτορικούς φοιτητές Σάββα Παραγκαμιάν και Χάρη Ζαφειρόπουλο για τη συνεργασία και την πολύτιμη βοήθειά τους στην ολοκλήρωσή της.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. EXTENDED SYNOPSIS

Extremely thermophilic microorganisms are organisms highly adapted to high temperatures considered as “extreme” by human perception. These conditions are the norm under which these organisms are able to metabolically and biochemically operate. The study of the metabolism of hyperthermophilic microorganisms is of great industrial importance since these organisms’ enzymes are able to catalyze reactions at elevated temperatures. Species from various extreme thermophilic genera of archaea (e.g., *Thermococcus*, *Pyrococcus*, etc.) and bacteria (e.g., *Caldicellulosiruptor*, *Thermotoga*, etc.) are established as laboratory models. These models’ metabolic pathways undergo genetic modifications using genetic engineering techniques aiming at optimal production of fuels and chemicals at elevated temperatures in industrial scale.

This thesis aims to ease and accelerate the literature study of the biological processes and related metabolites of the hyperthermophilic microorganisms of industrial interest. A first challenge here is to identify the mentions of chemical entities (metabolites, substrates, enzyme cofactors, etc.) in a collection of abstracts that refer to the hyperthermophilic microorganisms of interest. A second challenge is to apply association rule mining techniques in order to retrieve co-occurrence associations between these microorganism mentions and the chemical entities identified in the abstracts and also to retrieve co-occurrence associations between the identified chemical entities. The retrieved associations from the selected abstracts give an overview indicating the most studied subjects referring to chemicals regarding the selected hypertherophilic microorganisms.

Named entity recognition (NER) refers to identifying mentions of entities of specific types in natural language texts. The main focus of NER in biomedical text, until recently, was placed on the identification of gene and protein names in scientific text. The interest in other types of entities including chemicals and drugs is actively developed. In this project the main focus is small chemical molecule NER (”chemical NER” in brief). Various systems addressing chemical NER have already been developed. The available tools follow various approaches such as using dictionaries or rules or machine-learning methods or a combination of the above, in order to identify chemical mentions in scientific text.

The chemical NER task had to tackle various problems. Until recently, the main stepping stone for the development of chemical NER tools was the limited number of available training resources and the lack of homogeneity of resources for training and evaluation systems. Therefore the BioCreative IV CHEMDNER task was organized with the aim of evaluating the effectiveness of using automated tools for the identification of mentions of chemical compounds and drugs in scientific documents, while at the same time promoting the development of such tools. During the BioCreative IV CHEMDNER task a newly annotated corpus (CHEMDNER corpus) was created, composed of ten thousand abstracts from journals in different sub-fields of chemistry, and containing over 84 thousand entity mentions organized in seven different classes - systematic, trivial, abbreviation, family, identifiers, formula and multiple - plus a small number of unclassified annotations.

In this project the first step was to develop a novel chemical Named Entity Recogniser. The chemical Named Entity Recogniser was built upon the spaCy (`https://spacy.io/`) library. spaCy and scispaCy (`https://spacy.io/universe/project/scispacy`) are state-of-the-art python packages that offer various features for biomedical NLP. spaCy is an open-source Python library designed for building applications for advanced Natural Language Processing (NLP) and scispaCy is a specialized NLP Python package for processing biomedical text that uses as its basis the spaCy library. spaCy is an easy to deploy system

that now starts to gain popularity among the biomedical domain. The chemical NER tool was trained using the training and development datasets of the CHEMDNER corpus. The evaluation was performed on the evaluation set of the CHEMDNER corpus and the performance was compared to the 2016 Biocreative task about chemical NER. The developed chemical NER scored the 10th position, with performance scores (precision: 84.71%, recall 77.07%, f1-score: 80.71%) close to the best scoring tools (precision: 89.09%, recall 85.75%, f1-score: 87.39%).

The developed chemical Named Entity Recogniser was applied on a selection of abstracts specific to a group of hyperthermophilic microorganisms. The literature of the following microorganisms that have been extensively studied for applications in industrial scale: *Thermococcus kodakarensis, Pyrococcus furiosus, Metallosphaera sedula, Thermotoga maritima, Caldicellulosiruptor bescii, Sulfolobus solfataricus, Thermus thermophilus, Thermoanaerobacter mathranii* and *Caldicellulosiruptor hydrothermalis* was retrieved via the ORGANISMS `http://organisms.jensenlab.org`) web source. The novel chemical Named Entity Recogniser was used in order to identify mentions of chemicals in these abstracts.

The chemical entities found to be mentioned in the abstracts were further analysed for the appearance of patterns that appear frequently (frequent pattern mining) in the dataset and their co-occurrences. First the frequent itemsets were detected and then strong association rules were generated from the frequent itemsets using the fpmax algorithm. 320 frequent itemsets were identified and in total 432 association rules were generated. The association rules whose lift value is more than 5 were selected for further investigation in the literature.

More specifically in the case study of the selected hyperthermophilic species two examples of interesting associations in the examined literature are: the association of carbohydrates to *C. bescii* and the association of copper and *T .thermophilus* to heme. Further study of the available literature reveals that carbohydrate metabolism has been extensively studied in *C. bescii* for the production of ethanol from lignocellulosic biomass. Extensive studies of *T. thermophilus* show that the heme–copper oxygen reductases are able to catalyze the reduction of nitric oxide to nitrous oxide under reducing anaerobic conditions.

Beyond the hyperthermophilic microorganisms case study, the presented method could be applied to any microorganism specific abstract collection as a starting point in checking out the most studied subjects referring to the selected dataset of microorganisms and chemical entities. Furthermore it can assist in the retrieval of co-occurrence associations between microorganisms and chemical entities and between chemical entities in the selected abstract collection.

The main contributions of this work can be summarized as follows:

- We present a detailed survey of state-of-the-art approaches in chemical NER, together with related benchmarks and data collections.
- We have designed and developed a chemical NER tool with performance close to state-of-the-art that can be easily used for identifying chemicals in scientific text.
- We have used the spaCy library, a user friendly open source software library that offers prebuilt statistical neural network models to create models for part-of-speech tagging, dependency parsing, text categorization and named entity recognition (NER).
- We have adopted association rule mining techniques to microbiology that can help

in retrieving co-occurrence associations between microorganisms and chemical entities and between chemical entities in a selection of organism specific abstracts.

- We further investigated the association rules that were generated from the frequent itemsets that were detected in the literature about the selected hyperthermophilic species. Interesting association rules are:
  - the association of carbohydrates to *C. bescii*
  - and the association of copper and *T. thermophilus* to heme.

  Manual curation of the literature showed that indeed Carbohydrate metabolism has been extensively studied in *C. bescii* for the production of ethanol from lignocellulosic biomass. Similarly, extensive studies of *T. thermophilus* report that the heme–copper oxygen reductases are able to catalyze the reduction of nitric oxide to nitrous oxide under reducing anaerobic conditions.

Outline. In the following Chapter, we present the requisite background and related work. In Chapter 3, we present in detail the available databases about chemichals and the available corpora with chemical entities, emphasising on the annotation principles of the CHEMDNER corpus. In Chapter 4, we present the development of the chemical NER using the spaCy package and its use on an abstract collection about nine selected species of hyperthermophilic microorganisms, in order to identify mentions of chemicals in the abstracts. In Chapter 5, we discuss the association rules that merged from the aforementioned abstract collection. In Chapter 6, we discuss our results.

2. BACKGROUND AND RELATED WORK

2.1 Text mining

Marti A. Hearst defined text mining as *"the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources"* [20]. He questioned the use of the term "mining" since it refers to "extracting precious nuggets of (or from) otherwise worthless rock". In practice, data mining applications can be used to derive new information or make novel hypotheses from the data and discover trends and patterns across very large data sets in an (semi)automated way, usually for the purposes of decision making. The general idea behind text mining applications is to use text for discovery in a more direct manner.

The possibility of finding useful links between information in related literature (literature-based knowledge discovery) was first demonstrated by Swanson [26]. Swanson introduced the ABC-principle, which proposes that notion A and notion C might be indirectly related (A leads to C) since they are related to notion B (A leads to B and B leads to C), even though they are never mentioned together in the same publication. Swanson applied the ABC-principle through the semi-automated two-node search tool: "Arrowsmith". "Arrowsmith" is a "closed" framework in which the user provides the hypothesis (notion A is related to notion C). Any articles where both A and C are mentioned together are removed so that the analysis would consider only indirect linkages between the two sets of articles. The hypothesis is then tested by a computational search for shared, related words (B) that could support the hypothesis [63]. Swanson inferred hypothetical relationships within the medical literature such as the helpful impact of fish-oil on patients suffering from Raynaud's disease [66] , the connection between magnesium deficiency and migraines [67] , the consumption of arginine affects the levels of somatomedin C in the blood [68] and the possible links between oestrogen and Alzheimer disease [62]. However it is important to experimentally verify these hypotheses. The first two hypotheses have been experimentally supported.

The first step in text-mining is the retrieval of textual resources relevant to a given scientific domain. This process is referred to as **information retrieval (IR)** [14]. IR techniques narrow down the search space from the entire document collection to the ones that belong to the area of interest [23]. PubMed is a popular biomedical IR system that is designed specifically to query databases of biomedical publications (MEDLINE and PubMed Central) and facilitates researchers to find publications relevant to their area of interest [26]. It uses:

- the Boolean model: that facilitates the retrieval of publications that contain a selected combination of terms by using logical operations
- and the vector model: that facilitates the query of terms since it represents each document by a term vector and each term has a frequency based value.

The document vectors can be compared to a query vector or compared to each other to calculate document similarity. Enhanced IR methods expand the queries by using synonyms and alternative terms based on a vocabulary [14].

Entity recognition (ER) is the identification of entities (for example biological terms such as species, genes, proteins, chemicals) in scientific literature. ER is a process that facilitates information extraction (IE). ER is divided into a first task which is the recognition

of words that refer to entities and a second task which is the unique identification of the entities (for example the accession number in a database). ER systems based on dictionaries use dictionary matching algorithms that allow variation in how the names are written. Controlled keyword vocabularies group together entities that belong to the same subject or category for keyword matching in documents. Ontologies are formally defined concepts that include relationships and rules that specify the dependencies between concepts. Ontologies are used to formally structure and categorize domain specific information. The BioPortal for Biomedical Ontologies contains a selection of dedicated ontologies. Machine Learning approaches in ER use algorithms such as: hidden Markov models (HMM), maximum entropy Markov models (MEMM) , conditional random fields (CRF), and support vector machines (SVM) that need to be trained on a carefully annotated training data sets [26, 14] (corpora) where entities of interest have been tagged [14]. Many ER systems combine dictionary matching with rule-based or statistical methods. The main challenge in ER is that biological entities have ambiguous names [26].

Information Extraction (IE) is the inference of new relations pulled out from scientific papers. The simplest approach followed in IE is the identification of entities that co-occur within abstracts, under the assumption that entities that often co-occur might be functionally related. The drawbacks of this approach are that entities might be mentioned together without being related and that directional relationships cannot be inferred. A more sophisticated approach is based on Natural Language Processing (NLP) methodologies that combine the analysis of syntax and semantics and can extract relationships based on the syntax tree and the semantic labels. The text is “tokenised”, part-of-speech tags are assigned to the tokens and a syntactic tree is obtained for each sentence [26].

Figure 2.1 presents the schematic presentation of a text mining workflow. The presented text mining workflow starts with information retrieval (IR) in order to get documents relevant to a subject of interest. Not all articles in PubMed Central (PMC) are available for text mining and other reuse. License terms vary. The license statement in each article states specific terms of use. The retrieved documents that will be used in the text mining workflow must be under license for text and data mining. Using named entity recognition (NER) these documents will be analyzed for the occurrence of specific keywords. Information extraction (IE) is about detecting links between the found keywords. During knowledge discovery (ND) keywords that could be used to infer new relations are linked together.

Text mining tools in biomedical literature can facilitate the work of researchers by providing a structured overview of their scientific area of interest and of recent developments made in scientific areas related to their work. The literature database PubMed gives access to more than 32 million scientific literature citations from MEDLINE, life science journals, and online books. The number of articles that are added to PubMed each year is growing fast. More specifically as shown in figure 2.2, in November 2017, over 100,000 new PubMed records were added to the database [55].

Text mining in biomedical literature is challenging for various reasons. The writing style is formal and complex, different types of documents (journal papers, patents or clinical reports) are written in different styles and also there is ambiguity in the terms that can be used, referring to genes, species, procedures, and techniques and also within each specific term, it is common to have multiple spellings, abbreviations and database identifiers [23].

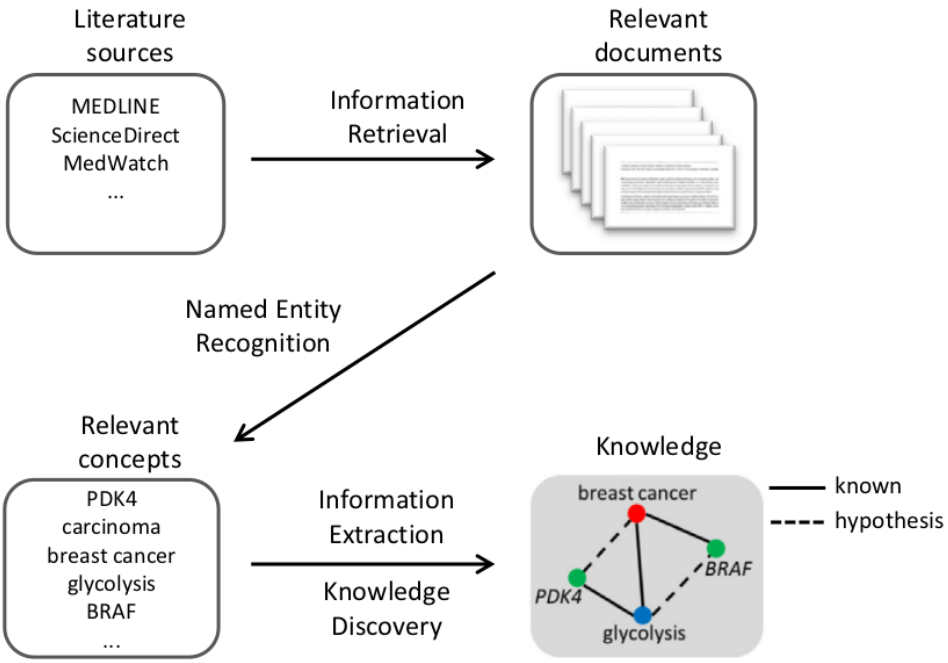

Figure 2.1: Schematic presentation of a Text Mining workflow [14].

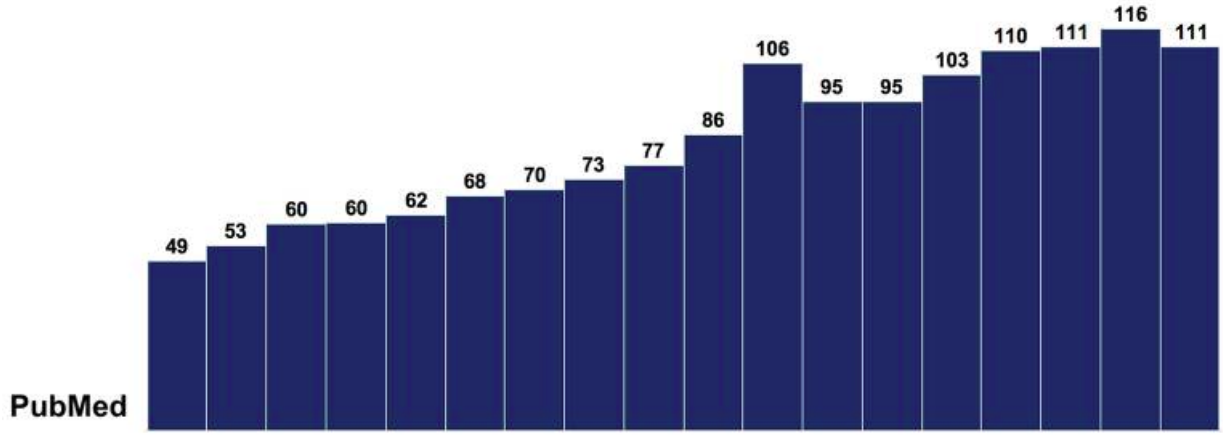

Figure 2.2: New PubMed records added by year (by 10,000s) between 2000 and 2017 [55].

2.2 Chemical Named Entity Recognition

2.2.1 Chemical Named Entity Recognition approaches

Chemical Named Entity Recognition (chemical NER) is the identification of chemical compounds and drugs from the rapidly growing scientific literature [13]. Chemical entities are important for chemistry, but also for other research areas such as life sciences, pharmacology, medicine, material sciences or physics. Natural language processing (NLP) and text mining technologies for the chemical domain (ChemNLP or chemical text mining) can be used to improve the integration of information from unstructured data such as patents or the scientific literature [35].

Chemical NER is a challenging task mainly due to the variability of the terms referring to chemicals that can be found in the literature [36]. Chemistry has various sub-disciplines

and is also studied in publications from other disciplines such as medicine, biology and pharmacology. Thus giving a strict definition to “what a chemical entity is?” across all the aforementioned disciplines is not possible. This can explain the variability of language expressions that refer to chemical molecules that is found in the literature. The International Union of Pure and Applied Chemistry (IUPAC) has defined a set of rules for the chemical nomenclature, but those naming standards are not strictly followed in the scientific literature. Furthermore chemical compounds and drugs often have many synonyms or aliases (e.g. systematic names, trivial names and abbreviations referring to the same entity). For example the anti-diabetic and anti-inflammatory drug ‘troglitazone’ can also be found in the literature with its brand name ‘Rezulin’ and its systematic (IUPAC) name is ‘(RS)−5−(4−[(6−hydroxy−2,5,7,8−tetramethylchroman−2−yl)methoxy]benzyl)thiazolidine −2,4−dione’. IUPAC naming (hyphens, brackets, spacing, etc.) complicates the identification of the entity boundaries (the tokenization component of an NLP pipeline). Also the use of acronyms, abbreviations, short chemical formulas and trivial names used in the literature complicates more the recognition of chemicals from NER systems and the likelihood of mapping all the alternative mentions of a chemical to its unique chemical structure. It should also be mentioned that new chemical compounds, with novel chemical names are discovered and described in new publications every day.

Over the past decades, many automatic chemical NER systems have been developed. These systems can be categorised into four groups according to the NLP approach that they follow. These systems might be dictionary-based, rule-based and machine learning-based, as well as hybrid chemical NER systems. A general overview of the steps followed in order to develop a chemical NER system [13] can be given by the following steps:

- Step 1: Preprocessing, the determination of the boundaries of the entities by splitting the text and tokenizing.
- Step 2: Feature processing, the linguistic analysis of the text, such as assigning parts-of-speech and features to words and phrases.
- Step 3: Name recognition, the recognition of the entity and its assignment to an entity type or class.
- Step 4: Normalization, the mapping of entities to their canonical names and their association with unique representations or identifiers in a database.

Dictionary-based NER systems [13] use lists of terms in dictionaries to identify whether a word or a phrase in the text matches any of the entries in the dictionary. These systems identify the chemical entity occurrences in text by implementing string-matching algorithms. Exact matching approaches make an exact string-match search against the text to a given list of terms. Flexible or approximate matching approaches perform “fuzzy” matching and are more popular in chemical NER than exact matching approaches. An example of dictionary-based systems that are used to extract drug and small molecules names and molecules via string matching methods is the method developed based on the dictionary built by Hettne et al.,[22] that combines information from the following databases: UMLS, MeSH, ChEBI, DrugBank, KEGG, HMDB and ChemIDplus.

Rule-based NER systems [13] use a set of hand-made rules to identify the names of entities in text. Usually the models consist of grammatical and syntactic rules that are sometimes combined with dictionaries as well. The two types of rules that are used in the

rule-based systems are the Pattern-based rules that depend on the orthographic or morphological patterns of the words and the Context-based rules that depend on the context of the words in the text.

Machine learning (ML)-based NER systems [13] use statistical models for recognising entities. The ML algorithms used in NER systems are divided into two categories: supervised learning algorithms and unsupervised learning algorithms. Supervised learning algorithms learn a classification system on a labeled training corpus. Examples of supervised learning models are CRFs, Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MEMMs), Naïve Bayes and Support Vector Machines (SVMs). Whereas the goal of unsupervised learning algorithms is to build representations from data whose labels are not known during training. Semi-supervised algorithms use both labelled and unlabelled data.

2.2.2 Tools for chemical Named Entity Recognition

A number of chemical NER tools have been developed. These tools follow a wide spectrum of approaches. In recently developed ones, it is observed authors build and deploy neural network architectures. A brief description is following about the most recent and popular chemical NER tools.

- Multi-Task Learning for Chemical Named Entity Recognition with Chemical Compound Paraphrasing tool [73]: This method uses multi-task learning by jointly training a chemical NER model and a chemical compound paraphrase model. The Long short-term memory (LSTM) network of the NER model captures chemical compound paraphrases by sharing the parameters of the LSTM and character embeddings between the two models.
- ChemListem [11]: This approach deploys two NER systems. The first system, similarly to traditional CRF-based systems, assigns tags to a sequence of tokens, each token bearing features from a rich feature set. It uses a bidirectional LSTM network. It does not include information about neighbouring tokens in the feature set, instead it relies on the neural network structure to carry the information from neighbouring tokens to the right place. The second system labels a sequence of characters, rather than words (i.e. it does not use a tokeniser), and does not use a rich feature set. It instead uses character embeddings and multiple LSTM layers in order to induce the equivalent of a feature set internally. The two systems can be used independently or as an ensemble (`https://bitbucket.org/rscapplications/chemlistem/src/master/`).
- deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules [34]: A combination of convolutional and stateful recurrent neural networks with attention-like loops and hybrid word-and-character-level embeddings.
- Attention-based BiLSTM-CRF tool [47]: This approach uses an attention-based bidirectional Long Short-Term Memory Neural Network with a conditional random field layer (Att-BiLSTM-CRF) (`https://github.com/lingluodlut/Att-ChemdNER`).
- LeadMine [46]: It a hybrid system that combines dictionary and rule-based lookup. In order to increase the chance of recognising the trivial names slightly outside the

coverage of the dictionary it uses the following approaches: spelling correction, merging of adjacent entities and entity extension. It is a commercially licensed tool (https://www.nextmovesoftware.com/leadmine.html).

- OSCAR [27]: OSCAR4 employs a naïve Bayesian model to identify "chemical" tokens in text and offers a choice of two methods for the identification of multi-token named entities:
  - The PatternRecogniser uses predetermined regular-expression style heuristics.
  - The MEMMRecogniser employs machine learning in the form of a Maximum Entropy Markov Model (MEMM).

  OSCAR4 uses these methods to identify four classes of named entities (Chemical, Reaction, Chemical Adjective and Enzyme) as well as dictionary lookup to identify a predetermined set of ontology terms. (https://sourceforge.net/projects/oscar3-chem/, https://github.com/BlueObelisk/oscar4)
- CheNER [71, 70]: A hybrid system that applies a CRFs tagger and a Regular Expression tagger (which include dictionary and regular expression approaches) to identify formulae and identifier name types.
- ChER [4, 5]: A CRF-based method whose performance was optimised by:
  - (a) the selection of best-suited pre-processing components,
  - (b) the incorporation of CRF features capturing chemistry-specific information, and
  - (c) the application of post-processing heuristics.

  It is available as a workflow in the Argo text mining workbench Users may select one of chemical, drug or metabolite, as the model that will be used for the recognition.
- ChemSpot [61] A chemical NER tool for identifying mentions of chemical entities (trivial names, drugs, abbreviations, molecular formulas and IUPAC) in text. It implements a hybrid approach that combines a CRF model for identifying IUPAC entities with a dictionary built from ChemIDplus for extracting drugs, abbreviations, molecular formulas and trivial names (https://github.com/rockt/ChemSpot).
- Yeast MetaboliNER [54] Tool based on a CRF model that utilised the Chemical Entities of Biological Interest (ChEBI) and Human Metabolome (HMDB) databases as dictionaries (http://nactem7.mib.man.ac.uk/metaboliner/).
- ChemicalTagger [19]: ChemicalTagger uses a combination of OSCAR, domain-specific regex and English taggers to identify parts-of-speech. The ANTLR grammar is used to structure this into tree-based phrases.

2.3 spaCy

spaCy [50] (https://spacy.io/) is a free, open-source Python library designed to be used in building a wide variety of different applications for advanced Natural Language

Processing (NLP). According to the designers and developers of spaCy, it is an opinionated and easy to deploy system that is designed to offer to the user a limited amount of the state-of-the-art algorithms which have equivalent functionality in order to deliver good performance, while being as user friendly as possible. spaCy is designed to offer a good developer experience through its detailed documentation, consistent naming and good error handling.

2.3.1 Computational linguistic features and machine learning.

When analyzing text, the syntactic structure of the sentences is important. For example there is a difference whether a noun is the subject or the object of a sentence. spaCy provides a variety of linguistic annotations like “word types” (distinct words), “parts of speech”, and how the words are syntactically related to each other, in order to give insights into a text’s grammatical structure. spaCy’s text processing capabilities in attributing linguistic features to the words of a text combined with its machine learning functionalities can be used in various applications of NLP. spaCy is designed to return a Doc object with a variety of annotations having as input raw text. In this section a brief description of spaCy’s features and capabilities is given. Some of them refer to linguistic concepts, while others are related to more general machine learning functionality.

- Tokenization: Segmenting text into words, punctuation marks etc. This is done by applying rules specific to each language.
- Part-of-speech (POS) Tagging: Assigning word types to tokens, like “verb” or “noun”.
- Dependency Parsing: Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
- Lemmatization: Assigning the base forms of words. For example, the lemma of “was” is “be”, and the lemma of “phenols” is “phenol”.
- Sentence Boundary Detection (SBD): Finding and segmenting individual sentences.
- Named Entity Recognition (NER): Labelling named “real-world” objects, like persons, companies or locations.
- Entity Linking (EL): Disambiguating textual entities to unique identifiers in a knowledge base.
- Similarity: Comparing words, text spans and documents and how similar they are to each other.
- Text Classification: Assigning categories or labels to a whole document, or parts of a document.
- Rule-based Matching: Finding sequences of tokens based on their texts and linguistic annotations, similar to regular expressions.
- Training: Updating and improving a statistical model’s predictions.
- Serialization: Saving objects to files or byte strings.

spaCy offers the nlp function that first tokenizes the input raw text and returns a Doc object. The Doc object is then processed through a processing pipeline whose steps can include: Part-of-speech (POS) Tagging, Lemmatization, Dependency Parsing and Named Entity Recognition (NER). Each pipeline component returns a processed Doc object, which is then passed on to the next component. The final Doc object includes a variety of linguistic annotations. The capabilities of a processing pipeline depend on the components, their models and how they were trained. For example, a pipeline for named entity recognition needs to include a trained named entity recognizer component with a statistical model and weights that enable it to make predictions of entity labels. Figure 2.3 presents an example of a processing pipeline that includes a tagger, a lemmatizer, a parser and an entity recognizer. The nlp function tokenizes the text to produce a Doc object. The Doc is then processed in several different steps. Each pipeline component returns the processed Doc, which is then passed on to the next component.

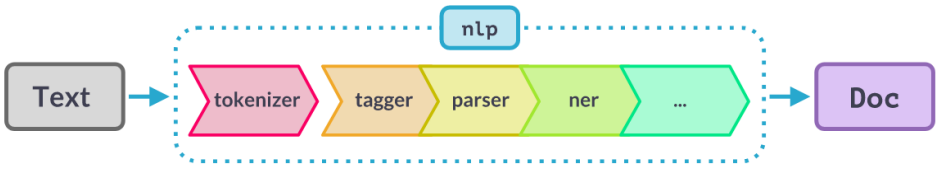

Figure 2.3: spaCY processing pipeline that includes a tagger, a lemmatizer, a parser and an entity recognizer.

spaCy offers already trained pipelines for various languages, which can be installed as individual Python modules. A trained pipeline consists of multiple components that use a statistical model trained on labeled data. spaCy's pipeline packages typically include binary weights for the part-of-speech tagger, the dependency parser and the named entity recognizer, lexical entries in the vocabulary (words and their context-independent attributes like the shape or spelling), data files with lemmatization rules and lookup tables, word vectors (multi-dimensional meaning representations of words) and configuration options, like the language and processing pipeline settings and model implementations to use.

2.3.1.1 Tokenizer

During pipeline processing, the first step is the tokenization of the raw text. The text is segmented into words and punctuation marks. This is done by applying rules specific to each language. For example, punctuation at the end of a sentence should be split off whereas in the example of the word "U.K." should remain one token. At the end of the tokenization component of the pipeline the Doc object consists of individual tokens that can be iterated over.

2.3.1.2 Part-of-speech (POS) Tagger and Dependency Parser

Part-of-Speech(POS) Tagging is the process of assigning different labels known as POS tags to the words in a sentence that tell us about the part-of-speech of the word. Dependency parsing is the process of analyzing the grammatical structure of a sentence based on the dependencies between the words in a sentence. To use this functionality spaCy needs a trained pipeline that supports the pipeline component: parser. After tokenization,

spaCy can parse and tag a given Doc object. spaCy's trained pipeline component: parser and its statistical models enable spaCy to predict which tag and label most likely applies to each token in the context of each sentence.

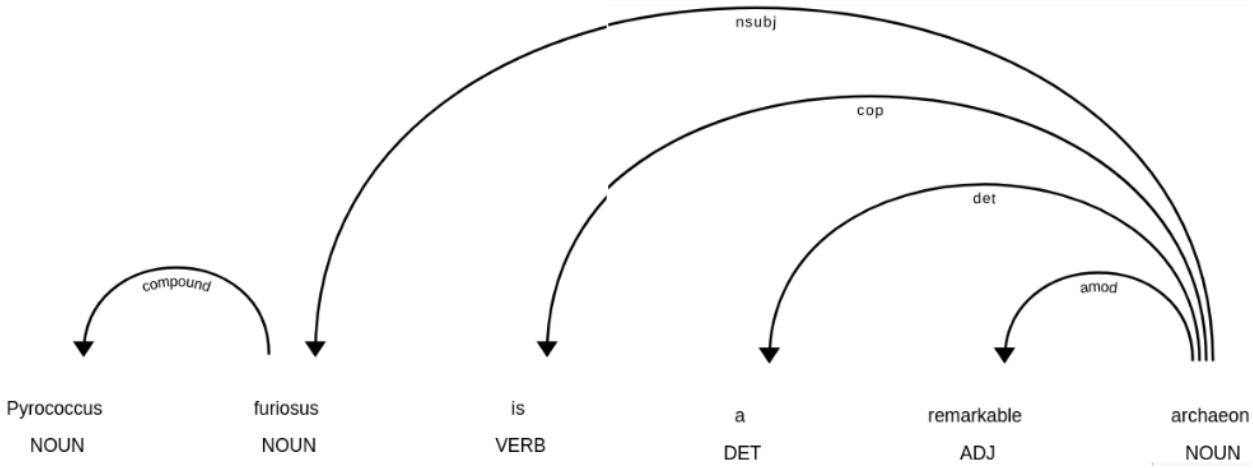

Figure 2.4: Part of an example sentence: " Pyrococcus furiosus is a remarkable archaeon able to grow at temperatures around 100 °C." and its dependencies visualised using spaCy's built-in displaCy visualizer.

In figure 2.4, the syntactic dependencies between the words of part of the sentence: " Pyrococcus furiosus is a remarkable archaeon able to grow at temperatures around 100 °C." are visualised sing spaCy's built-in displaCy visualizer. The arrows represent the dependency between two words in which the word at the arrowhead is the child, and the word at the end of the arrow is head. The root word can act as the head of multiple words in a sentence but is not a child of any other word. The root word of a sentence has multiple outgoing arrows but none incoming.

2.3.1.3 Named Entity Recognizer (NER)

To use this functionality spaCy needs a trained pipeline that supports the pipeline component: "ner". "*A Named Entity (NE) is a proper noun, serving as a name for something or someone*" [42]. NEs are generally divided into two categories: generic NEs (e.g., a person, a location, a product or a book title) and domain-specific NEs (e.g. in the biomedical domain: proteins, small chemical molecules, and genes). The ner component of a pretrained model can predict various types of NEs in a Doc object according to the annotated dataset it was trained on. Pre-Trained statistical models strongly rely on the examples that they were trained on so the ner component of a pipeline usually needs fine tuning, depending on the application.

An example of the function of scispaCy's pretrained models for processing biomedical text is given in figure 2.5. In figure 2.5 are visualised the Named Entities identified by the ner component of the pipeline of the en_ner_bc5cdr_md model from the scispacy package. The ner component of the aforementioned pipeline is trained on the BC5CDR corpus that has annotated chemicals, diseases and chemical-diseases interactions.

2.3.1.4 Word embeddings and semantic similarity

To use this functionality spaCy needs a trained pipeline that supports the pipeline component: "tok2vec". spaCy provides pre-trained word vectors for various languages. Similarity

Pyrococcus furiosus is a remarkable archaeon able to grow at temperatures around 100 °C. At the metabolite level, 37 compounds were quantified. The level of di-myo-inositol phosphate CHEMICAL , a canonical heat stress solute among marine hyperthermophiles, increased considerably (5.4-fold) at elevated temperature. Also, the levels of mannosylglycerate CHEMICAL , UDP-N-acetylglucosamine CHEMICAL (UDPGlcNac) and UDP-N-acetylgalactosamine CHEMICAL were enhanced. The increase in the pool of UDPGlcNac was concurrent with an increase in the transcript levels of the respective biosynthetic genes.

Figure 2.5: The Named Entities identified by the ner component of the pipeline of the en_ner_bc5cdr_md model from the scispacy package.

is determined by comparing word embeddings or word vectors (multi-dimensional meaning representations of a word). Word vectors can be generated using an algorithm like Word2vec or GloVe. In spaCy's pipeline packages (packages whose names end in "_md" and "_lg") individual tokens have vectors. Pipeline packages whose names end in "_lg" include a larger number of unique vectors than whose names end in "_md" and should be preferred if an application would benefit more from a larger vocabulary with more vectors.

2.3.1.5 Training

spaCy's "tagger", "parser", "text categorizer" and many other components are powered by statistical models. Every "decision" these components make – for example, which part-of-speech tag to assign, or whether a word is a named entity – is a prediction based on the model's current weight values. The weight values are estimated based on examples the model has seen during training. To train a model, the user first needs training data – examples of text, and the labels that the model will "learn" to predict. This could be a part-of-speech tag, a named entity or any other information.

Training is an iterative process in which the model's predictions are compared against the reference annotations in order to estimate the gradient of the loss. The gradient of the loss is then used to calculate the gradient of the weights through back-propagation. The gradients indicate how the weight values should be changed so that the model's predictions become more similar to the reference labels over time.

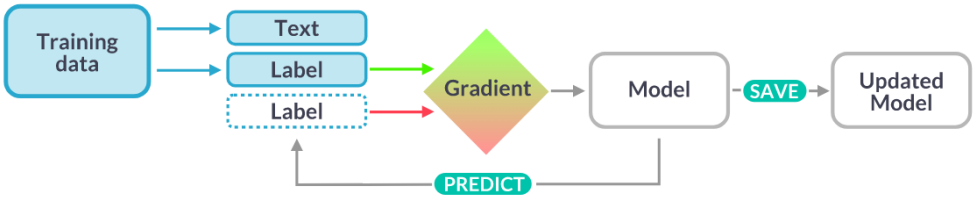

Figure 2.6: Visual representation of the training process of a spaCy model.

When training a model, it is important that the model generalizes well across unseen data. For example when training a model to identify company names in texts, we don't just want the model to learn that this one instance of "Amazon" right here is a company – we want it to learn that "Amazon", in contexts like this, is most likely a company. That is why the training data should always be representative of the data we want to process. A model trained on Wikipedia, where sentences in the first person are extremely rare, will likely perform badly on Twitter. Similarly, a model trained on novels will likely perform badly on

scientific text. In order to test the performance of the model and how well it generalizes an extra set of unseen examples is needed the evaluation dataset.

The spacy v3 release introduced the “spacy train” command that is used with the Command Line Interface (CLI). In Spacy v3, there has been a shift towards training model pipelines using the “spacy train” command on the CLI instead of writing a training loop in Python. This is recommended by the authors of the package because it is faster in training a model and helps with the model validation process (evaluating a trained model on a test dataset) while training. During training the user must provide the training dataset and a testing dataset (optional) that can be used for validation (in spaCy’s binary format). The trained model will be evaluated periodically during training and the scores (e.g precision, recall, f1-score while training the “ner” component of the pipeline) will be printed. The model is trained on the training data and the evaluation data is not used for training, it is used for scoring during the training process.

The number of times that the learning algorithm will work through the entire training dataset is the hyperparameter: number of epochs. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch consists of one or more batches. Furthermore, the “spacy train” command comes with early stopping logic built in. The training will stop once the model performance stops improving on a hold out validation dataset.

Training config files include all settings and hyperparameters for training the pipeline. Under the hood, the training config uses the configuration system provided by the machine learning library Thinc (`https://thinc.ai/`). Training config files include the definition of the nlp object, its tokenizer and processing pipeline component names, the pipeline components models, settings and controls for the training and evaluation process.

2.4 scispaCy

scispaCy [52] is a specialized NLP Python library for processing biomedical text, using as its basis the spaCy library. scispaCy contains pretrained models for processing scientific text: biomedical or clinical text. The Allenai Institute of Artificial Intelligence developed several model pipelines for natural language processing tasks focused on biomedical text in order to be used in new applications in biomedical information extraction.

The POS tagger and dependency parser of the models distributed by the scispaCy project are a joint component in the models’ pipeline that have been developed based on the arc-eager transition-based parser described by Goldberg and Nivre (2012) [16], trained with a dynamic oracle. In transition-based dependency parsing, a parser processes an input sentence and predicts a sequence of parsing actions in a left-to-right manner. During the training of the transition-based dependency parsers, a static oracle, which predicts an optimal transition sequence for a sentence and its gold parse tree, is used. Goldberg and Nivre (2012) developed an improved dynamic oracle for the arc-eager transition system that gives a set of optimal transitions for every valid parser configuration, including configurations from which the gold tree is not reachable. In such cases, the oracle provides transitions that will lead to the best reachable tree from the given configuration. The dynamic oracle was used to train a deterministic left-to-right dependency parser. The advantages of using a dynamic oracle in training a parser are that it is not restricted to a particular canonical order of transitions and that it is less sensitive to the effect of error propagation because it can handle configurations that are not part of any gold sequence.

The architecture of the feature function for the underlying statistical model in parsing design is an important challenge. Typically state-of-the-art parsers rely on linear models over hand-crafted feature functions. The feature functions look at core elements of the transition-based dependency parsing framework (for example “word on top of stack”, “leftmost child of the second-to-top word on the stack”, “distance between the head and the modifier words”) and consist of several templates, where each template represents a binary indicator function over a conjunction of core elements (resulting in features of the form “word on top of stack is X and leftmost child is Y and . . . ”). Kiperwasser and Goldberg (2016) proposed to replace the hand-crafted feature functions with simpler feature functions which make use of automatically learned Bidirectional LSTM representations.

Kiperwasser and Goldberg (2016) [32] presented a scheme for dependency parsing using techniques from the neural-networks literature. Their scheme is based on bidirectional-LSTMs (BiLSTMs). They demonstrated the effectiveness of their approach by using the BiLSTM feature extractor in a transition-based parsing architecture, as well as in a graph-based. They used the greedy transition-based parser according to the standard techniques described in the literature: margin-based objective, dynamic oracle training, error exploration, MLP-based non-linear scoring function, but also introduced the idea of representing a few important items on the stack and the buffer using BiLSTMs and training the BiLSTM encoder together with the rest of the network. BiLSTMs (an extension of Recurrent neural networks (RNNs), composed of two RNNs one reading the sequence in its regular order: RNN F and the other in reverse: RNN R) are very good at displaying elements (for example words) in a sequence together with their contexts, taking into account the elements in its surrounding contexts. In Kiperwasser and Goldberg’s approach each word is represented by its BiLSTM encoding. A small set of these BiLSTM encodings is used as the feature function which is then scored using a non-linear scoring function (multi-layer perceptron - MLP) with one hidden layer. The scoring function has access to the words and POS-tags of the BiLSTM vectors and to the words and POS-tags of the words in an infinite window surrounding them, thus it is plausible that the scoring function can be sensitive also to the distance between the BiLSTM vectors. In the proposed architecture the BiLSTM is trained with the rest of the parser in order to learn a good feature representation for the parsing problem.

In the architecture of the arc-eager transition parsing system proposed by Kiperwasser and Goldberg (2016), at each stage in the training process, the parser assigns scores with an MLP to all the possible transitions, selects a transition, applies it, and moves to the next step. The highest scoring transition is followed and error-exploration training is performed using the dynamic-oracle [16].

Figure presents an Illustration of the neural model scheme of the transition-based parser when calculating the scores of the possible transitions in a given configuration. Each transition is scored using an MLP that is fed the BiLSTM encodings of the first word in the buffer and the three words at the top of the stack (the colors of the words correspond to colors of the MLP inputs above), and a transition is picked greedily. Each xi is a concatenation of a word and a POS vector, and possibly an additional external embedding vector for the word. The figure depicts only one single-layer BiLSTM. When parsing a sentence, the scores for all possible transitions are computed iteratively and then the best scoring action is applied until the final configuration is reached [32].

scispaCy models’ components of the pipelines are trained on data from a variety of sources. The dependency parser and part of speech tagger are jointly trained on the GENIA 1.0 Treebank of McClosky and Charniak [48], which was created by self-training (a method of

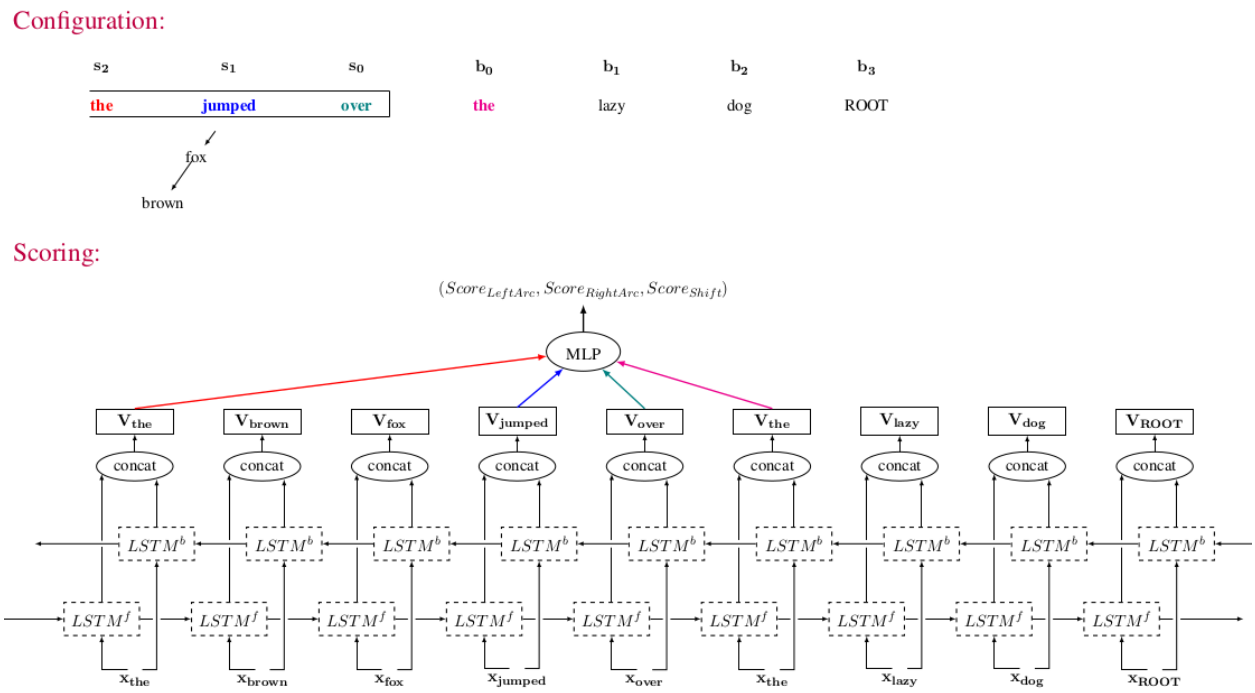

Figure 2.7: Illustration of the neural model scheme of the transition-based parser when calculating the scores of the possible transitions in a given configuration. The configuration (stack and buffer) is depicted on the top[32].

using an existing parser for parsing extra data and then creating a second parser by treating the extra data as further training data) the standard Charniak/Johnson Penn-Treebank parser using biomedical abstracts from the Genia 1.0 corpus [29]. The treebank was converted to basic Universal Dependencies using the Stanford Dependency Converter35. In order to further increase the robustness of the dependency parser and part of speech tagger to generic text, during their training the POS and dependency parsing annotations of the OntoNotes 5.0 corpus were additionally used30. The OntoNotes [74] corpus derived from a project where various types of text: news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, talk shows in three languages: English, Chinese, and Arabic were annotated with structural information (syntax and predicate argument structure) and shallow semantics (word sense linked to an ontology and coreference).

scispaCy contains three core released packages: en_core_sci_sm, en_core_sci_md and en_core_sci_lg. They are full pipelines for the processing of biomedical data. The pipelines in the en_core_sci_md and en_core_sci_lg packages have a larger vocabularies and include word vectors (50k and 600k word vectors respectively), while those in the en_core_sci _sm package have a smaller vocabulary and do not include word vectors.

The Named Entity Recogniser (NER) content in spaCy pipeline is a transition-based system based on the chunking model introduced by Lample et al. (2016) [39]. Lample et al., explored a new architecture that chunks and labels a sequence of inputs using an algorithm similar to transition-based dependency parsing. This model is referred to as the Stack-LSTM model. Long Short-term Memory Networks (LSTMs) are artificial Recurrent neural networks (RNNs) designed to learn long-range dependencies from an input sequence of vectors. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. Several gates control the proportion of the input to give to the memory cell, and the proportion from the previous state to forget, in order not to be biased towards the most recent inputs and to capture long-range dependencies. While sequen-

tial LSTMs model sequences from left to right, LSTMs augmented with a “stack pointer (Stack-LSTMs) permit embedding of a stack of objects that are both added to (using a push operation) and removed from (using a pop operation). This allows the Stack-LSTM to work like a stack that maintains a “summary embedding” of its contents. This model architecture can directly construct representations of multi-token names; for example, a people’s names and last names are directly composed into a single representation. Tokens are represented as hashed, embedded representations of the prefix, suffix, shape and lemmatized features of individual words.

The main NER model in the released packages in scispaCy is trained on the MedMentions Entity Linking dataset [51], so it recognises a wide variety of entity types but does not predict the entity type.

Four additional packages whose ner component of the pipeline was trained on the entities of available corpuses were released: en_core_craft|jnlpba|bc5cdr|bionlp13cg_md. The models’ ner component of the pipeline were trained on the CRAFT corpus [3]: 67 full-text biomedical journal articles from PubMed with approximately 100,000 concept annotations to 7 different biomedical ontologies/terminologies (Chemical Entities of Biological Interest, Cell Ontology, Entrez Gene, Gene Ontology: biological process, cellular component, and molecular function, NCBI Taxonomy, Protein Ontology, Sequence Ontology), the JNLPBA corpus [24, 30]: it contains entity types including protein, DNA, RNA, cell line and cell type, 85000 entity mentions, 25000 entity mentions with database identifiers and 5000 attribute tags, the BC5CDR corpus [41]: 1500 PubMed articles with 4409 annotated chemicals, 5818 diseases and 3116 chemical-disease interactions, the BioNLP13CG corpus [59]: annotates 16 entity types in 600 PubMed abstracts relevant to the field of cancer genetics, 250 of which are part of the multi-level event extraction (MLEE) corpus and the other 350 abstracts were selected by querying PubMed for MeSH terms that relate to hallmarks of cancer, such as apoptosis and metastasis.

2.5 Association Rules

2.5.1 Association Rule mining basic concepts

Association rule learning is a rule-based machine learning method used in discovering relations between variables in large databases. The method’s purpose is to identify strong rules discovered in databases using some predefined measures (thresholds). In this section the basic concepts of association rule mining are described based on the book of J. Han and M. Kamber, Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)[18].

Frequent patterns are itemsets and substructures (such as subgraphs, subtrees) that appear frequently in a data set. Identifying frequent patterns (frequent pattern mining) is an important task in mining relationships among data (such as associations, correlations) and it helps in data mining tasks (such as in data classification and clustering). In general, association rule mining can be viewed as a two-step process:

1. Detecting all the frequent itemsets: The frequent itemsets will occur at least as frequently as a predetermined minimum support count. This is a challenging step since a huge number of itemsets might be generated, when setting a low minimum support threshold.

2. Generating strong association rules from the frequent itemsets: The association rules must satisfy a predetermined minimum support and minimum confidence. Association rules can help in showing the probability of relationships between data items.

In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best-known constraints are minimum thresholds on *support* and *confidence*. In simple words support is an indication of how frequently the itemset appears in the dataset, confidence is an indication of how often the rule has been found to be true and lift considers both the support of the rule and the overall data set.

D is the task-relevant data. $I = \{I1, I2, ..., Im\}$ is an itemset. T is a nonempty itemset subset of T, $T \subseteq I$. A is a set of items subset of T, $A \subseteq T$. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, $A \neq \varnothing$, $B \neq \varnothing$, and $A \cap B = \varnothing$. The rule $A \Rightarrow B$ holds in the set D with support s, where s is the percentage of itemsets in D that contain $A \cup B$ (the union of sets A and B). This is the probability, $P(A \cup B)$. The rule $A \Rightarrow B$ has confidence c in the set D, where c is the percentage of itemsets in D containing A that also contain B. This is the conditional probability, $P(B|A)$.

$$support\,(A \Rightarrow B) = P(A \cup B)$$

$$confidence\,(A \Rightarrow B) = P(B|A)$$

Rules that satisfy both a minimum support threshold and a minimum confidence threshold are called strong.

A set of items is an itemset for example an itemset that contains k items is a K-itemset. The occurrence frequency (or or count of the itemset) of an itemset is the number of sets that contain the itemset. If the value of the relative support of an itemset I is higher than a prespecified minimum support threshold then I is a frequent itemset.

$$confidence\,(A \Rightarrow B) = P(B|A) = \frac{supportcount(A \cup B)}{supportcount(A)}$$

2.5.2 Frequent Itemset Mining Methods

A milestone in frequent itemset discovery is the development of an Apriori-based, levelwise mining method for associations, which has encouraged the development of various kinds of association mining algorithms and frequent itemset mining techniques. The Apriori algorithm is the basic algorithm for finding frequent itemsets. It is based on the observation that all nonempty subsets of a frequent itemset must also be frequent.The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. First, the dataset is scanned to collect the count for each item and to gather those items that satisfy minimum support (the set of frequent 1-itemsets). This set is used to find the set of frequent 2-itemsets, which is used to find the set of frequent 3-itemsets etc, until no more frequent k-itemsets can be found. The finding of each frequent k-itemsets requires one full scan of the database. Once the frequent itemsets have been found, strong association rules are generated by satisfying both a predefined minimum support and minimum confidence threshold. Many variations of the Apriori algorithm

have been proposed that focus on improving the efficiency and scalability of the original algorithm (for example by using hashed based techniques, reducing the number of itemsets scanned, partitioning the data, mining subsets of the given data).

Frequent pattern-growth methods for mining frequent itemsets follow a divide-and-conquer strategy that decreases the search space to only the data sets containing the current frequent itemsets. The algorithm first compresses the database representing frequent items into a frequent pattern tree, (FP-tree), which retains the itemset association information. It then divides the compressed database into a set of conditional databases (a special kind of projected database), each associated with one frequent item or "pattern fragment," and mines each database separately. For each "pattern fragment," only its associated data sets need to be examined. Therefore, this approach may substantially reduce the size of the data sets to be searched, along with the "growth" of patterns being examined.

Traditional frequent pattern analysis focuses on binary transaction data, such as the data that accumulate when customers purchase items, for example in a supermarket. These market basket data can be represented as a collection of transactions, where each transaction corresponds to the items purchased by a specific customer. In terms of a frequent pattern analysis and association rule mining this dataset is represented as a binary matrix, where there is one row for each transaction and one column for each item. In this matrix if a transaction contains an item (the customer has purchased the item) the entry in the cell of the matrix is 1 and if a transaction does not contain an item (the customer has not purchased the item) the entry in the cell of the matrix is 0, indicating whether or not an item was purchased by a particular customer. Given such a binary matrix representation, a key task in association analysis is to find frequent itemsets in this matrix, which are sets of items that frequently occur together in a transaction. The strength of an itemset is measured by its support, which is the number of transactions in the data set in which all items of the itemset appear together. The interesting patterns in these data sets are the frequent itemsets (sets of items that are frequently purchased together) and the association rules (rules that capture the fact that the purchase of one item or itemset possibly implies the purchase of a second set of items or itemsets). This example shows the potential economic benefits of pattern discovery with association analysis. Several efficient algorithms, such as *Apriori* and *FPGrowth* have been designed for discovering frequent itemsets in a given binary data matrix.

An example of an association rule is given in the figure 2.8. An association rule consists of an antecedent and a consequent, both of which are an item or an itemset (list of items). The inferred relation here is co-occurrence. Here is an association rule example:

$$(KidneyBeans, Onion) \Rightarrow (Eggs)$$

This association rule is a representation of finding eggs on the basket which has kidney beans and onion in it. It implies that the purchase of kidney beans and onions possibly implies the purchase of eggs. The directed graph in figure 2.8 is built for this rule (co-occurrence relationship). Arrows are drawn in blue. The node labeled R0 refers to this rule, and it has incoming and out coming edges. Incoming edge(s) represent antecedents and outgoing edge(s) represent consequentes.

2.5.3 Pattern Evaluation Methods

A major bottleneck for successful applications of association rule mining is evaluating the importance of the generated association rules and collecting the most "interesting" as-

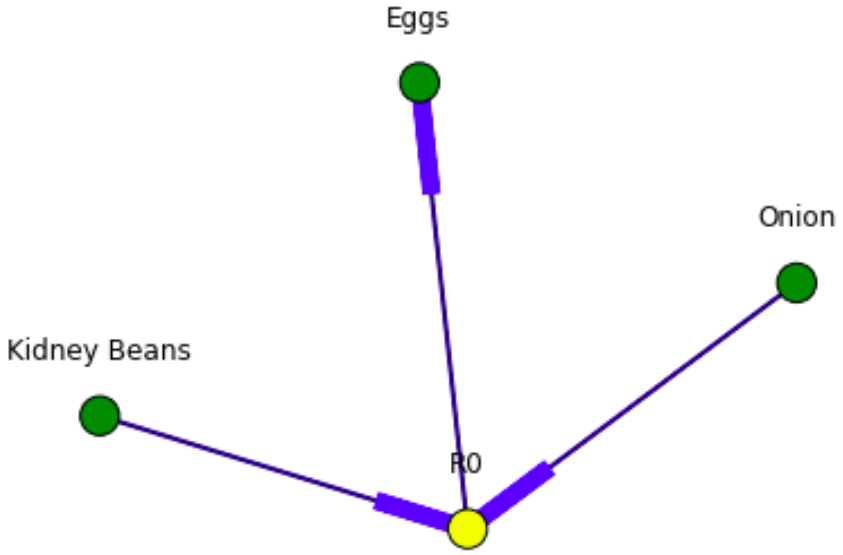

Figure 2.8: Visualization of the association rule: (Kidney Beans, Onion) ==> (Eggs)

sociation rules. Most applications employ a minimum support threshold and a minimum confidence threshold in order to exclude a good number of “uninteresting” rules and still many of the rules generated are still not “interesting” to the users. Of course labeling an association rule as “interesting” is a subjective accession and often even strong association rules can be “uninteresting”. Whether or not a rule is interesting can be judged only by the user according to the used dataset and the addressed questions.

In this section additional pattern evaluation measures for the discovery of only interesting rules, are going to be presented (except employing a minimum support threshold and a minimum confidence threshold, since they are insufficient at filtering out uninteresting association rules). A correlation rule does not give significance only to the support and confidence but also to the correlation between itemsets A and B. A few correlation measures for mining large data sets are presented:

- Lift: The lift between the occurrence of A and B assesses the degree to which the occurrence of one “lifts” the occurrence of the other. The occurrence of itemset A is independent of the occurrence of itemset B if $P(A \cup B) = P(A)P(B)$; otherwise, itemsets A and B are dependent and correlated as events.

  $$lift(A, B) = P(A \cup B)/P(A)P(B)$$

  If the resulting value of lift is less than 1, then the occurrence of A is negatively correlated with the occurrence of B, meaning that the occurrence of one likely leads to the absence of the other one. If the resulting value is greater than 1, then A and B are positively correlated, meaning that the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, then A and B are independent and there is no correlation between them.

- x^2 measure: The x^2 value is the squared difference between the observed and expected value for a pair A and B (slot in the contingency table), divided by the expected value. This amount is summed for all slots of the contingency table.

So instead of just using the simple support–confidence framework to evaluate frequent patterns, other measures, such as lift and x^2 should also be taken into consideration since

they might reveal more fundamental pattern relationships. Researchers have studied more alternative pattern evaluation measures: all_confidence, max_confidence, Kulczynski, cosine (which can be viewed as a harmonized lift measure) and other measures that have been studied in the literature.

Overall, the use of only support and confidence measures to mine associations may generate a large number of rules, many of which can be uninteresting to users. It is recommended that the support–confidence framework should be followed by a pattern interestingness measure (lift, x^2, all_confidence, max_confidence, Kulczynski, cosine, etc) that will contribute in focusing the mining toward rules with strong pattern relationships. The added interestingness measure usually considerably reduces the number of rules generated and leads to the discovery of more meaningful rules.

2.5.4 Association Rule Mining Techniques in Bioinformatics

As it has been previously described, the area of data mining known as association analysis seeks to find relationships among a set of objects. The most studied example of data sets analyzed by the group of techniques of association analysis is the market basket data. The group of techniques of association rule mining is not widely used in the domain of bioinformatics and computational biology [2].

The majority of studies that implement association rule mining methodologies in the field of biomedicine are applied to Healthcare. The increasing amount of medical and research clinical data accumulated in medical databases offers a wide reservoir of data where association rule mining is essential for identifying new, unexpected and interesting patterns in medical databases. In Healthcare, association rules mined from medical databases, are considered to be useful as they can play an important role in the possibility to conduct intelligent diagnosis and extract valuable information and build important knowledge bases in an automated way.

Jung et al. (2013), searched for risk factors that are associated with complications of cerebral infarction in patients with atrial fibrillation (AF) and for association rules among these factors. They identified four independent risk factors (age, hypertension, initial electrocardiographic rhythm, and initial echocardiographic left atrial dimension) as strong predictors for complications of cerebral infarction in patients with AF and four rules to identify complications of cerebral infarction, based on medical record data [28]. The authors point out that for the assession of the reliability of these risk factors and association rules further clinical research is required. Cheng et al. (2013), investigated the associations between developing cognitive impairment and emotional abnormality and neurobehavioral and motor disorders from a clinical database of pediatric subjects diagnosed with CP [8]. Li et al. (2017), “mined” for meaningful association rules between the high risk factors of stroke, based on the apriori algorithm [43]. For the association analysis they used a dataset of information (inducing factors: hypertension, atrial fibrillation, dyslipidemia, diabetes mellitus, smoking, exercise, overweight and family history of stroke) about 985,325 Chinese adults aged 40 years and over, 1.65% of whom were stroke patients and 98.35% without stroke. Based on the threshold they set for the Apriori algorithm they found eight meaningful association rules between stroke and its high risk factors and between high risk factors they found 25 meaningful association rules. Their results were supported by a large number of medical studies. Parente et al., (2018) proposed a method of association rule analysis (AR) that can be used as a clinical tool for assessing a patient’s with Acquired Brain Injury ability to organize [57]. Nishtala et al. (2018), applied association

rules analysis to discover medication combinations that contribute to the risk of fractures in older adults [53]. Their analysis pointed out that psychotropic medications and codeine are frequently associated with fractures and they propose this methodology to be applied to big data as a tool for figuring out medication combinations associated with adverse drug events.

Examples of data types massively produced in biomedicine laboratories are genomics data, transcriptomics data (bulk RNA sequencing, single cell RNA sequencing), data on genetic variations (e.g., single nucleotide polymorphism (SNP) data, copy number variation (CNV) data), proteomics data. The use of clustering and classification techniques is common for the analysis of these biological data sets, whereas techniques from association analysis are rarely employed. Few bioinformatics pipelines find and assess the association rules across large amounts of multi-omics experiments datasets. The R-package OmicsARules [7] identifies the concerted changes among genes under association rules mining framework. OmicsARules searches for concurrent patterns among frequently altered genes and can be used in exploring single or multiple omics data across sequencing platforms.

2.5.5 Hyperthermophilic microorganisms

Extremely thermophilic microorganisms are organisms highly adapted to high temperatures considered as “extreme” by human perception. These conditions are the norm under which these organisms are able to metabolically and biochemically operate. The highest documented temperature that microbial life can survive is 130 ◦ C (*Geogemma barossii*) and the highest documented temperature that microbial life can be metabolically active is 122 ◦ C (Methanopyrus kandleri)[49] . The identification, isolation and culture under controlled laboratory conditions of extremophiles has led to numerous advances in molecular biology, biotechnology and medicine. Also the study of the metabolism of thermophilic microorganisms is of great industrial importance since these organisms’ enzymes are able to catalyze reactions of industrial significance at elevated temperatures [76].

Examples of the use of thermophilic microorganisms are: the thermostable DNA polymerases used in the polymerase chain reaction (PCR), various enzymes used in the process of making biofuels, organisms used in biomining-biotechnologies (or bioleaching) for extracting metals and carotenoids used in the food and cosmetic industries. In addition to that, other potential applications include making lactose-free milk, the production of antibiotics, anticancer, and antifungal drugs[9]. Given the usefulness of thermostable enzymes they are considered as powerful assets in industrial catalysis and great effort is being made into incorporating genetically improved thermostable enzymes of thermophiles in the bioprocessing industry. The current status in the use of hyperthermophiles in biotechnology is that their enzymes of interest are routinely produced in recombinant mesophilic hosts to be used as biocatalysts, but in recent years the use of hyperthermophiles as metabolic engineering platforms has also become possible. Various archaea species (e.g., *Sulfolobus, Thermococcus, Pyrococcus* etc.) and bacteria species (e.g., *Caldicellulosiruptor, Thermotoga* etc.) are established as laboratory models and their metabolic pathways have been altered or engineered so that the production of fuels and chemicals at elevated temperatures has become possible[76]. Genomic and metagenomic studies of hyperthermophiles provide useful information for putative biocatalysts.

The list of extremely thermophilic microorganisms that can be cultured and sustained in artificial conditions in the laboratory has been expanded over the past decades, but only a

small subset among them has been studied in detail in terms of genetics and physiology. These organisms are studied with the effort to develop molecular genetics tools that can open up opportunities for metabolic engineering [76]. A list of such microorganisms is presented below.

- *Thermococcus kodakarensis*: is a species of thermophilic archaea and a well established source of thermophilic proteins. It has been used to produce recombinant versions of proteins from other thermophiles and shows promise as a bio-hydrogen production platform.
- *Pyrococcus furiosus*: is a species of thermophilic archaea and the greatest success story so far in metabolic engineering of extreme thermophiles that allows production of non-native product such as lactate, 3-hydroxypropionate (3HP) and butanol.
- *Sulfolobus species* (*S. acidocaldarius, S. solfataricus, and S. islandicus*): are species of thermophilic and acidophilic archaea. They have been used extensively in the study of transcription in archaea, as a model host for archaeal viruses, and as a source of crystallized thermophilic proteins.
- *Thermus thermophilus*: is a species of thermophilic bacteria that has been used to overexpress some of its own proteins and as a source of crystallized thermophilic proteins. It has been metabolically engineered to grow anaerobically by denitrification.
- *Metallosphaera sedula*: is a species of thermophilic and acidophilic archaea with a high tolerance to metal ions. It is a promising candidate for the production of electrofuels and bioleaching operations in high-temperature.
- *Thermoanaerobacter mathranii*: is a species of thermophilic bacteria that is a promising candidate for biofuel production.
- *Caldicellulosiruptor bescii*: is a species of thermophilic bacteria. Genetically engineered strains can produce ethanol and increased quantities of hydrogen. Moreover, a heterologous gene encoding an archaeal tungsten-containing enzyme was successfully expressed in *C. bescii*, so that the organism could assimilate tungsten, a metal rarely used in biological systems.
- *Thermotoga* species (*T. maritima* and *T. neapolitana*): are species of thermophilic bacteria. Strains have been metabolically engineered to express cellulases from Caldicellulosiruptor saccharolyticus giving them cellulolytic activity. Also it can be transformed with an *E. coli* shuttle vector giving kanamycin resistance.

These organisms exhibit unusual and potentially useful native metabolic capabilities, including cellulose degradation, metal solubilization, and RuBisCO-free carbon fixation [76].

3. DATA COLLECTION SURVEY

3.1 Databases about chemicals

In this section we present various publicly available databases about chemicals. These databases include information about chemical structures, physicochemical properties, biological functions. We also present in more detail the STITCH database. The STITCH database includes information about known interactions between chemicals and proteins using text-mining methodologies and aggregating information from other databases. The STITCH database also includes information about predicted interactions between chemicals and proteins (computational prediction, knowledge transfer between organisms). The interactions include direct (physical) and indirect (functional) associations.

- Chemical Entities of Biological Interest (ChEBI) - EMBL-EBI[12] : is a freely available dictionary of 'small' chemical compounds, but genome-encoded macromolecules (nucleic acids, proteins and peptides derived from proteins by cleavage) are not included. ChEBI contains groups (parts of molecular entities) and classes of entities. ChEBI includes the relationships between molecular entities or classes of entities and their parents and/or children (ontological classification).
- PubChem[72]: is a public repository for biological properties of small molecules. PubChem organizes its data into three databases: Substance, Compound and BioAssay. In the Substance database depositor-provided chemical data are stored. In the Compound database unique chemical structures are stored. In the BioAssay database biological assay descriptions and test results are stored. PubChem is used as a 'big data' source in machine learning and data science studies for virtual screening, drug repurposing, chemical toxicity prediction, drug side effect prediction and metabolite identification.
- The UMLS Metathesaurus (`https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html`) contains information about biomedical and health-related concepts, their various names and the relationships among them. UMLS includes:
  - the Chemical Biology and Drug Development Vocabulary (`https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/NCI_CBDD/index.html`)
  - and the Alcohol and Other Drug Thesaurus (`https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/AOD/index.html`).
- DrugBank (`http://www.drugbank.ca/`) combines detailed drug data with drug target information.
- KEGG drug (`http://www.genome.jp/kegg/drug/`) is a chemical structure based information resource for all approved drugs in the US and Japan.
- Metabolic substances KEGG compound (`http://www.genome.jp/kegg/compound/`) is a database for metabolic compounds and other chemical substances that are relevant to biological systems.
- HMDB (`http://www.hmdb.ca/`) contains detailed information about small molecule metabolites found in the human body.

- STITCH [37, 38, 69]: integrates information about interactions of chemicals and interactions of chemicals with proteins from experimental information (crystal structures, binding experiments, high-throughput screens), manually curated databases (biological actions of chemicals, protein binding constants for compounds, metabolic pathways, drug–target relationships) and text-mining. Text mining of MEDLINE and OMIM abstracts, as well as PubMed Central open-access full-text articles (for articles that are available for text mining reuse), is performed with a simple co-occurrence scheme and a more complex natural language processing (NLP) approach. A full-text search is available for identifiers and common names of chemicals and proteins. In order to increase the coverage of the text-mining approach, groups of proteins that are described in MeSH terms are also used as entities during text mining. STITCH creates a united single group of chemicals from PubChem by merging stereo isomers and salt forms of the same molecule into one compound. The interaction types - links between the nodes (chemicals and proteins) are derived (as “actions”) from natural language processing (NLP), pathway and interaction databases. Chemical structures may be entered as SMILES strings to search for similar chemicals that are stored in the database. Chemical structure similarity is used to predict relations between chemicals, so chemical–protein interactions are transferred between species based on the sequence similarity of the proteins.

3.2 Corpora with chemical mentions

In this section we present the various corpora with manually labeled chemical entities that are publicly available in this day:

- CRAFT corpus[3] : consists of 67 full-text biomedical journal articles from PubMed with approximately 100,000 concept annotations to 7 different biomedical ontologies/terminologies (Chemical Entities of Biological Interest, Cell Ontology, Entrez Gene, Gene Ontology: biological process, cellular component, and molecular function, NCBI Taxonomy, Protein Ontology, Sequence Ontology.
- GENIA corpus[29] : consists of 2000 MEDLINE abstracts with more than 400 000 words and almost 100 000 annotations for biological terms belonging to 47 biologically relevant nominal categories.
- PennBioIE CYP 1.0 [44] : consists of 1100 PubMed abstracts on the inhibition of cytochrome P450 enzymes, comprising approximately 274,000 words of biomedical text, tokenized and annotated for paragraph, sentence, part of speech, and five types of biomedical named entities.
- ADE corpus[17] : The Adverse Drug Effect (ADE) Corpus consists of 3 different datasets: DRUG-AE.rel provides relations between drugs and adverse effects, DRUG-DOSE.rel provides relations between drugs and dosages and ADE-NEG.txt provides all sentences in the ADE corpus that do not contain any drug-related adverse effects.
- NLM-Chem corpus[25]: contains 150 full-text journal articles selected both to be rich in chemical mentions.
- BC5CDR corpus [41]: consists of 1500 PubMed articles with 4409 annotated chemicals, 5818 diseases and 3116 chemical-disease interactions.

- DDI corpus [21]: consists of 1025 Documents from two different sources: DrugBank database and MedLine and has been manually annotated with drugs and pharmacokinetics and pharmacodynamics interactions.
- EDGAR corpus [60]: contains annotations for genes, drugs and cells, including binary relationships between genes and drugs, genes and cells, and drugs and cells.
- Metabolites and Enzymes corpus [54]: consists of 296 MEDLINE abstracts that have been manually annotated about metabolites.
- ChEBI Patent Gold Standard corpus [1]: consists of a full set with 400,125 annotations and a harmonized set with 36,537 annotations. Annotation guidelines were developed and 200 full patents from the World Intellectual Property Organization, United States Patent and Trademark Office, and European Patent Office were selected to be annotated.The patents were pre-annotated automatically and made available to four independent annotator groups each consisting of two to ten annotators. The annotators marked chemicals in different subclasses, diseases, targets, and modes of action. A subset of 47 patents was annotated by at least three annotator groups, from which harmonized annotations and inter-annotator agreement scores were derived. All patents and annotated entities are publicly available (www.biosemantics.org).
- Chem EVAL corpus (SCAI corpus) [33]: consists of of 100 abstracts (with 1206 chemical mentions) annotated with chemical entities
- CHEMDNER corpus will be described in detail in the following section.

The aforementioned datasets consist of a relatively small number of abstracts that contain a small number of chemical mentions. The majority of them are not specific for the field of chemical NER since they include more annotations of other biological entities such as genes, proteins, diseases than of chemical entities. Also some are specified to drugs and drug-diseases relations. The disadvantages of the existing corpuses, in combination with the difficulties in giving a specific definition of what a chemical is in the various scientific fields where chemistry is studied and the polymorphism in which the chemical entities are written in text have lead to the creation of the CHEMNDNER dataset.

3.3 CHEMDNER corpus

The CHEMDNER corpus[36] consists of an assembly of 10,000 PubMed abstracts that contain a total of 84,355 chemical entity mentions, corresponding to 19,805 unique strings of chemical names. The chemical entities were manually annotated by expert chemistry literature curators, who followed annotation guidelines specifically defined by experts on the field for this task. The CHEMDNER corpus is designed to be a useful resource not only for chemical named entity recognition but also for the development of chemical text processing software (chemistry-tuned tokenization methods optimized for the correct identification of chemical entities) and of text categorization systems for the selection of documents that contain chemical mentions.

The selection of the 10,000 PubMed abstracts of the CHEMDNER corpus was based on representing all the major chemical disciplines. The chemical entity mentions were

manually labeled and were manually classified to one of the seven predefined structure-associated chemical entity mention (SACEM) classes: Abbreviation (short form of chemical names including abbreviations and acronyms), Family (chemical families with a defined structure), Formula (molecular formulas), Identifier (chemical database identifiers), Multiple (non-continuous mentions of chemicals in text), Systematic (IUPAC names of chemicals) and Trivial (common names of chemicals and trademark names). Examples of Structure Associated Chemical Entity Mentions (SACEMs) which are annotated in the CHEMDNER corpus are the following: 'nitric oxide' (Systematic), 'Aspirin' (Trivial), 'GABA' (Abbreviation), whereas general chemical concepts like 'inactivator' or pigment', molecules with biological roles like 'hormone', 'antibiotic' or 'metabolite' and molecules with reactivity roles like 'nucleophile' or 'chelator' do not qualify as SACEMs and are not annotated in the CHEMDNER corpus. In Figure 3.1 the seven structure-associated chemical entity mention (SACEM) classes introduced are being presented together with a short description and example cases.

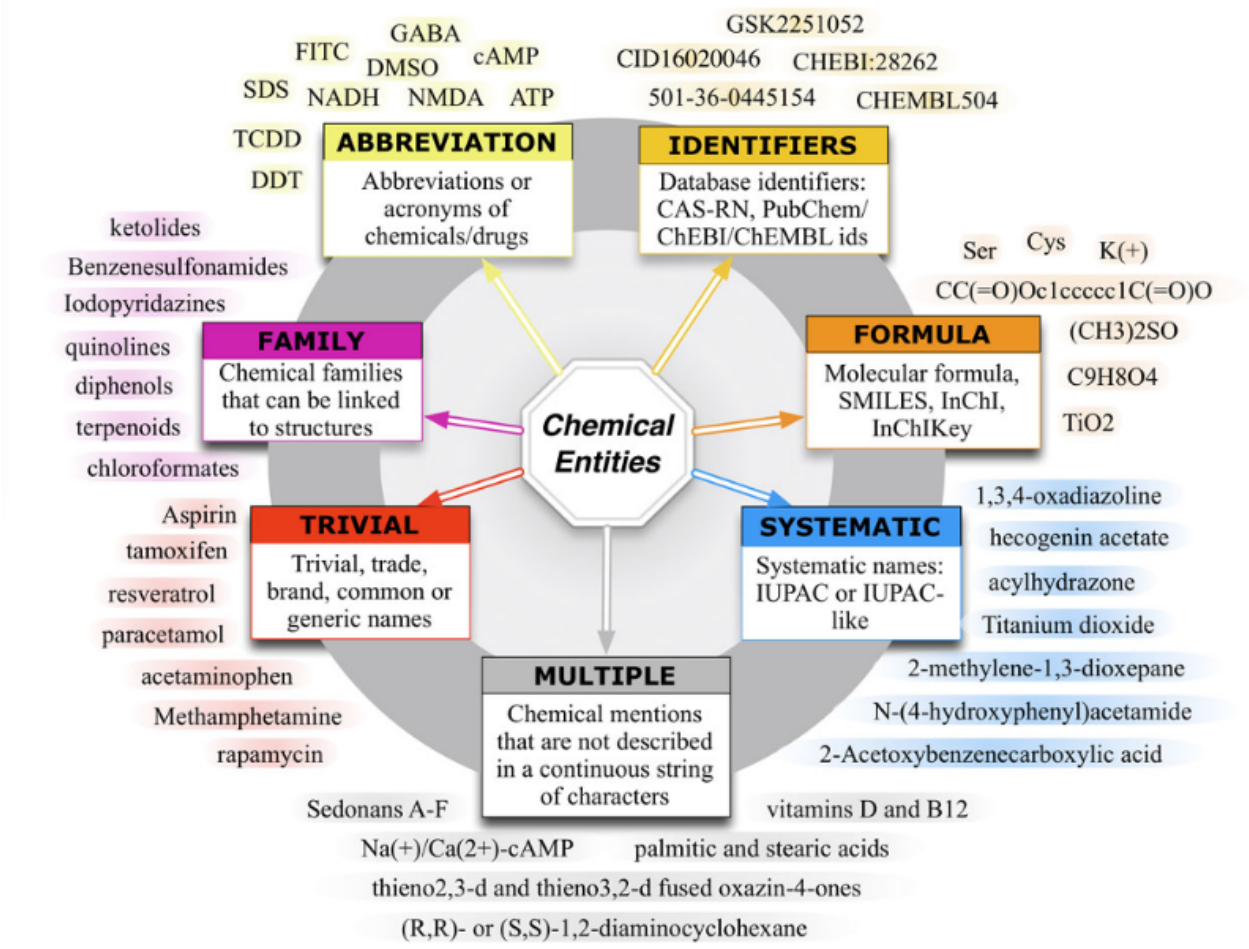

Figure 3.1: CHEMDNER chemical entity mention classification chart (ABBREVIATION, IDENTIFIERS, FORMULA, SYSTEMATIC, MULTIPLE, TRIVIAL, FAMILY) with examples[36].

The abstracts for the CHEMDNER corpus were selected based on their subject category in order for the major chemical disciplines: Biochemistry Molecular Biology, Applied Chemistry, Medicinal Chemistry, Multidisciplinary Chemistry, Organic Chemistry, PhysicalChemistry, Endocrinology Metabolism, Chemical Engineering, Polymer Science, Pharmacology Pharmacy and Toxicology, to be represented in the corpus. The selected abstracts come from the top 100 journals (that had at least 100 articles) from each discipline based on the journals' impact factor. The selected articles were published in 2013 in English in order for the corpus to be representative of modern chemical language, and

their full text is accessible in the PubMed database. The final corpus consists of 10,000 abstracts that belong to the aforementioned subject categories which were randomly selected. The final selection of abstracts was split into three datasets: 3500 (training set), 3500 (development set) and 3000 (evaluation set) abstracts.

The guidelines for the annotation of chemicals and their classification to the SACEM classes were prepared by chemists with a Ph.D., who received feedback from trained literature curators also with a Ph.D. in chemistry. The detailed annotation guidelines are distributed together with the corpora. This is critical for possible future extension of the corpora and to deal with potential future causes of inconsistencies and annotation errors. The annotators had to have a background in chemistry in order to guarantee that the annotations are correct. The annotaters worked on the *AnnotateIt* tool. Nested annotations were not allowed and distinct entity mentions could not overlap.

In order to make sure that during the annotation process the number of missed chemical mentions and wrong annotations was as low as possible, a second group of additional curators also annotated the test set abstracts. The conflicting annotations between the two curator teams were collected and presented to the main curation group for a second round of manual revision. At the end of the harmonization process 1,185 annotations were added to the original 24, 671 test set annotations (4.08%) and 505 (2.05%) were removed, leading to the final harmonized test set that consists of 25,351 annotations.

The CHEMDNER corpus' PubMed abstracts are distributed in a tab-separated text format, with the following three columns: article identifier (PMID, PubMed identifier), title of the article, and abstract of the article. The annotation file has a tab-separated format with columns corresponding to the article identifier, the part of the document processed (T: title, A: abstract), the start and end characters offsets of the chemical, the text string of the chemical entity mention and the corresponding chemical entity mention class.

Figure 3.2 provides an overview of the CHEMDNER corpus in terms of the number of manually revised abstracts (Abstracts) with their total sizes as number of characters and tokens, the number of abstracts containing at least one chemical entity mention (Abstracts with CEM), the number of annotated mentions of chemical entities, the number of unique chemicals annotated (the non-redundant list of mentions) and the number of corresponding journals for the annotated abstracts. The number of mentions for each CHEMDNER entity class (see Figure 1) is provided for each set and the entire corpus in the lower half of the table.

Figure 3.3 provides an overview of the number of abstracts associated with each chemical discipline in the CHEMDNER corpus (Abstracts column). The total number of chemical entity mentions in the abstracts of each chemical discipline (Mentions column). The percentage of chemical entity mentions of each SACEM class AB: ABBREVIATION, FA: FAMILY, FO: FORMULA, ID: IDENTIFIER, MU: MULTIPLE, NO: NO CLASS, SY: SYSTEMATIC, TR: TRIVIAL.

The gradual release of the CHEMDNER corpus datasets was combined with the drugs and chemical names extraction challenge[35] that consisted of two tasks: the indexing of documents with chemicals (chemical document indexing - CDI task), and the identification of exact mentions of chemicals in text (chemical entity mention recognition - CEM task). The CEM task was evaluating the ability to specifically locate within a document every chemical entity mention, by exactly locating their start and end character indices. The participating systems were evaluated on the evaluation set of the CHEMDNER corpus that had 25,351 chemical entity mentions (7,563 unique chemical names). The evaluation metrics for comparing the performance of the systems that participated in the challenge

	Training set	Development set	Test set	Entire corpus
Abstracts	3,500	3,500	3,000	10,000
Nr. characters	4,883,753	4,864,558	4,199,068	13,947,379
Nr. tokens	770,855	766,331	662,571	2,199,757
Abstracts with SACEM	2,916	2,907	2,478	8,301
Nr. mentions	29,478	29,526	25,351	84,355
Nr. chemicals	8,520	8,677	7,563	19,805
Nr. journals	193	188	188	203
TRIVIAL	8,832	8,970	7,808	25,610
SYSTEMATIC	6,656	6,816	5,666	19,138
ABBREVIATION	4,538	4,521	4059	13,118
FORMULA	4,448	4,137	3,443	12,028
FAMILY	4,090	4,223	3,622	11,935
IDENTIFIER	672	639	513	1,824
MULTIPLE	202	188	199	589
NO CLASS	40	32	41	113

Figure 3.2: CHEMDNER corpus overview[36].

Chem. subject categories	Abstracts	Mentions	AB	FA	FO	ID	MU	NO	SY	TR
PHARMACOLOGY	1,983	23,368	18.81	10.54	6.42	4.93	0.64	0.29	17.28	41.09
MEDICINAL CHEMISTRY	1,957	17,543	10.00	21.11	8.00	2.10	1.56	0.12	25.88	31.23
ORGANIC CHEMISTRY	1,893	22,622	18.77	10.56	6.56	5.00	0.63	0.30	17.43	40.74
TOXICOLOGY	1,664	21,608	20.82	10.59	14.16	1.35	0.46	0.13	22.68	29.81
MULTIDISCIPL. CHEM.	1,217	11,892	14.38	12.15	27.97	0.52	0.55	0.13	25.62	18.67
PHYSICAL CHEMISTRY	997	9,682	12.14	9.81	36.39	0.27	0.43	0.15	27.57	13.24
BIOCHEMISTRY	879	6,503	18.75	16.55	14.24	1.12	0.34	0.11	23.17	25.73
APPLIED CHEMISTRY	843	7,759	8.48	24.45	7.71	0.17	1.37	0.10	24.99	32.74
ENDOCRINOLOGY	652	5,484	14.66	16.01	9.87	1.33	0.15	0.15	20.13	37.71
POLYMER SCIENCE	232	1,999	33.82	17.26	6.50	0.05	0.10	0.00	25.86	16.41
CHEMICAL ENGINEERING	3	42	0.00	0.00	38.10	0.00	0.00	0.00	61.90	0.00

Figure 3.3: CHEMDNER abstracts, split into chemical disciplines[36].

were:

- Recall: the percentage of correctly labeled positive results over all positive cases (a measure of a systems ability to identify positive cases.

$$r = \frac{TP}{TP + FN}$$

- Precision: the percentage of correctly labeled positive results over all positive labeled results (a measure of a classifier's reproducibility of the positive results).

$$p = \frac{TP}{TP + FP}$$

- The balanced F-measure: a parameter for the relative importance of precision over recall.

$$f1 - score = \frac{2 \times p \times r}{p + r}$$

  False negative (FN) results correspond to incorrect negative predictions (cases that were part of the CHEMDNER annotations, but missed by the automated systems), False positive (FP) results correspond to incorrect positive predictions (wrong results

predicted by the systems that had no corresponding annotation in the CHEMDNER annotations) and True positive (TP) results correspond to correct positive predictions (correct predictions matching exactly with the CHEMDNER annotations).

The results of the top performing teams in the CEM task are presented in figure 3.4. Top scoring team of the CEM task obtained an F-score of 87.39%. They developed a hybrid strategy that integrated a machine learning based approach based on CRF models, a dictionary based approach to find special types of mentions such as chemical formula and sequences of amino acids and an abbreviation detection method.

Team	*P*	*R*	F_1
173	89.09%	85.75%	87.39%
231	89.10%	85.20%	87.11%
179	88.73%	85.06%	86.86%
184	92.67%	81.24%	86.58%
198	91.08%	82.30%	86.47%
197	86.50%	85.66%	86.08%
192	89.42%	81.08%	85.05%
233	88.67%	81.17%	84.75%
185	84.45%	80.12%	82.23%
245	84.82%	72.14%	77.97%
199	85.20%	71.77%	77.91%
222	85.83%	71.22%	77.84%
259	88.79%	69.08%	77.70%
214	89.26%	68.08%	77.24%
262	78.28%	74.59%	76.39%
263	82.14%	70.94%	76.13%

Figure 3.4: Chemical Entity Mention recognition (CEM) task evaluation results[35].

4. CHEMICAL NAMED ENTITY RECOGNITION

In this chapter, we present a novel NER tool with the capability to identify mentions of chemical entities in biomedical text. The Named Entity Recognition tool is developed using spaCy [50] (`https://spacy.io/`). spaCy is a Python library for advanced Natural Language Processing. spaCy provides a variety of user friendly and practical tools to build information extraction or natural language understanding systems, including pre-trained Neural Network (NN) models for part-of-speech tagging, dependency parsing, named entity recognition, etc. and it makes it easy to train existing pipelines or create new NLP pipelines. In the development of the chemical Named Entity Recogniser, pipeline packages from scispaCy [52] were also deployed. scispaCy (`https://allenai.github.io/scispacy/`) is a specialized NLP Python library for processing biomedical text with increasing popularity among scientists in the biomedical domain.

Various corpora with mentions of chemical entities are available (CRAFT corpus [3] , GENIA corpus [29], PennBioIE CYP 1.0[44], ADE corpus[17] , NLM-Chem corpus[25], BC5CDR corpus [41], DDI corpus[21], EDGAR corpus [60], Metabolites and Enzymes corpus[54], ChEBI Patent Gold Standard corpus[1], Chem EVAL corpus (SCAI corpus) [33], CHEMDNER corpus [36]). In this task the selected corpus for training the neural network developed using the spaCy library, is the CHEMDNER corpus. The CHEMDNER corpus is the largest manually annotated, publicly available, easy to use corpus, with millions of annotated chemical entities (and only chemical entities) on recent papers on various domains of Chemistry (it is representative of modern chemical language). The corpus is divided into a training, a development and a evaluation set. Each subset is publicly available in the form of two text files:

- One text file lists:
  - the PubMed IDs,
  - the abstracts of the selected publications (of various scientific domains of Chemistry),
  - and the publications' titles.
- and the other text file includes the annotated chemical entities in each abstract. More specifically it includes the following information about each manually tagged chemical entity:
  - the article's Pubmed ID where each chemical entity is manually tagged,
  - whether it is located in the title or in the abstract of the article,
  - the start and end position of the entity in the text,
  - the annotated chemical entity,
  - and the class that it has been assigned to: Abbreviation (short form of chemical names including abbreviations and acronyms), Family (chemical families with a defined structure), Formula (molecular formulas), Identifier (chemical database identifiers), Multiple (non-continuous mentions of chemicals in text), Systematic (IUPAC names of chemicals) or Trivial (common names of chemicals and trademark names).

The supplementary files of the CHEMDNER corpus also include the following information about each publication's abstract:

- to which chemical discipline (or chemical disciplines) each abstract belongs to. The selected articles were classified to one (or more) of the following chemical disciplines: Biochemistry, Applied Chemistry, Medicinal Chemistry, Multidisciplinary Chemistry, Organic Chemistry, Physical Chemistry, Chemical Endocrinology Engineering, Pharmacology, Polymer Science, Toxicology.
- whether each abstract includes or not chemical entities (if it belongs to the true negative dataset or not).

The BioCreative IV community challenge [35] focused on promoting the development of systems for the automatic recognition of chemical entities in text. Two tasks were organised during the challenge: the chemical document indexing - CDI task and the chemical entity mention recognition - CEM task. The performance of the participating teams in the CEM task provides a measure for comparing the performance of the chemical Named Entity Recogniser that is developed in this project. The comparison with the state-of-the-art chemical Named Entity Recognising tools will be based on their performance on the evaluation set of the CHEMDNER corpus (based on their precision, recall and f1-score).

4.1 Development of the chemical Named Entity Recogniser

4.1.1 Training/Test Dataset Configuration

The CHEMDNER corpus contains 7000 PubMed abstracts categorised as training (3500 abstracts) and development set (3500 abstracts). In total, the abstract have 17,197 mentions of unique chemicals (59,004 mentions in total taking into account multiple mentions of several chemicals). In the dataset of the 7000 PubMed abstracts, a small portion (1,177 abstracts) does not have any mention of chemical entities at all (true negative dataset). The CHEMDNER corpus also contains 3000 PubMed abstracts categorised as evaluation set.

First, we attempted to choose the best proportion of the CHEMDNER's training and development datasets that should be used as the training and the test set while training the spaCy model. In order to do so we performed three sets of experiments:

- CHEMDNER's training (3500 abstracts) and development set (3500 abstracts) were used for training and testing the scispaCy pipelines. The performance of trained model was then evaluated using the
- The 7000 PubMed abstracts from CHEMDNER corpus' training and development datasets were randomized and split into 80% for training and 20% for testing.
- The 7000 PubMed abstracts from CHEMDNER corpus' training and development datasets were randomized and split into 90% for training and 10% for testing.

The three models: en_core_sci_md, en_core_sci_lg and en_core_sci_scibert were selected for training using the CHEMDNER corpus. With the three models trainings were performed:

- one with the development and training set as provided by the CHEMDNER corpus,

- one with the 7000 training and development abstracts of the CHEMDNER corpus randomly divided in 80% of the abstracts for training and 20% for testing, and
- one with the 7000 training and development abstracts of the CHEMDNER corpus randomly divided in 90% of the abstracts for training and 10% for testing.

The processing pipeline of the en_core_sci_md model and the en_core_sci_lg model consist of a tokenizer and an entity recogniser (figure 4.1). The entity recogniser component was trained on the CHEMDNER corpus.

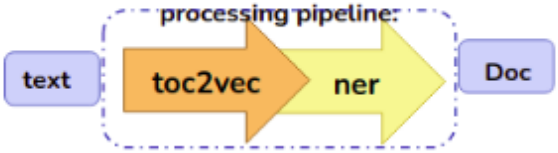

Figure 4.1: The processing pipeline of the en_core_sci_md/lg models that consist of a tokenizer and an entity recogniser.

4.1.1.1 Results using the en_core_sci_md model.

The first set of experiments was performed using the en_core_sci_md model. First the ner component of the pipeline was trained on the training set as provided by the CHEMDNER corpus. While training the development set (as provided by the CHEMDNER corpus) was used for testing the model's performance. The precision, recall and f1-score are measured on the test set (development set) while training for 8 epochs with batch size 128 and are visualised in figure 4.2. The loss value count on the test set (development set) while training is visualised in figure 4.3. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.1. The results of the overall performance of the model are presented in table 4.2.

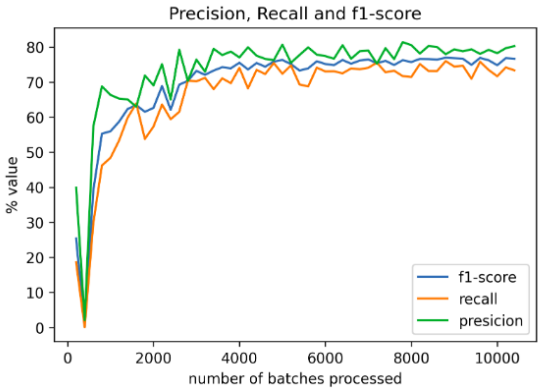

Figure 4.2: % values of p, r and f1-score while training the ner of the en_core_sci_md model on the CHEMDNER corpus. Training lasted for 8 epochs (batch size 128).

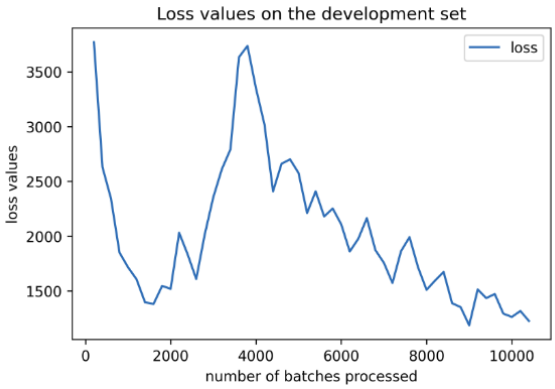

Figure 4.3: The loss values while training the ner component of the en_core_sci_md model on the CHEMDNER corpus. Training lasted 8 epochs with batch size 128.

Table 4.1: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_md model trained on the CHEMDNER training set.

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	46.00	74.35	56.84
FAMILY	62.31	57.59	59.86
TRIVIAL	76.66	77.12	76.38
FORMULA	5.27	76.97	9.87
MULTIPLE	0.65	10.52	1.22
ABBREVIATION	46.02	76.87	57.57
IDENTIFIER	87.50	70.00	77.77

Table 4.2: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER training set.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
30.49	71.14	42.68

Then the 7000 training and development abstracts of the CHEMDNER corpus were randomly divided in 80% of them for training and 20% of them for testing the model's performance while training. The precision, recall and f1-score are measured on the test set while training for 8 epochs with batch size 128 and are visualised in figure 4.4. The loss value count on the test set (development set) while training is visualised in figure 4.5. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.3. The results of the overall performance of the model are presented in table 4.4.

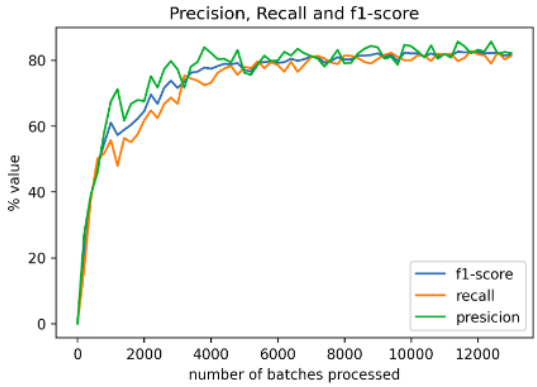

Figure 4.4: % values of p, r, f1-score while training the ner of the en_core_sci_md model on the CHEMDNER corpus (80% training set - 20% test set). Training lasted for 8 epochs (batch size 128).

Figure 4.5: The loss values while training the ner component of the en_core_sci_md model on the CHEMDNER corpus (80% training set - 20% test set). Training took place for 8 epochs with batch size 128.

Then, the 7000 training and development abstracts of the CHEMDNER corpus were randomly divided in 90% of them for training and 10% of them for testing the model's performance while training. The precision, recall and f1-score are measured on the test set while training for 4 epochs with batch size 128 and are visualised in figure 4.6. The loss value count on the test set (development set) while training is visualised in figure 4.11. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying

Table 4.3: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_md model trained on the CHEMDNER corpus (80% training set - 20% test set).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	59.33	81.55	68.69
FAMILY	72.55	66.34	69.31
TRIVIAL	82.96	76.35	79.52
FORMULA	70.39	76.36	73.26
MULTIPLE	8.82	31.58	13.79
ABBREVIATION	89.68	65.32	75.59
IDENTIFIER	81.25	65.00	72.22

Table 4.4: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER corpus (80% training set - 20% test set).

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
70.47	73.85	72.12

each entity and correctly assigning it to each SACHEM class, are given in the table 4.5. The results of the overall performance of the model are presented in table 4.6.

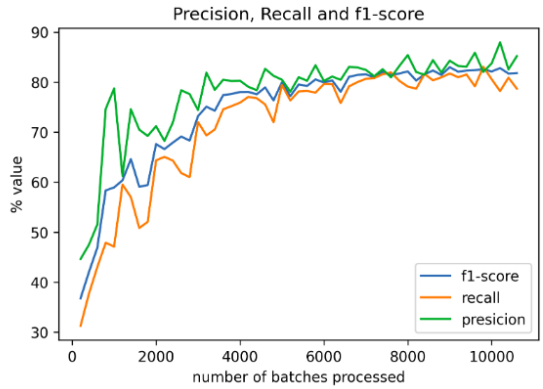

Figure 4.6: % values of p, r, f1-score while training the ner of the en_core_sci_md model on the CHEMDNER corpus (90% training set - 10% test set). Training lasted for 4 epochs (batch size 128).

Figure 4.7: The loss values while training the ner component of the en_core_sci_md model on the CHEMDNER corpus (90% training set - 10% test set). Training took place for 4 epochs with batch size 128.

Table 4.5: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_md model trained on the CHEMDNER corpus (90% training set - 10% test set).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	54.74	75.65	63.52
FAMILY	58.58	64.40	61.35
TRIVIAL	70.37	79.05	74.46
FORMULA	57.14	75.15	64.92
MULTIPLE	50.00	42.11	45.71
ABBREVIATION	81.25	75.14	78.08
IDENTIFIER	80.00	40.00	53.33

Table 4.6: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER corpus (90% training set - 10% test set).

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
62.76	73.44	67.68

4.1.1.2 Results using the en_core_sci_lg model.

The second set of experiments was performed using the en_core_sci_lg model. The en_core_sci_lg model has a larger vocabulary and 600k word vectors, therefore needs a larger training dataset for training. First the 7000 training and development abstracts of the CHEMDNER corpus were randomly divided in 80% of them for training and 20% of them for testing the model's performance while training. The precision, recall and f1-score are measured on the test set while training for 5 epochs with batch size 128 and are visualised in figure 4.8. The loss value count on the test set (development set) while training is visualised in figure 4.9. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.7. The results of the overall performance of the model are presented in table 4.8.

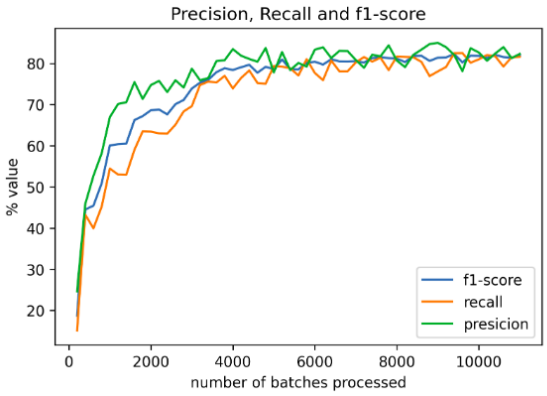

Figure 4.8: % values of p, r, f1-score while training the ner of the en_core_sci_lg model on the CHEMDNER corpus (80% training set - 20% test set). Training lasted for 5 epochs (batch size 128).

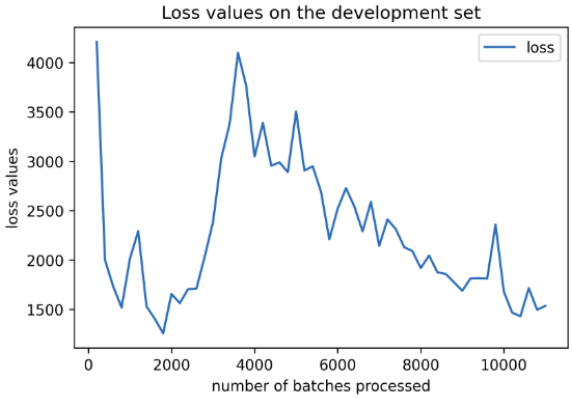

Figure 4.9: The loss values while training the ner component of the en_core_sci_lg model on the CHEMDNER corpus (80% training set - 20% test set). Training took place for 5 epochs with batch size 128.

Table 4.7: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_md model trained on the CHEMDNER corpus (80% training set - 20% test set).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	54.27	79.70	64.57
FAMILY	72.57	63.81	67.91
TRIVIAL	65.35	82.65	72.99
FORMULA	9.96	80.61	17.73
MULTIPLE	26.32	26.32	26.32
ABBREVIATION	81.61	82.08	81.84
IDENTIFIER	75.00	75.00	75.00

And then the 7000 training and development abstracts of the CHEMDNER corpus were randomly divided in 90% of them for training and 10% of them for testing the model's

Table 4.8: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER corpus (80% training set - 20% test set).

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
57.62	60.25	58.91

performance while training. The precision, recall and f1-score are measured on the test set while training for 5 epochs with batch size 128 and are visualised in figure 4.6. The loss value count on the test set (development set) while training is visualised in figure 4.11. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.5. The results of the overall performance of the model are presented in table 4.6.

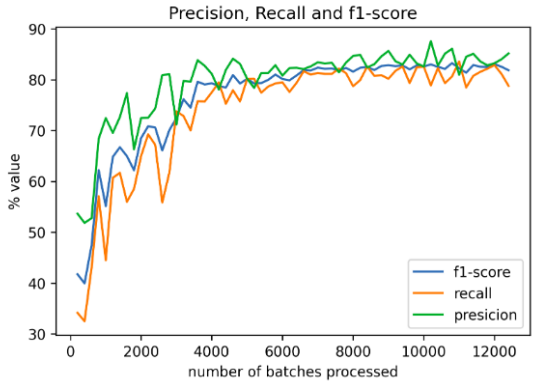

Figure 4.10: % values of p, r, f1-score while training the ner of the en_core_sci_md model on the CHEMDNER corpus (90% training set - 10% test set). Training lasted for 5 epochs (batch size 128).

Figure 4.11: The loss values while training the ner component of the en_core_sci_md model on the CHEMDNER corpus (90% training set - 10% test set). Training took place for 5 epochs with batch size 128.

Table 4.9: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_lg model trained on the CHEMDNER corpus (90% training set - 10% test set).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	54.85	82.47	65.88
FAMILY	48.18	66.93	56.03
TRIVIAL	69.49	75.84	72.53
FORMULA	75.48	70.91	73.12
MULTIPLE	14.29	26.32	18.52
ABBREVIATION	82.69	74.57	78.42
IDENTIFIER	69.57	80.00	74.42

Table 4.10: The overall precision, recall and f1 -score of the en_core_sci_lg model trained on the CHEMDNER corpus (90% training set - 10% test set).

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
59.99	74.44	66.44

4.1.1.3 Results using the en_core_sci_scibert model.

The last set of experiments in this section was performed using the en_core_sci_scibert model. The processing pipeline of the en_core_sci_scibert model consists of a transformer and an entity recogniser (figure 4.12). The entity recogniser component was trained on the CHEMDNER corpus.

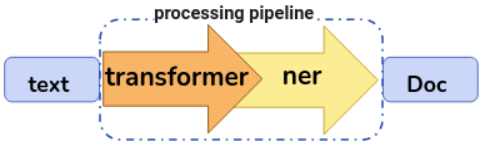

Figure 4.12: The processing pipeline of the en_core_sci_scibert model that consist of a ttransformer and an entity recogniser.

First the ner component of the pipeline was trained on the training set as provided by the CHEMDNER corpus. While training the development set (as provided by the CHEMDNER corpus) was used for testing the model's performance. The precision, recall and f1-score are measured on the test set (development set) while training for 9 epochs with batch size 128 and are visualised in figure 4.13. The loss value count on the test set (development set) while training is visualised in figure 4.14. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.11. The results of the overall performance of the model are presented in table 4.12.

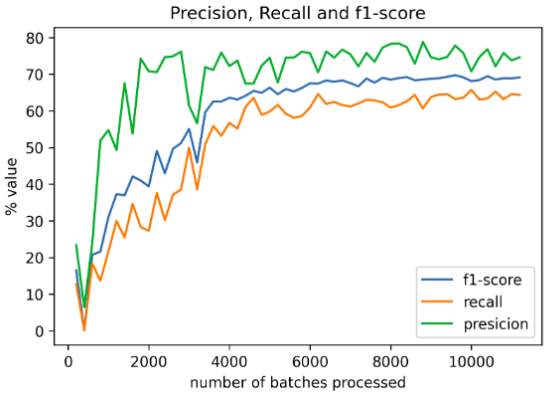

Figure 4.13: % values of p, r, f1-score while training the ner of the en_core_sci_md model on the CHEMDNER corpus. Training lasted for 9 epochs (batch size 128).

Figure 4.14: The loss values while training the ner component of the en_core_sci_md model on the CHEMDNER corpus. Training took place for 9 epochs with batch size 128.

Then the 7000 training and development abstracts of the CHEMDNER corpus were randomly divided in 80% of them for training and 20% of them for testing the model's performance while training. The precision, recall and f1-score are measured on the test set while training for 5 epochs with batch size 128 and are visualised in figure 4.15. The loss value count on the test set (development set) while training is visualised in figure 4.16. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.13. The results of the overall performance of the model are presented in table 4.14.

Table 4.11: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_scibert model trained on the CHEMDNER training set.

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	43.00	65.68	51.97
FAMILY	69.01	51.56	59.02
TRIVIAL	64.22	59.51	61.77
FORMULA	57.21	74.55	64.74
MULTIPLE	0	0	0
ABBREVIATION	76.67	66.47	71.21
IDENTIFIER	75.00	60.00	66.67

Table 4.12: The overall precision, recall and f1 -score of the en_core_sci_scibert model trained on the CHEMDNER training set.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
57.62	60.25	58.91

Table 4.13: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_scibert model trained on the CHEMDNER corpus (80% training set - 20% test set).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	62.78	73.43	67.69
FAMILY	65.56	57.78	61.43
TRIVIAL	76.96	63.11	69.35
FORMULA	71.60	70.30	70.95
MULTIPLE	33.33	21.05	25.81
ABBREVIATION	78.34	71.10	74.55
IDENTIFIER	88.24	75.00	81.08

Table 4.14: The overall precision, recall and f1 -score of the en_core_sci_scibert model trained on the CHEMDNER corpus (80% training set - 20% test set).

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
69.66	65.22	67.37

Transformer-based pipelines are not perfect for every use-case. Even though transformer models have been breaking new accuracy records every month, it's not easy to apply them directly to most practical problems. Also transformer architectures are not designed to operate efficiently on CPU, so it is recommended to have a GPU available for both training and usage. In our applications training was too slow.

4.1.1.4 Initial Chemical NER Result Summary

In table 4.15 we present an overview of the experimental results in order:

- to assess which is the best scispaCy pipeline (en_core_sci_md, en_core_sci_lg, en_core_sci_scibert) for the amount of training data available in the CHEMDNER corpus, and
- to assess which is the best porpotion of the CHEMDNER's training and development set (7000 abstracts) to be used as training set and test set for thraining the aforementioned models.

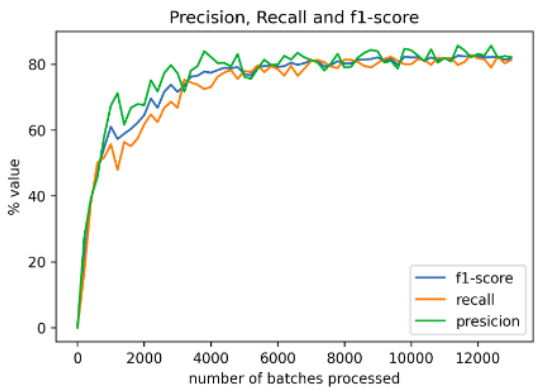

Figure 4.15: % values of p, r, f1-score while training the ner of the en_core_sci_scibert model on the CHEMDNER corpus (80% training set - 20% test set). Training lasted for 5 epochs (batch size 128).

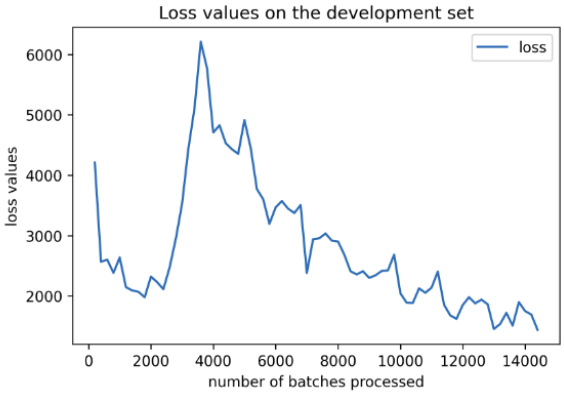

Figure 4.16: The loss values while training the ner component of the en_core_sci_scibert model on the CHEMDNER corpus (80% training set - 20% test set). Training took place for 5 epochs with batch size 128.

The best f-score was obtained using the pipeline with the smaller vocabulary and 50k word vectors (the en_core_sci_md pipeline) and using 80% of the CHEMDNER training and development set as the training set and 20% as the test set. The pipeline with the larger vocabulary and 600k word vectors (the en_core_sci_lg pipeline) gave better results when trained using 90% of the CHEMDNER training and development set as the training set and 10% as the test set, but still the f1-score was lower that the best score of the en_core_sci_md. This model needs a larger training dataset for better performance on the CHEMDNER's evaluation set. The en_core_sci_scibert model when trained using 80% of the CHEMDNER training and development set as the training set and 20% as the test set gave better results (f1-score = 67.37) than when trained using 50% of the CHEMDNER training and development set as the training set and 50% as the test set. The problem with the en_core_sci_scibert model was that since we trained using a CPU, training was too slow (would last for days). Training a transformer-based model without a GPU is too slow.

Table 4.15: Presentation of the training experiments done to evaluate the best proportions of the training and the development datasets of the CHEMDNER corpus.

scispaCy model	Percentage of the CHEMDNER corpus for training and testing	f1-score on CHEMDNER evaluation set
en_core_sci_md	50% : 50%	42.68 %
en_core_sci_md	80% : 20%	72.12%
en_core_sci_md	90% : 10%	67.68 %
en_core_sci_lg	80% : 20%	56.66 %
en_core_sci_lg	90% : 10%	66. 44 %
en_core_sci_scibert	50% : 50%	58.91 %
en_core_sci_scibert	80% : 20%	67.37 %

4.1.2 The role of dictionaries

We investigated whether we can boost the performance of the best model so far (the en_core_sci_md model trained using 80% of the CHEMDNER training and development set as the training set and 20% as the testing set). We combined the statistical model approach with the usage of a dictionary of chemical molecules. The dictionary is created

by combining the ChEBI dictionary (50085 chemicals) with the KEGG compound dictionary (681 chemicals that do not already exist in ChEBI). Overall the dictionary consists of 50766 chemicals.

Adding a dictionary to a spaCy's pipeline can be done using the Entity Ruler pipeline component (`https://spacy.io/api/entityruler`) for rule-based named entity recognition. The Entity Ruler is a component that lets the user add named entities based on pattern dictionaries, which makes it easy to combine rule-based and statistical named entity recognition for even more powerful pipelines. The dictionary lists match patterns. A match pattern includes a label that in this case we assigned as the ChEBI or the KEGG compound ID and the actual pattern to be matched in the text, for example: 'label': 'ChEBI ID: 598', 'pattern': '1-alkyl-2-acylglycerol'.

The processing pipeline consists of the "tok2vec", the "ner" and the "entity_ruler" and it is presented in figure 4.17. The "entity_ruler" will only add new entities that match to the patterns only if they don't overlap with existing entities predicted by the statistical model.

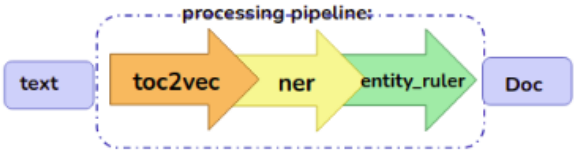

Figure 4.17: Processing pipeline that consists of the following components: toc2vec, ner, entity_ruler.

In order to asses whether adding a dictionary will boost the performance of the chemical NER model or not, we added the dictionary to the pipeline of the model with the best results so far. The dictionary was added to the en_core_sci_md model that was trained on the CHEMDNER corpus (80% of CHEMDNER's training and development set was used for training and 20% for testing). This models results were presented in tables 4.3 and 4.4. The overall precision, recall and f1 -score of the model after adding the dictionary is presented in table 4.16

Table 4.16: Precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER corpus (80% for training, 20% for testing) (left) and the precision, recall and f1 -score of the same model after adding a dictionary to the pipeline.

Processing pipeline	tok2vec,ner	tok2vec,ner,entity_ruler
NER PRECISION (%)	70.47	67.81
NER RECALL (%)	73.85	73.85
NER F1-SCORE (%)	72.12	70.70

Overview of results. It is observed that precision (and F1 score) is decreased, while at the same time the recall remains the same. This means that, in this dataset, the addition of the "entity_ruler" component has increased the false positive entities identified by the model. The chemical NER based only on dictionaries has the disadvantage of often identifying as chemical entities parts of other entities. For example, enzymes that are usually phrases might be splitted, and a part of them to be tagged as a chemical entity. In order to avoid adding more false positives, the "entity_ruler" component will not be used.

4.1.3 Training excluding true negative abstracts

In the spaCy guidelines for training the “ner” component of spaCy’s models it is stated that the model should also be presented to examples that do not have the entities that the user is interested in “teaching” the model. These examples should be included in the training process and be annotated as training data that do not include the entity of interest. In the CHEMDNER corpus (see section 3.3), in the dataset of the 7000 PubMed abstracts (labeled as training and development set) a small portion (1,177 abstracts) does not have any mention of chemical entities at all (true negative dataset). We investigated whether including a true negative dataset in the training process improved the models’ precision and recall or not. From the CHEMDNER training and development datasets the 1,177 abstracts that do not have any mention of chemical entities at all (true negative dataset) are removed. The remaining 5,823 abstracts were randomly split into 90% for training and 10% for development. The “ner” component of the en_core_sci_md model was trained.

The precision, recall and f1-score are measured on the test set (development set) while training for 5 epochs with batch size 128 are visualised in figure 4.18. The loss value count on the test set (development set) while training is visualised in figure 4.19. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.17. The results of the overall performance of the model are presented in table 4.18.

Figure 4.18: % values of p, r, f1-score while training the ner of the en_core_sci_md model on the CHEMDNER corpus without the true negative dataset. Training lasted for 5 epochs (batch size 128).

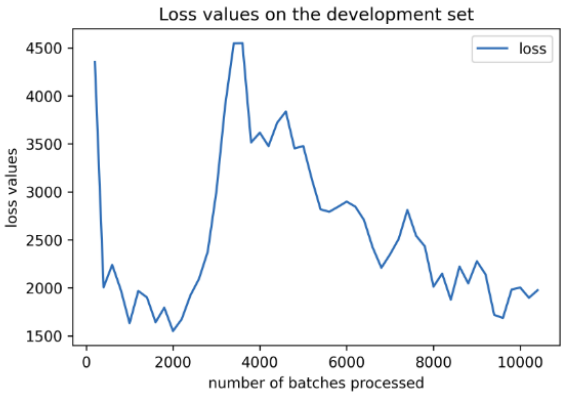

Figure 4.19: The loss values while training the ner component of the en_core_sci_md model on the CHEMDNER corpus without the true negative dataset. Training took place for 5 epochs with batch size 128.

Table 4.17: Precision, recall and f1-score on each chemical entity class of the en_core_sci_md model trained on the CHEMDNER corpus without using the true negative dataset.

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	44.21	81.00	57.20
FAMILY	70.98	60.89	65.55
TRIVIAL	73.17	77.12	75.09
FORMULA	71.43	81.82	76.27
MULTIPLE	30.00	31.58	30.77
ABBREVIATION	80.23	82.08	81.14
IDENTIFIER	88.24	75.00	81.08

Table 4.18: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER corpus without using the true negative dataset.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
62.08	74.53	67.73

Overview of results. The results of the en_core_sci_md model trained using the true negative subset of the CHEMDNER corpus showed better results (tables 4.4. It was observed that using a true negative dataset while training indeed improved the precision of the model and its f1-score.

4.1.4 CNN tuning

In this section, we present our investigation on how the width of the hidden layer of the ”ner” component of the model’s pipeline affects the overall performance of the model. The default architecture of the ”ner” component of the pipeline is a Convolutional Neural Network (CNN) with depth (the number of convolutional layers) four and with width of the hidden layers (the number of neurons in the hidden layer) 64 neurons.

In the following experiments, we set the width of the hidden layers to 84, 100, 128, 164. The en_core_sci_md models’ ”ner” component of the pipeline was trained on the CHEMDNER corpus training and development dataset (80% was used for training and 20% for evaluation). The evaluation results of each model on the CHEMDNER corpus evaluation set are presented in table 4.19 and visualised in figure 4.20.

Table 4.19: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the the CHEMDNER corpus (80% training set and 20% test). Results are from different training experiments where the width of the hidden layer of the ‘ner’ component of the pipeline was set to 64 (default value), 84, 100, 128 and 164.

number of neurons	64	84	100	128	164
NER PRECISION (%)	70.47	74.41	73.99	75.20	74.47
NER RECALL (%)	73.85	76.06	76.06	76.96	77.33
NER F1-SCORE (%)	72.12	75.23	75.01	76.07	75.87

We present the training results of the best model when the width of the hidden layer was adjusted to 128 neurons. The precision, recall and f1-score are measured on the test set (development set) while training for 7 epochs with batch size 128 and are visualised in figure 4.21. The loss value count on the test set (development set) while training is visualised in figure 4.22. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.20. The results of the overall performance of the model are presented in table 4.21.

Overview of results. The performance of the model is improved when the width of the hidden layer is 128 neurons (precision: 75.20%, recall: 76.96%, f1-score: 76.07%).

4.1.5 Boosting NER using tagging/parsing

In the previous training experiments the used model’s processing pipeline consisted of the ”tok2vec” and the ”ner” components. In this section, we examine if adding components

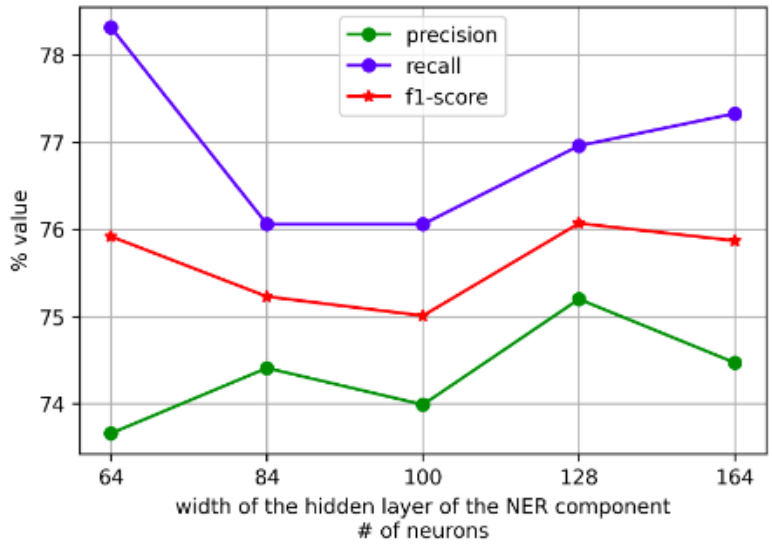

Figure 4.20: %values of p, r, f1-score of the en_core_sci_md model trained on the the CHEMDNER corpus (80% training set and 20% test). Results from different training experiments where the width of the hidden layer of the 'ner' component of the pipeline was set to 64 (default value), 84, 100, 128 and 164.

Table 4.20: Precision, recall and f1 -score on each chemical entity class of the en_core_sci_md model trained on the CHEMDNER corpus (80% training set and 20% testing set). Training took place for 7 epochs with batch size 128. The width of the hidden layer is 128.

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	64.64	82.29	72.40
FAMILY	84.84	67.51	75.19
TRIVIAL	79.28	82.13	80.68
FORMULA	69.32	73.94	71.55
MULTIPLE	50.00	47.37	48.65
ABBREVIATION	84.31	74.57	79.14
IDENTIFIER	85.71	60.00	70.59
MULTIPLE	50.00	47.37	48.65

Table 4.21: The overall precision, recall and f1 -score of the en_core_sci_md model (hidden layer width 128) trained on the CHEMDNER corpus.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
74.47	77.33	75.87

that also take into account the grammatical relations between the tokens improves the model's performance. In this training, the pipeline consists of the following components: "tok2vec", "tagger", "parser", "ner".

- scispaCy's standard "tok2Vec" generates the contextual embeddings for the input tokens.
- The "tagger" and "parser" process the grammatical relations between the tokens.
- The "ner" component of the pipeline is trained on the CHEMDNER corpus.

This experiment was performed using the en_core_sci_md model. The "ner" component of the pipeline was trained on the CHEMDNER corpus. While training the test set was used

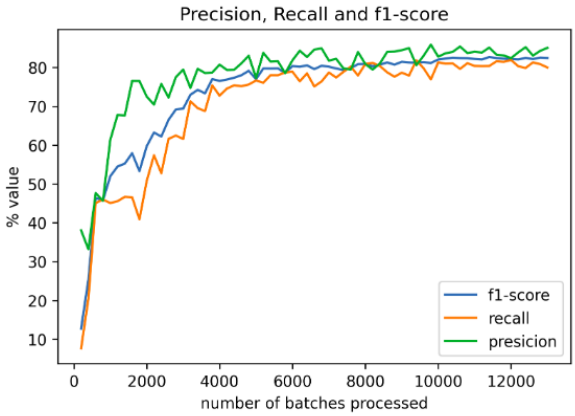

Figure 4.21: % values of p, r, f1-score while training the ner of the en_core_sci_md model (hidden layer width 128) on the CHEMDNER corpus (80% training set and 20% testing set). Training lasted for 7 epochs (batch size 128).

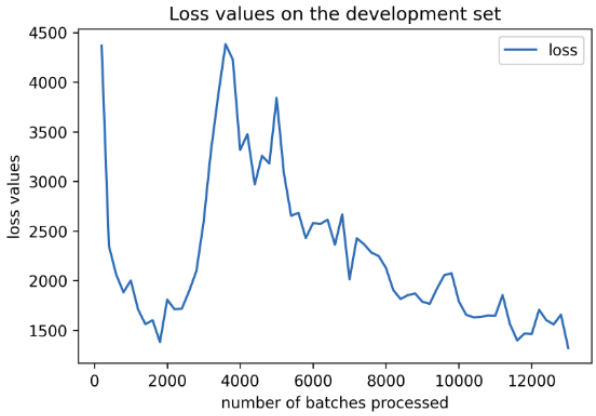

Figure 4.22: The loss values while training the ner component of the en_core_sci_md model (hidden layer width 128) on the CHEMDNER corpus (80% training set and 20% testing set). Training took place for 7 epochs with batch size 128.

for testing the model's performance. 80% of the CHEMDNER's training and development set was used for training and 20% for testing. The precision, recall and f1-score are measured on the test set (development set) while training for 10 epochs with batch size 1000 and are visualised in Figure 4.23. The loss value count on the test set while training is visualised in figure 4.24. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.22. The results of the overall performance of the model are presented in table 4.23.

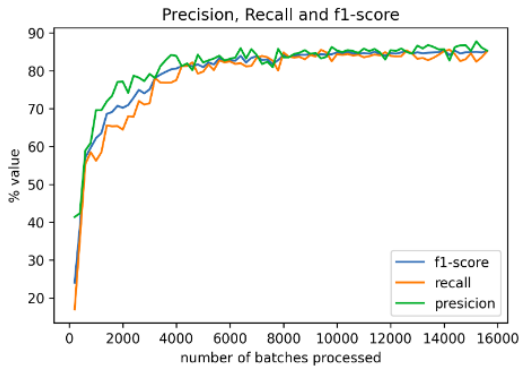

Figure 4.23: % values of p, r, f1-score while training the ner of the en_core_sci_md model(processing pipeline: tok2vec, tagger, parser, ner) on the CHEMDNER corpus (80% train set, 20% test set). Training lasted for 10 epochs with batch size 1000.

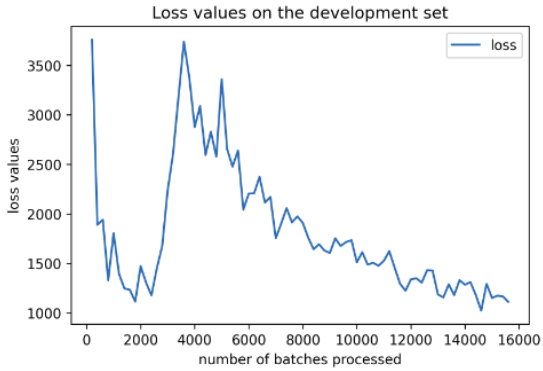

Figure 4.24: The loss values while training the ner component of the en_core_sci_md model(processing pipeline: tok2vec, tagger, parser, ner) on the CHEMDNER corpus (80% train set, 20% test set). Training took place for 10 epochs with batch size 1000.

The next experiment was performed using the en_core_sci_lg model. The "ner" component of the pipeline was trained on the CHEMDNER corpus. While training the test set was used for testing the model's performance. 80% of the CHEMDNER's training and development set was used for training and 20% for testing. The precision, recall and f1-score are measured on the test set (development set) while training for 7 epochs with batch size 1000 and are visualised in Figure 4.25. The loss value count on the test set while

Table 4.22: The overall precision, recall and f1 -score of the en_core_sci_md (processing pipeline: tok2vec, tagger, parser, ner) model trained on the CHEMDNER corpus (80% of CHEMDNER's training and evaluation set was used for training and 20% for evaluation).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	62.84	84.87	72.21
FAMILY	73.00	71.01	71.99
TRIVIAL	83.14	83.68	83.41
FORMULA	70.05	79.39	74.43
MULTIPLE	40.00	52.63	45.45
ABBREVIATION	84.57	79.19	81.79
IDENTIFIER	75.00	75.00	75.00

Table 4.23: The overall precision, recall and f1 -score of the en_core_sci_md model trained on the CHEMDNER training set.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
73.43	79.96	76.53

training is visualised in figure 4.26. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are given in the table 4.24. The results of the overall performance of the model are presented in table 4.25.

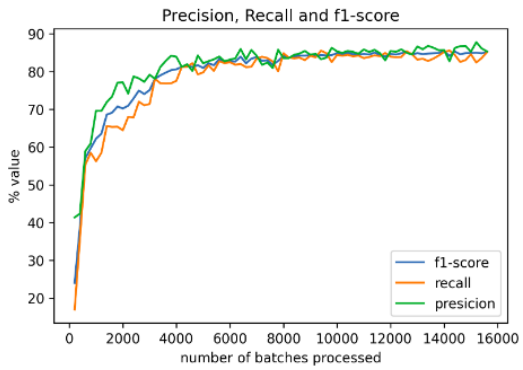

Figure 4.25: % values of p, r, f1-score while training the ner of the en_core_sci_lg model(processing pipeline: tok2vec, tagger, parser, ner) on the CHEMDNER corpus (80% train set, 20% test set). Training lasted for 7 epochs with batch size 1000.

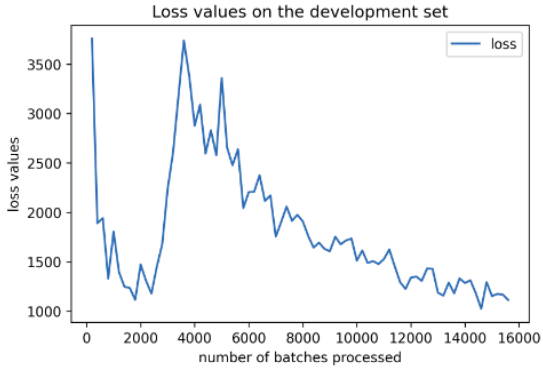

Figure 4.26: The loss values while training the ner component of the en_core_sci_lg model(processing pipeline: tok2vec, tagger, parser, ner) on the CHEMDNER corpus (80% train set, 20% test set). Training took place for 7 epochs with batch size 1000.

We also performed an experiment using the en_core_sci_scibert model. In this experiment, the processing pipeline of the en_core_sci_scibert model consists of the "transformer", the "tagger", the "parser" and the "ner". The "ner" component of the pipeline was trained on the CHEMDNER corpus. While training the test set was used for testing the model's performance. 80% of the CHEMDNER's training and development set was used for training and 20% for testing. The precision, recall and f1-score are measured on the test set (development set) while training for 5 epochs with batch size 128 and are visualised in Figure 4.27. The loss value count on the test set while training is visualised in figure 4.28. When the training was done, the performance of the model was evaluated using the evaluation set of the CHEMDNER corpus. The results of the performance of the model on identifying each entity and correctly assigning it to each SACHEM class, are

Table 4.24: The overall precision, recall and f1 -score of the en_core_sci_lg (processing pipeline: tok2vec, tagger, parser, ner) model trained on the CHEMDNER corpus (80% of CHEMDNER's training and evaluation set was used for training and 20% for evaluation).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	50.22	84.69	63.05
FAMILY	73.45	73.74	73.59
TRIVIAL	85.44	84.45	84.94
FORMULA	74.16	80.00	76.97
MULTIPLE	60.00	47.37	52.94
ABBREVIATION	80.59	79.19	79.88
IDENTIFIER	77.78	70.00	73.68

Table 4.25: The overall precision, recall and f1 -score of the en_core_sci_lg model trained on the CHEMDNER training set.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
69.26	80.71	74.55

given in the table 4.26. The results of the overall performance of the model are presented in table 4.27.

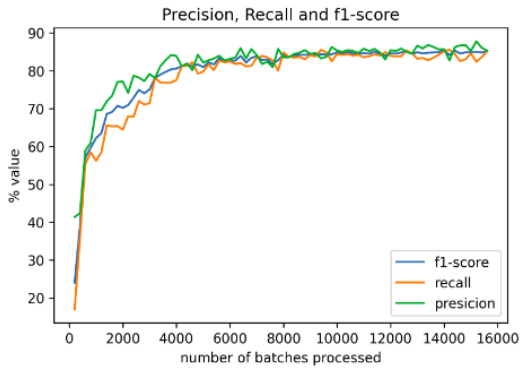

Figure 4.27: % values of p, r, f1-score while training the ner of the en_core_sci_lg model(processing pipeline: tok2vec, tagger, parser, ner) on the CHEMDNER corpus (80% train set, 20% test set). Training lasted for 5 epochs with batch size 128.

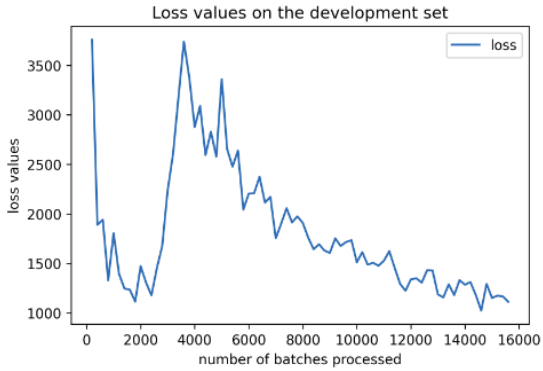

Figure 4.28: The loss values while training the ner component of the en_core_sci_lg model(processing pipeline: tok2vec, tagger, parser, ner) on the CHEMDNER corpus (80% train set, 20% test set). Training took place for 5 epochs with batch size 128.

Table 4.26: The overall precision, recall and f1 -score of the en_core_sci_scibert (processing pipeline: tok2vec, tagger, parser, ner) model trained on the CHEMDNER corpus (80% of CHEMDNER's training and evaluation set was used for training and 20% for evaluation).

Chemical Entities categories	PRECISION(%)	RECALL (%)	F1-SCORE (%)
SYSTEMATIC	65.21	82.66	72.90
FAMILY	79.59	68.29	73.51
TRIVIAL	77.79	81.49	79.60
FORMULA	68.09	77.58	72.52
MULTIPLE	66.67	52.63	58.82
ABBREVIATION	84.57	79.19	81.79
IDENTIFIER	94.12	80.00	86.49

When trying to train the en_core_sci_md/lg/scibert models that include in their processing pipeline a "tagger" and a "parser", there is not enough memory. Probably a machine with

Table 4.27: The overall precision, recall and f1 -score of the en_core_sci_scibert model trained on the CHEMDNER training set.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
74.15	77.87	75.96

more than 8 G RAM is needed.

Overview of results. We observed that adding a ”tagger” and a ”parser” at the processing pipeline improved the performance of the spaCy models. The comparison of the f1-scores of the trained models that did not have a ”tagger” and a ”parser” to those that had a ”tagger” and a ”parser” in the processing pipeline is presented in Table 4.28. The model with the highest f1-score is the en_core_sci_md model with processing pipeline that consists of a ”tokenizer”, a ”tagger”, a ”parser” and a ”ner”. The entity recogniser component was trained on the CHEMDNER corpus using 80% of the training and dvelopment datasets for training and the remaining 20% for testing. It is a statistical model with weights that enable it to make predictions of chemical entity labels: ABBREVIATION, FAMILY, FORMULA, IDENTIFIER, MULTIPLE, SYSTEMATIC, TRIVIAL. Figure 4.29 presents the processing pipeline of the trained spaCy model with the highest f1-score evaluated on the CHEMDNER test set.

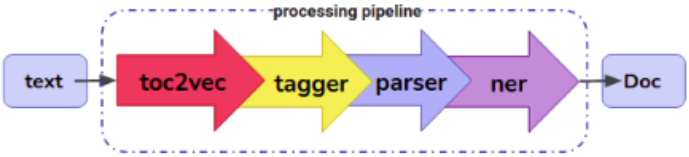

Figure 4.29: Processing pipeline that consists of the following components: toc2vec, tagger, parser ner.

Table 4.28: Presentation of the training experiments.

scispaCy model	Percentage for training and testing	f1-score on evaluation set	f1-score on evaluation set (tagger,parser)
en_core_sci_md	50% : 50%	42.68 %	-
en_core_sci_md	80% : 20%	72.12%	76.53 %
en_core_sci_md	90% : 10%	67.68 %	not enough memory
en_core_sci_lg	80% : 20%	56.66 %	74.55 %
en_core_sci_lg	90% : 10%	66. 44 %	not enough memory
en_core_sci_scibert	50% : 50%	58.91 %	-
en_core_sci_scibert	80% : 20%	67.37 %	75.96 %
en_core_sci_scibert	90% : 10%	not enough memory	not enough memory

4.1.6 Chemical Named Entity Recogniser - evaluation of the final model

According to the experiments that have been described and presented in this chapter the spaCy model that gave the best results:

- was developed using the en_core_sci_md processing pipeline: a full spaCy pipeline for biomedical data with a large vocabulary and 50k word vectors, and

- its processing pipeline included the tok2vec, the tagger, the parser and the ner components (figure 4.29),
- its NER component of the pipeline was trained on the training and development sets of the CHEMDNER corpus that were randomly split into 80% for training and 20% for testing.

The precision, recall and f1-score evaluating the performance of the spaCy model are calculated on the evaluation set of the CHEMDNER corpus. The evaluation set of the CHEMDNER corpus consists of 3000 abstracts (see section 3.3) of various disciplines of Chemistry and was not used during training.

For the calculation of these scores, not only the correct identification of each chemical entity is taken into account, but also if each chemical entity is correctly classified to one of the seven classes: ABBREVIATION,FAMILY, FORMULA, IDENTIFIER, MULTIPLE, SYSTEMATIC and TRIVIAL. In the BioCreative IV community challenge [35], for the chemical entity mention recognition - CEM task, the participating teams were evaluated according to the performance of their chemical NER tool in correctly identifying the chemical entities without taking into account the correct classification of each entity to one of the seven classes.

We wanted to compare the performance of the the spaCy model with the best f1-score to the tools that partipated in the CEM task of BioCreative IV community challenge. In order to do so, we evaluated the spaCy model with the best performance according to whether it identifies correctly the chemical entities in the evaluation set of the CHEMDNER corpus.

We then compared the precision, recall and f-score of the model to the results of the participating teams that are presented in figure 3.4. In total 26 teams participated in the CEM task with 106 submissions [35]. The mean of the precision, recall and f1-scores of the participating submissions are: p = 80.27%, r = 70.94%, f1-score = 73.49% and the medians are: p = 85.35%, r = 71.87%, f1-score = 76.62% (one of the submissions had 0 values and was not included in the calculations of the mean and the median).

The spaCy model's precision, recall and f1-score are presented in table 4.29. Our model, in comparison to the chemical NER tools that participated in the CEM task of the BioCreative competition takes the 10th position and its overall performance is relatively close to the scores of the winning team (precision: 89.09%, recall: 85.75%, fi-score: 87.39%).

Table 4.29: The overall precision, recall and f1 -score of the spaCy model on correctly identifying chemical entities in CHEMDNER's evaluation set.

NER PRECISION (%)	NER RECALL (%)	NER F1-SCORE (%)
84.71	77.07	80.71

4.2 The Case of Hyperthermophile Microorganisms

4.2.1 Abstract collection

The aim of this case study is to research all the available literature of selected species of hyperthermophilic microorganisms (in the PubMed database) about the chemical entities (metabolites, substrates, enzyme cofactors, etc) that are mentioned in the abstracts of the selected publications. Then, with the help of generated association rules (see Subsection

2.5 for background), the goal is to study the relationships between data items, such as the relationships between the microorganisms and the chemical entities and the relationships among the chemical entities themselves. The relations between data items that will be identified will be verified by reading the relevant literature.

This work aims to facilitate the study of the biological processes of hyperthermophilic microorganisms by emphasising on the chemical entities that co-occur in the literature about hyperthermophiles. The association rules that will be generated will serve as a starting point in checking out the main focus of the research on the metabolic processes of the organisms of interest (in this case nine selected species of hyperthermophilic microorganisms).

The selected hyperthermophilic microorganisms for this task are species that have been isolated in cultures in the laboratory for many years and have been extensively studied for large scale applications in industry [76]: *Thermococcus kodakarensis, Pyrococcus furiosus, Metallosphaera sedula, Thermotoga maritima, Caldicellulosiruptor bescii, Sulfolobus solfataricus, Thermus thermophilus, Thermoanaerobacter mathranii* and *Caldicellulosiruptor hydrothermalis*.

The ORGANISMS web resource [56] (`http://organisms.jensenlab.org`) was used in order to retrieve abstracts with mentions of selected hyperthermophilic microorganisms. The search is done according to each species TaxID, retrieved from NCBI taxonomy (`www.ncbi.nlm.nih.gov/taxonomy/`). The tool ORGANISMS and SPECIES [56] takes a dictionary-based flexible matching approach and tags species and other taxa names (Linnaean binomial names and other synonyms) in text.

We collected 14415 unique abstracts in Medline database. In these abstract we found mentions of:

- *Thermococcus kodakarensis* (899 abstracts),
- *Pyrococcus furiosus* (2740 abstracts),
- *Metallosphaera sedula*(181 abstracts),
- *Thermotoga maritima*(3075 abstracts),
- *Caldicellulosiruptor bescii*(224 abstracts),
- *Sulfolobus solfataricus*(3249 abstracts),
- *Thermus thermophilus*(5757 abstracts),
- *Thermoanaerobacter mathranii* (150 abstracts),
- *Caldicellulosiruptor hydrothermalis* (47 abstracts).

The number of abstracts retrieved for each microorganism is presented in figure 4.30 and in table 4.30. The abstracts were extracted in June 2021. *Thermus thermophilus, Sulfolobus solfataricus, Thermotoga maritima* and *Pyrococcus furiosus* seem to be the most studied organisms in comparison to the other species, since millions of abstracts that mention them have been deposited in PubMed.

Table 4.30: The number of the abstracts retrieved with ORGANISMS web source about each hyperthermophile microorganism and the number of abstracts that the spaCy NER model identified chemical entities in.

Microorganism	TaxID	# of abstracts	# of abstracts with chemicals
Thermococcus kodakarensis	311400	899	737
Pyrococcus furiosus	2261	2740	2185
Metallosphaera sedula	43687	181	151
Thermotoga maritima	2336	3075	2553
Caldicellulosiruptor bescii	31899	224	195
Sulfolobus solfataricus	2287	3249	2482
Thermus thermophilus	274	5757	4718
Thermoanaerobacter mathranii	583357	150	115
Caldicellulosiruptor hydrothermalis	413888	47	33

4.2.2 Chemical NER on Abstract Collection for Hyperthermophile Microorganisms

The spaCy model was used for identifying the chemical entities in the selected abstracts. In these 14415 abstracts that are being studied, 14 were included in the CHEMDNER corpus (PubMed IDs: 23336064, 23429192, 23293964, 23295222- about *T.maritima*, 23344974, 23303790, 23589624, 23408858, 23182463 - abstracts about *P. furiosus*, 23622866, 23361460, 23411285 - abstracts about *S. solfataricus*, 23118486, 23411229 - abstracts about *T. thermophilus*) and were used during the training of the model. The spaCy model did not identify any chemical entity in 2997 abstracts and 11418 abstracts were found to have mentions of chemical entities according to the spaCy model. The number of abstracts extracted from PubMed using the ORGANISMS resource about each microorganism and the number among them that has mentions of chemical entities identified by the spaCy model are described in more detail in figure 4.30 and in table 4.30.

When the chemical NER step was finished, a dictionary was provided mentioning for each PubMed ID (key of the dictionary) all the chemical entities that were identified by the spaCy model in the abstract and the species names that were tagged in the abstract by the SPECIES tagger (values of the dictionary). It was common among the identified chemicals for one chemical to be present in various forms, for example, in singular and plural form, with a capital first letter, etc. (for example the chemical entities: phenol, Phenol). This is why the identified chemicals were further processed in the way that the chemical entities (words or phrases) were reduced to their stem so that common suffixes were removed. In total 10562 unique chemical entities were retrieved from the selected abstracts.

Bar plot of the number of abstracts mentioning each microorganism & the number of them that have references of chemicals.

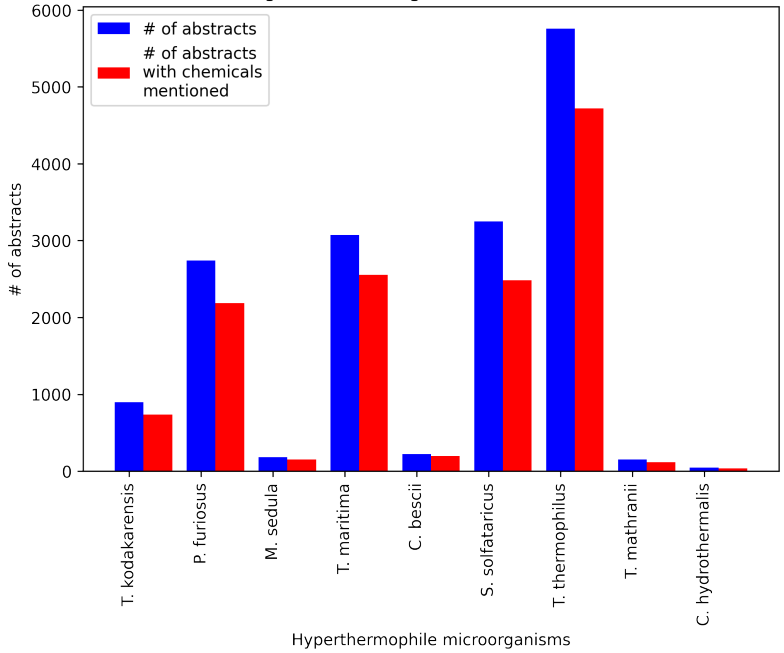

Figure 4.30: The bar plot presents the number of abstracts where each of the hyperthermophile microorganisms is tagged (blue bars) and the number of these abstracts that also have mentions of chemicals identified by the spacy model (red bars).

5. ASSOCIATION RULES FOR HYPERTHERMOPHILES AND CHEMICALS

5.1 Frequent itemset generation & Association rules extraction

In the process of association rule mining between hyperthermophile microorganisms and chemical entities or between chemical entities and chemical entities the frequent itemsets are detected using the fpgrowth algorithm (see section 2.5.2 for more detail in frequent itemset mining methods) with a low minimum support threshold: min_support = 0.005. The number of frequent itemsets that were identified is 320. In Appendix B the Table B presents the detected frequent items and itemsets and their support. Once the frequent itemsets were found, association rules were generated by satisfying both the minimum support and minimum confidence threshold. The threshold in both cases was 0.005 (see section 2.5.2 for more detail about generating association rules from frequent itemsets). In total 432 association rules were generated.

In Figures 5.1 and 5.2 the association rules between frequent items and/or itemsets are visualized according to their support (x axis) and confidence (y axis) (see section 2.5.1 for the definitions of support and confidence). In Figure 5.1 is also presented the length of the itemsets between which association rules are generated. We observe that the association rules are generated mostly between two frequent items (in purple) and between a frequent item and a frequent itemset that consists of two items (in yellow). In Figure 5.2 is also visualised the lift value (see section 2.5.3) of each association rule.

The selected association rules for further study and visualization were the association rules whose lift value is more than 5 (lift > 5) (see section 2.5.3 for more detail in evaluating the importance of the generated association rules). 36 association rules have lift values more than 5. These 36 association rules are presented in Appendix A in the Table A. These 36 association rules between items and/or itemsets of hyperthermophilic microorganisms and chemical entities and among chemical entities are visualised in figure 5.3. We further investigated the available literature on the nine selected species of hyperthermophilic microorganisms about the co-occurrence of the itemsets between whom the association rules were generated.

5.2 Investigation of the generated association rules

The association rules generated by the association rule analysis (Figure 5.3) require further research. The organism specific collection of abstracts was specifically studied for the co-occurence of these items or itemsets and about their biological significance. In this section first the association rules with biological interest are going to be presented and discussed and then the association rules that were expected in terms of their biological meaning.

Biologically interesting association rules:

- Association Rules 35: (aminoacyl) ==> (*T. thermophilus*) & 36: (*T. thermophilus*, amino acid) ==> (aminoacyl)

  Aminoacylation is the attachment of an amino acid to a tRNA. It is a two-step process catalyzed by aminoacyl-tRNA synthetases (aaRSs).

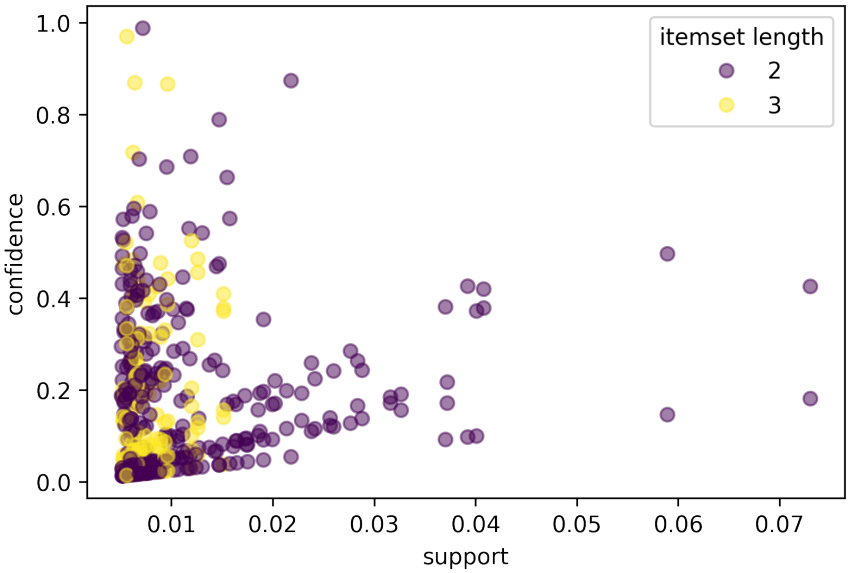

Figure 5.1: Visualization of the generated association rules: their support (x axis) and confidence (y axis), the length of the frequent itemsets.

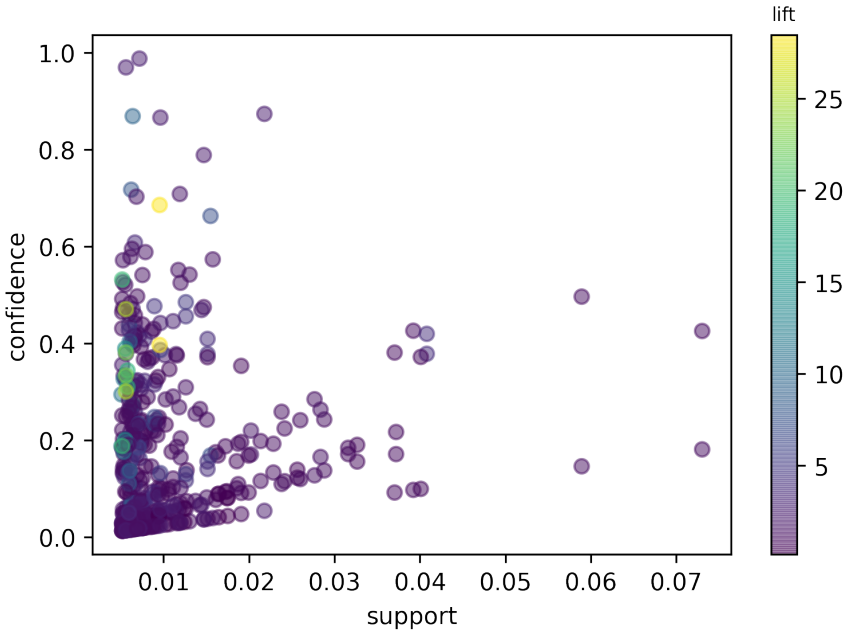

Figure 5.2: Visualization of the generated association rules: their support (x axis) and confidence (y axis), the lift value.

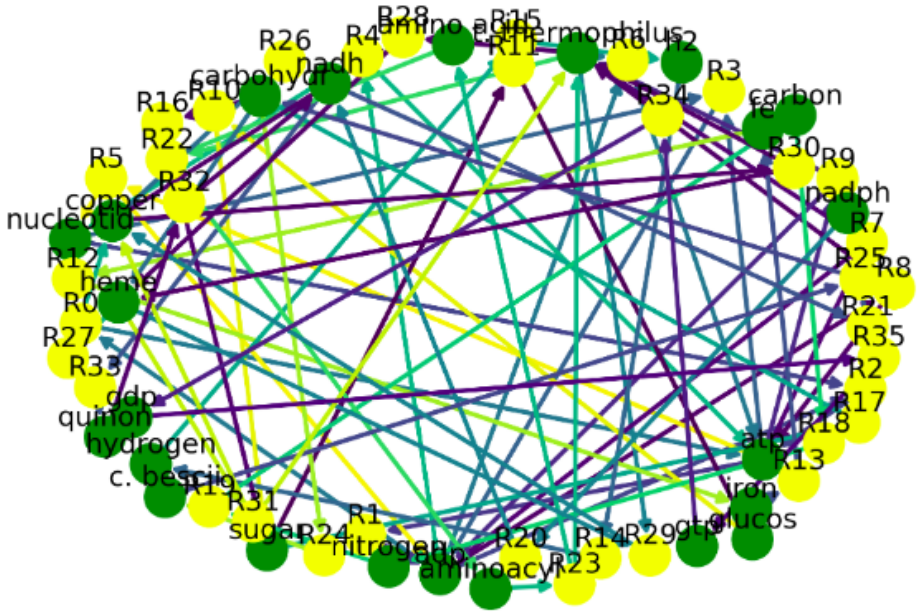

Figure 5.3: Visualization of the 36 association rules (lift > 5).

- The first step, termed “activation”, is the formation of an aminoacyl-AMP (aminoacyl-adenylate) on the enzyme through the hydrolysis of adenosine triphosphate (ATP).
- The second step is the transfer of the activated amino acid residue from the adenylate to a tRNA in a reaction referred to as “charging” [45].

In general, the genetic system of *T. thermophilus* has been used to overexpress active tagged versions of its own proteins and a vast amount of studies have used *T. thermophilus* as a source of crystallizable proteins, in order to study their functions [76]. In the downloaded abstracts about *T. thermophilus*, the aminoacyl-tRNA synthetases (aaRSs) of this hyperthermophile microorganism seem to have been extensively studied mostly in crystallization studies and mechanistic studies (examples of PubMed IDs on this subject: 31869198, 31084346, 26184179, 24095058, 23536245, 19496540, 9115984). Also the aminoacyl-tRNA synthetases (aaRSs) have been studied as targets for new therapies with antibiotics that can block the translation of bacteria (examples of PubMed IDs on this subject: 32817463, 32631562, 32088946, 31600972).

- Association Rules 3 - 6 & 9, 10: copper, heme, *T. thermophilus*

  Heme–copper oxygen reductases (HCO) [64] are transmembrane enzymes: the last enzymatic complexes of most aerobic respiratory chains. Their catalytic role in the respiratory chain is to reduce O2 to water in a process coupled to proton translocation across the membrane. These enzymes couple the catalytic reaction to charge separation and charge translocation across the prokaryotic cytoplasmic or mitochondrial membrane and help in energy conservation. In this way they contribute to synthesis of ATP, solute/nutrient cell import and motility. In subunit I, there is an heme and the catalytic site. The catalytic site is formed by an heme and a copper ion which is bound to a histidine residue covalently linked to a tyrosine residue. Subunit I is common to all enzymes. A second subunit (subunit II) might be present, which may have

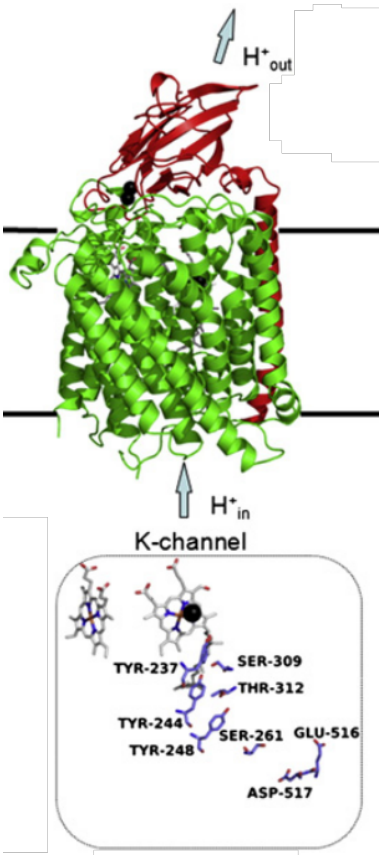

Figure 5.4: Crystallographic structure of heme–copper oxygen reductase of *T. thermophilus*. The catalytic subunit is shown in green and an additional subunit is presented in red. Copper ions are represented as black spheres and the heme is shown as sticks (orange)[64].

a binuclear copper center. In Figure 5.4 is presented the crystallographic structure of heme–copper oxygen reductase of *T. thermophilus* [64]. A number of studies have focused on the mechanistic function and structure of HCOs of *T. thermophilus* (examples of PubMed IDs on this subject: 34022199, 33962016, 33535124, 30523412, 2375508, 22139175, 15041681). It has been shown that the HCOs of Thermus thermophilus are able to catalyze the reduction of nitric oxide (NO) to nitrous oxide (N2O) under reducing anaerobic conditions, supporting the hypothesis of the presence of a denitrification pathway and aerobic respiration [15].

- Association Rules 25: ATP ==> (ADP, T. thermophilus) & 26 (ADP, T. thermophilus) ==> ATP

  In the abstracts about *T. thermophilus* the role of ATP has been studied for its role in aminoacylation, in tRNA degradation, in the production of CoenzymeA, about transporters (mostly the ATP Binding Cassette -ABC family of transporters) and enzymes (f.ex. ATPases, kinases).

- Association Rules 17: (carbohydrate) ==> (C. bescii) & 18: (C. bescii) ==> (carbohydrate)

  Carbohydrate metabolism has been extensively studied in *Caldicellulosiruptor bescii* for the production of ethanol from lignocellulosic biomass (biomass hydrolysis, carbohydrate transport and utilization, and production of ethanol) [75, 58, 31, 65, 40]. A wide number of studies focus on engineering this bacterium for optimal conversion of lignocellulose to commercial products.

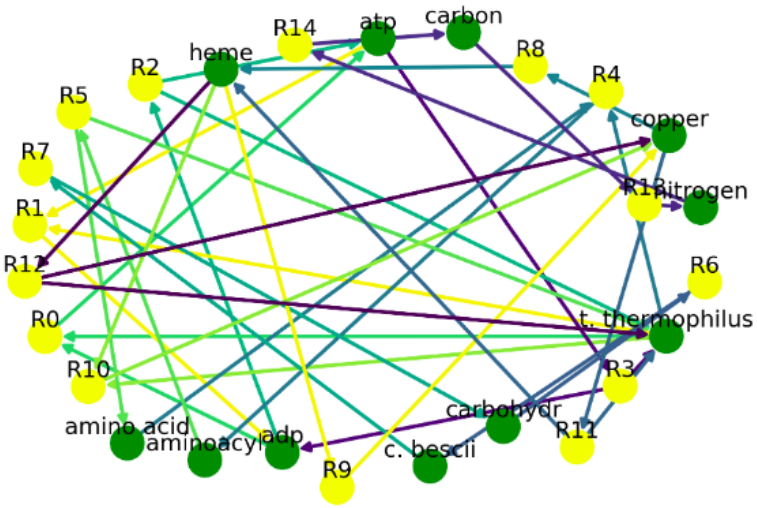

Figure 5.5: Visualization of the 15 association rules that indicate an interesting relationship about hyperthermophiles and chemical compounds related to a biological process.

- Association Rules 21: (carbon) ==> (nitrogen) & 22: (nitrogen) ==> (carbon)

  Carbon metabolism and Nitrogen metabolism studied together in thermophilic microbes [10].

Other association rules with more general and expected occurrences also came up:

- Association Rules 1 & 2 between itemsets: GTP, GDP
- Association Rules 27 & 28 between itemsets: ADP, ATP It has been observed in P. furiosus that under heat shock conditions the pools of ADP and ATP increased significantly. A similar increase has been observed for E. coli in response to a temperature upshift. This probably reflects an increased demand for energy during adaptation to stressful conditions78.
- Association Rules 11 & 12 between itemsets: NADH, NADPH
- Association Rules 7 & 8 between itemsets: quinone, NADH
- Association Rules 13, 14, 19, 20: ADP, ATP, amino acids
- Association Rules 31-34: glucose, sugar, carbonhydrates
- Association Rules 23 & 24 between itemsets: hydrogen, H2 There are various applications for extremophiles in the production of hydrogen through anaerobic fermentation and hydrogenases [9] . The problem indicated here is that this system can not discriminate between a chemical molecule mentioned by its name or its molecular form.
- Association Rules 15 & 16 between itemsets: iron, Fe Similarly about iron (Fe). Iron oxidation is extensively studied in thermophilic archaea.

6. CONCLUSIONS AND FURTHER WORK

This project focuses on creating a chemical Named Entity Recogniser (NER). The chemical NER tool is created using the open source python library for NLP: spaCy and the python package that contains spaCy models for processing scientific biomedical text: scispaCy. The training and development datasets that were used for the training of the chemical NER tool came from the CHEMDNER corpus: the largest, publicly available, manually annotated corpus of chemical entities that includes 10,000 abstracts from recent publications on diverse disciplines of chemistry. The chemical entity mentions of the CHEMDNER corpus were manually labeled and were manually classified to one of the seven predefined structure-associated chemical entity mention (SACEM) classes: Abbreviation, Family, Formula, Identifier, Multiple, Systematic and Trivial.

A full spaCy pipeline for biomedical data with a large vocabulary and 50k word vectors was used. Various training experiments showed that the pipeline architecture, that gives the best score (f1-score) on the CHEMDNER corpus' evaluation dataset, consists of the following components: the "tokenizer" that generates the contextual embeddings for the input tokens, the "part-of-speech-tagger" (assigns part-of-speech tags) and the "dependency parser" (assigns grammatical dependency labels) that process the grammatical relations between the tokens and the the "named entity recogniser" that detects and labels the named entities. Adding a "tagger" and a "parser" at the processing pipeline improved the performance of the spaCy models, in comparison to the pipeline that consisted of the "tok2vec" and the "ner" components (that doesn't take into account the grammatical dependencies between tokens). The results are presented in Table 4.28

The named entity recogniser component of the pipeline was trained on the CHEMDNER corpus. The CHEMDNER corpus' training (3,500 abstracts) and development datasets (3,500 abstracts) consisting of in total 7,000 abstracts, were randomly splited in 80% for training and 20% for development. The model was evaluated on the CHEMDNER corpus' evaluation set. Training including the true negative subset of the CHEMDNER corpus showed better results than without including it (Table 4.18). A dictionary of chemicals was not used since experiments showed that it adds false positives to the identified chemical entities (Table 4.16).

The chemical NER tool' performance was compared to the performance of other available tools. In the BioCreative IV community challenge, the chemical entity mention recognition - CEM task evaluated the ability of chemical NER tools to specifically locate within a document every chemical entity mention, by exactly locating their start and end character indices. The participating systems were evaluated on the evaluation set of the CHEMDNER corpus that had 25,351 chemical entity mentions (7,563 unique chemical names). The evaluation metrics for comparing the performance of the systems that participated in the challenge were: the precision, the recall and the f1-score. The participating teams were evaluated according to the performance of their chemical NER in correctly identifying the chemical entities without taking into account the correct classification of each entity to one of the seven classes that the annotators of the CHEMDNER corpus have manually classified each chemical entity, but only by taking into account the correct identification of the chemical entities in the test set.

The evaluation of the participating chemical NER tools was done on the more relaxed criterion of only correctly identifying a chemical entity, without correctly classifying it to one of the aforementioned SACEM classes. The best teams in the BioCreative IV CEM task scored f1-scores 72% - 88% on the CHEMDNER corpus' test set. The winning team's

scores were: precision: 89.09%, recall: 85.75%, fi-score: 87.39%. The chemical NER tool that was developed using spaCy scored precision: 84.71%, recall: 77.07%, fi-score: 80.71% (Table 4.29).

After the chemical NER tool was developed it was used in a case study about hyperthermophile microorganisms. The available literature of the following microorganisms: *Thermococcus kodakarensis, Pyrococcus furiosus, Metallosphaera sedula, Thermotoga maritima, Caldicellulosiruptor bescii, Sulfolobus solfataricus, Thermus thermophilus, Thermoanaerobacter mathranii* and *Caldicellulosiruptor hydrothermalis* was retrieved with the ORGANISMS web source (`http://organisms.jensenlab.org`). The chemical NER tool was used to identify mentions of chemical entities in these organism specific abstracts. The chemical entities found to be mentioned in the hyperthermophilic microorganisms abstracts were further analysed for the appearance of patterns that appear frequently in a dataset and their co-occurrences. Association rule mining is a two-step process:

1. Detecting all the frequent itemsets that occur at least as frequently as a predetermined minimum support count (min_support=0.005) using the fpgrowth algorithm, and

2. Generating strong association rules from the frequent itemsets, that satisfy a predetermined minimum support and minimum confidence (in both cases 0.005). The selected association rules for further study and visualization were the association rules whose lift value is more than 5 (lift > 5).

This project aims to facilitate the study of the biological processes of microorganisms of interest by first identifying the mentions of chemical entities (metabolites, substrates, enzyme cofactors, etc.) in abstracts of publications about them and then by retrieving co-occurrence associations between microorganisms and chemical entities and between chemical entities. The relations pulled out from the selected papers can give an overview indicating the most studied subjects referring to chemicals, about the selected dataset of organisms. More specifically in the case study of hyperthermophiles examples of interesting associations in the examined literature are: the association of carbohydrates to *C. bescii*,the association of *T .thermophilus* to copper and to heme and the association of *T .thermophilus* to aminoacyl. Carbohydrate metabolism has been extensively studied in *C. bescii* for the production of ethanol from lignocellulosic biomass. Extensive studies of T. thermophilus show that the heme–copper oxygen reductases are able to catalyze the reduction of nitric oxide to nitrous oxide under reducing anaerobic conditions. The aminoacyl-tRNA synthetases (aaRSs) of *T. thermophilus* seem to have been extensively studied mostly in crystallization studies and mechanistic studies. Also the aminoacyl-tRNA synthetases (aaRSs) have been studied as targets for new therapies with antibiotics that can block the translation of bacteria.

The contributions of this project are the following:

- We have created a chemical NER tool with performance close to state-of-the-art that can be easily used for identifying chemicals in scientific text in various applications.

- We have used the spaCy library, a user friendly open source open-source software library that offers prebuilt statistical neural network models to create convolutional neural network models for part-of-speech tagging, dependency parsing, text categorization and named entity recognition (NER).

- We have presented an application of association rule mining to microbiology that can help in retrieving co-occurrence associations between microorganisms and chemical entities is a selection of organism specific abstracts.
- We have collected and presented in detail the available literature in chemical NER: the approaches followed over the last decade and a variety of the tools that have been developed. As well as a detailed list of all the available corpora with mentions of chemical entities.

Beyond the hyperthermophilic microorganisms study, the presented method could be applied to any microorganism specific abstract collection. The chemical NER tool developed using spaCy facilitates the identification of chemical entities in scientific text. Furthermore, the association rule mining approach introduced in this project provides co-occurrence associations between microorganisms and chemical entities and between chemical entities in a selected abstract collection.

Process, environment, organism (PREGO) is a web resource (`https://imbbc.hcmr.gr/project/prego/`) that combines large-scale text-mining, data-mining, and network analysis in order to elucidate the biogeochemical processes carried out by organisms in various environments. The described framework could contribute in offering information about organisms of interest and the chemical entities that are associated with them. The chemicals could be linked to biological processes and give information about the organism's metabolic pathways, homeostasis regulation, interactions between other organisms and interactions with its environment.

The current state-of-the-art in NLP is using unsupervised pretrained transformer - based pipelines such as Bidirectional Encoder Representations from Transformers (BERT). The main use case for pretrained transformer models is transfer learning. A pretrained large generic model (trained on a huge amount of plain text corpus) is loaded and is then trained on a smaller labeled dataset (for example in our case on the CHEMDNER corpus which has labeled chemical entities). The Allen AI Institute has released SCIBERT, a pretrained language model for scientific text based on BERT [6]. SCIBERT is trained on the full text of 1.14M biomedical and computer science papers from the Semantic Scholar corpus (`https://github.com/allenai/scibert`). scispaCy includes a full spaCy pipeline for biomedical data with a 785k vocabulary and allenai/scibert-base as the transformer model (the en_core_sci_scibert model).

During the training experiments we tried to fine-tune the en_core_sci_scibert model for the chemical NER task (see results in Table 4.28). The drawback was that the model needs much longer than the other spaCy models to be trained and is harder to deploy on a machine with limited resources. The authors of spaCy recommend to use a GPU for both training and usage (`https://explosion.ai/blog/spacy-transformers`), since transformer architectures are not designed to operate efficiently on CPU. A next step, in order to improve the performance of the chemical NER tool, would be to train scispaCy's en_core_sci_scibert model on the CHEMDNER corpus using a GPU.

ABBREVIATIONS - ACRONYMS

NER	Named ENtity Recognition
NLP	Natural Language Processing
HMM	Hidden Markov Models
MEMM	Maximum Entropy Markov Models
CRF	Conditional Random Fields
SVM	Support Vector Machines
ER	Entity Recognition
IE	Information Extraction
PMC	PubMed Central
NN	Neural Networks
DNN	Deep Neural Networks
CNN	Convolutional neural network
LSTM	Long short-term memory Networks
BiLSTM	Bidirectional Long short-term memory Networks
RNN	Recurrent neural network
POS	Part-of-Speach
NE	Named Entity
IUPAC	International Union of Pure and Applied Chemistry

APPENDIX A. FIRST APPENDIX

#	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	length
1	gtp	gdp	0.0241	0.0139	0.0095	0.3964	28.4634	0.0092	1.6336	2
2	gdp	gtp	0.0139	0.0241	0.0095	0.6855	28.4634	0.0092	3.1034	2
3	copper, t. thermophilus	heme	0.0119	0.0187	0.0056	0.4706	25.2262	0.0054	1.8537	3
4	heme	copper, t. thermophilus	0.0187	0.0119	0.0056	0.3005	25.2262	0.0054	1.4125	3
5	heme, t. thermophilus	copper	0.0147	0.0168	0.0056	0.3810	22.6548	0.0054	1.5882	3
6	copper	heme, t. thermophilus	0.0168	0.0147	0.0056	0.3333	22.6548	0.0054	1.4779	3
7	quinone	nadh	0.0097	0.0275	0.0052	0.5315	19.3281	0.0049	2.0759	2
8	nadh	quinon	0.0275	0.0097	0.0052	0.1879	19.3281	0.0049	1.2194	2
9	copper	heme	0.0168	0.0187	0.0058	0.3438	18.4269	0.0055	1.4954	2
10	heme	copper	0.0187	0.0168	0.0058	0.3099	18.4269	0.0055	1.4246	2
11	nadph	nadh	0.0142	0.0275	0.0055	0.3889	14.1412	0.0051	1.5914	2
12	nadh	nadph)	0.0275	0.0142	0.0055	0.2006	14.1412	0.0051	1.2332	2

#	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	length
13	nucleotide', atp	adp	0.0202	0.0234	0.0064	0.3160	13.5142	0.0059	1.4278	3
14	adp	nucleotide, atp	0.0234	0.0202	0.0064	0.2734	13.5142	0.0059	1.3484	3
15	fe	iron	0.0173	0.0271	0.0051	0.2944	10.8791	0.0046	1.3789	2
16	iron	fe	0.0271	0.0173	0.0051	0.1877	10.8791	0.0046	1.2098	2
17	carbohydrate	c. bescii	0.0304	0.0164	0.0053	0.1758	10.7337	0.0048	1.1934	2
18	c. bescii	carbohydr	0.0164	0.0304	0.0053	0.3262	10.7337	0.0048	1.4390	2
19	atp	adp, nucleotide	0.0921	0.0074	0.0064	0.0694	9.4323	0.0057	1.0667	3
20	adp, nucleotide	atp	0.0074	0.0921	0.0064	0.8690	9.4323	0.0057	6.9328	3
21	carbon	nitrogen	0.0441	0.0150	0.0060	0.1369	9.1414	0.0054	1.1413	2
22	nitrogen	arbon	0.0150	0.0441	0.0060	0.4035	9.1414	0.0054	1.6025	2
23	h2	hydrogen	0.0136	0.0540	0.0060	0.4387	8.1186	0.0052	1.6853	2
24	hydrogen	h2	0.0540	0.0136	0.0060	0.1102	8.1186	0.0052	1.1086	2
25	atp	adp, t. thermophilus	0.0921	0.0087	0.0062	0.0675	7.7839	0.0054	1.0631	3
26	adp, t. thermophilus	atp	0.0087	0.0921	0.0062	0.7172	7.7839	0.0054	3.2100	3
27	adp	atp	0.0234	0.0921	0.0155	0.6629	7.1951	0.0133	2.6933	2
28	atp	adp	0.0921	0.0234	0.0155	0.1683	7.1951	0.0133	1.1742	2
29	t. thermophilus, atp	adp	0.0392	0.0234	0.0062	0.1585	6.7773	0.0053	1.1605	3
30	adp	t. thermophilus, atp	0.0234	0.0392	0.0062	0.2659	6.7773	0.0053	1.3088	3
31	glucose	sugar	0.0361	0.0384	0.0089	0.2476	6.4539	0.0075	1.2780	2
32	sugar	glucose	0.0384	0.0361	0.0089	0.2329	6.4539	0.0075	1.2565	2
33	carbohydrate	sugar	0.0304	0.0384	0.0071	0.2334	6.0852	0.0059	1.2545	2
34	sugar	carbohydrate	0.0384	0.0304	0.0071	0.1849	6.0852	0.0059	1.1896	2
35	aminoacyl	t. thermophilus, amino acid	0.0250	0.0730	0.0096	0.3860	5.2841	0.0078	1.5096	3

APPENDIX B. SECOND APPENDIX

support	itemsets
0.4039	t. thermophilus
0.2176	t. maritima
0.2098	s. solfataricus
0.1848	p. furiosus
0.1717	amino acid
0.1188	nucleotid
0.1079	n
0.0973	c
0.0921	atp
0.0730	amino acid, t. thermophilus
0.0622	t. kodakarensi
0.0589	nucleotid, t. thermophilus
0.0540	hydrogen
0.0441	carbon
0.0408	n, c
0.0401	n, t. thermophilus
0.0392	atp, t. thermophilus
0.0384	sugar
0.0372	t. maritima, amino acid
0.0370	c, t. thermophilus
0.0361	glucos
0.0327	amino acid, s. solfataricus
0.0316	amino acid, p. furiosus
0.0310	oxygen
0.0308	histidin
0.0308	cystein
0.0304	carbohydr
0.0288	s. solfataricus, nucleotid
0.0284	amino acid, n
0.0282	phosphat
0.0277	t. maritima, c
0.0275	nadh
0.0271	iron
0.0260	t. maritima', 'n
0.0257	s. solfataricus, p. furiosus
0.0250	aminoacyl
0.0242	s. solfataricus, n
0.0241	gtp
0.0238	atp, t. maritima
0.0234	adp
0.0229	amino acid, nucleotid
0.0226	glutam
0.0225	sulfur
0.0218	t. thermophilus, aminoacyl
0.0214	n, p. furiosus
0.0213	lysin

support	itemsets
0.0210	zinc
0.0206	arginin
0.0202	atp, nucleotid
0.0200	t. maritima, nucleotid
0.0191	s. solfataricus, c
0.0191	t. thermophilus, hydrogen
0.0190	serin
0.0187	amino acid, c
0.0187	heme
0.0186	nucleotid, p. furiosus
0.0175	t. maritima, t. thermophilus
0.0174	t. maritima, p. furiosus
0.0173	atp, p. furiosus
0.0173	fe
0.0172	aspart
0.0168	copper
0.0165	t. thermophilus, p. furiosus
0.0164	c. bescii
0.0164	alanin
0.0164	c, p. furiosus
0.0161	atp, s. solfataricus
0.0160	iron-sulfur
0.0158	t. thermophilus, nadh
0.0155	atp', 'adp
0.0152	n', 'c', 't. thermophilus
0.0151	t. kodakarensi, p. furiosus
0.0151	tryptophan
0.0150	sds
0.0150	nitrogen
0.0148	pyruv
0.0147	heme', 't. thermophilus
0.0147	t. thermophilus, oxygen
0.0147	alcohol
0.0146	ethanol
0.0145	amino
0.0145	histidin, t. maritima
0.0143	t. maritima, hydrogen
0.0142	nadph
0.0139	adenin
0.0139	gdp
0.0138	hydrogen, p. furiosus
0.0138	co2
0.0137	disulfid
0.0136	h2
0.0134	prolin
0.0130	gtp, t. thermophilus
0.0130	tyrosin
0.0129	mg(2+)

support	itemsets
0.0129	m. sedula
0.0127	atp', 'amino acid
0.0126	t. maritima, n, c
0.0124	s. solfataricus, t. thermophilus
0.0123	methionin
0.0122	purin
0.0120	glutamin
0.0120	amino acid, nucleotid, t. thermophilus
0.0119	copper, t. thermophilus
0.0118	maltos
0.0118	s. solfataricus, carbon
0.0117	t. thermophilus, lysin
0.0116	histidin, t. thermophilus
0.0116	t. kodakarensi', 'amino acid
0.0115	t. maritima, carbohydr
0.0111	amino acid, aminoacyl
0.0111	methyl
0.0111	atp, n
0.0111	t. maritima, sugar
0.0109	t. kodakarensi, s. solfataricus
0.0107	cystein, t. thermophilus
0.0106	nucleotid, n
0.0106	leucin
0.0106	flavin
0.0106	s
0.0105	glycin
0.0105	pyrimidin
0.0102	glucos, s. solfataricus
0.0102	t. thermophilus, carbon
0.0102	iron, p. furiosus
0.0100	adenosin
0.0100	nucleosid
0.0099	t. mathranii
0.0098	nucleotid, c
0.0097	quinon
0.0096	amino acid, t. thermophilus, aminoacyl
0.0095	carboxyl
0.0095	gtp, gdp
0.0095	co
0.0095	atp, c
0.0095	urea
0.0095	sugar, p. furiosus
0.0095	t. maritima, s. solfataricus
0.0095	polyacrylamid
0.0094	amp
0.0094	amino acid, n, t. thermophilus
0.0093	s. solfataricus, sugar
0.0093	t. maritima, carbon
0.0092	ribos
0.0092	fatty acid

support	itemsets
0.0091	mg2+
0.0091	t. thermophilus, phosphat
0.0090	xylos
0.0089	glucos, sugar
0.0089	coa
0.0089	amino acid, n, c
0.0089	o2
0.0088	t. thermophilus, arginin
0.0088	nad
0.0088	threonin
0.0087	magnesium
0.0087	t. thermophilus, adp
0.0087	uracil
0.0086	atp, nucleotid, t. thermophilus
0.0086	glycerol
0.0085	glucos, t. maritima
0.0084	4fe-4
0.0084	h
0.0083	carbon, p. furiosus
0.0083	glutam, t. thermophilus
0.0082	ser
0.0082	cys
0.0082	cellobios
0.0081	acetyl-coa
0.0081	sulfur, p. furiosus
0.0081	phenylalanin
0.0081	t. maritima', 'phosphat
0.0081	glucos, p. furiosus
0.0081	hydroxyl
0.0079	t. thermophilus, prolin
0.0079	s. solfataricus, hydrogen
0.0078	glucos, t. thermophilus
0.0078	superoxid
0.0078	aldehyd
0.0078	s. solfataricus, n, c
0.0078	guanin
0.0077	t. thermophilus, zinc
0.0077	fad
0.0076	ammonia
0.0075	t. kodakarensi, n
0.0075	iron, t. thermophilus
0.0075	t. thermophilus, aspart
0.0075	t. thermophilus, adenin
0.0074	sugar, t. thermophilus
0.0074	amino acid, cystein
0.0074	trehalos
0.0074	nucleotid, adp
0.0074	gtp, nucleotid
0.0074	t. maritima, cystein
0.0074	amino acid, c, t. thermophilus

support	itemsets
0.0073	3-isopropylmal
0.0073	t. kodakarensi, c
0.0072	3-isopropylmal, t. thermophilus
0.0072	glucos, carbon
0.0072	mn2+
0.0072	s-adenosylmethionin
0.0072	fe, p. furiosus
0.0071	carbohydr, sugar
0.0070	sulfat
0.0070	ribonucleotid
0.0070	carbohydr, p. furiosus
0.0069	sucros
0.0069	n, c, p. furiosus
0.0069	amino acid, hydrogen
0.0069	gdp, t. thermophilus
0.0068	manganes
0.0068	asparagin
0.0068	amino acid, lysin
0.0068	quinon, t. thermophilus
0.0067	asp
0.0067	amino acid, n, p. furiosus
0.0067	alanin, amino acid
0.0067	thiol
0.0067	amino, t. thermophilus
0.0067	t. thermophilus, iron-sulfur
0.0067	t. kodakarensi, s. solfataricus, p. furiosus
0.0066	lactos
0.0066	amino acid, sugar
0.0066	t. maritima', 'adp
0.0066	amino-acid
0.0065	lactat
0.0065	amino acid, s. solfataricus, n
0.0065	polyamin
0.0065	alanin, t. thermophilus

APPENDIX C. THIRD APPENDIX

BIBLIOGRAPHY

[1] Saber A. Akhondi, Alexander G. Klenner, Christian Tyrchan, Anil K. Manchala, Kiran Boppana, Daniel Lowe, Marc Zimmermann, Sarma A. R. P. Jagarlapudi, Roger Sayle, Jan A. Kors, and Sorel Muresan. Annotated chemical patent corpus: A gold standard for text mining. 9(9):e107477.

[2] Gowtham Atluri, Rohit Gupta, Gang Fang, Gaurav Pandey, Michael Steinbach, and Vipin Kumar. Association analysis techniques for bioinformatics problems. In Sanguthevar Rajasekaran, editor, *Bioinformatics and Computational Biology*, volume 5462, pages 1–13. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science.

[3] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, and Lawrence E Hunter. Concept annotation in the CRAFT corpus. 13(1):161.

[4] Riza Batista-Navarro and Sophia Ananiadou. Adapting ChER for the recognition of chemical mentions in patents. page 5.

[5] Riza Batista-Navarro, Rafal Rak, and Sophia Ananiadou. Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. 7:S6.

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, 2019.

[7] Danze Chen, Fan Zhang, Qianqian Zhao, and Jianzhen Xu. OmicsARules: a r package for integration of multi-omics datasets via association rules mining. 20(1):554.

[8] Chihwen Cheng, Thomas G. Burns, and May D. Wang. Mining association rules for neurobehavioral and motor disorders in children diagnosed with cerebral palsy. In *2013 IEEE International Conference on Healthcare Informatics*, pages 258–263. IEEE.

[9] James A. Coker. Extremophiles and biotechnology: current uses and prospects. 5:396.

[10] Fabian M Commichau, Karl Forchhammer, and Jörg Stülke. Regulatory links between carbon and nitrogen metabolism. 9(2):167–172.

[11] Peter Corbett and John Boyle. Chemlistem: chemical named entity recognition using recurrent neural networks. 10(1):59.

[12] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj, and M. Ashburner. ChEBI: a database and ontology for chemical entities of biological interest. 36:D344–D350.

[13] Safaa Eltyeb and Naomie Salim. Chemical named entities recognition: a review on approaches and applications. 6(1):17.

[14] Wilco W.M. Fleuren and Wynand Alkema. Application of text mining in the biomedical domain. 74:97–106.

[15] A. Giuffre, G. Stubauer, P. Sarti, M. Brunori, W. G. Zumft, G. Buse, and T. Soulimane. The heme-copper oxidases of thermus thermophilus catalyze the reduction of nitric oxide: Evolutionary implications. 96(26):14718–14723.

[16] Yoav Goldberg and Joakim Nivre. A dynamic oracle for arc-eager dependency parsing. page 18.

[17] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. 45(5):885–892.

[18] Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier. OCLC: 818864884.

[19] Lezan Hawizy, David M Jessop, Nico Adams, and Peter Murray-Rust. ChemicalTagger: A tool for semantic text-mining in chemistry. 3(1):17.

[20] Marti A. Hearst. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* -, pages 3–10. Association for Computational Linguistics.

[21] María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. 46(5):914–920.

[22] Kristina M. Hettne, Rob H. Stierum, Martijn J. Schuemie, Peter J. M. Hendriksen, Bob J. A. Schijvenaars, Erik M. van Mulligen, Jos Kleinjans, and Jan A. Kors. A dictionary to identify small molecules and drugs in free text. 25(22):2983–2991.

[23] Chung-Chi Huang and Zhiyong Lu. Community challenges in biomedical text mining over 10 years: success, failure and the future. 17(1):132–144.

[24] Ming-Siang Huang, Po-Ting Lai, Pei-Yen Lin, Yu-Ting You, Richard Tzong-Han Tsai, and Wen-Lian Hsu. Biomedical named entity recognition and linking datasets: survey and our recent development. 21(6):2219–2238.

[25] Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C. Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Gokal Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong Lu. NLM-chem, a new resource for chemical entity recognition in PubMed full text literature. 8(1):91.

[26] Lars Juhl Jensen, Jasmin Saric, and Peer Bork. Literature mining for the biologist: from information retrieval to biological discovery. 7(2):119–129.

[27] David M Jessop, Sam E Adams, Egon L Willighagen, Lezan Hawizy, and Peter Murray-Rust. OSCAR4: a flexible architecture for chemical text-mining. 3(1):41.

[28] Sun-Ju Jung, Chang-Sik Son, Min-Soo Kim, Dae-Joon Kim, Hyoung-Seob Park, and Yoon-Nyun Kim. Association rules to identify complications of cerebral infarction in patients with atrial fibrillation. 19(1):25.

[29] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus–a semantically annotated corpus for bio-textmining. 19:i180–i182.

[30] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA '04*, page 70. Association for Computational Linguistics.

[31] Sun-Ki Kim, Daehwan Chung, Michael E. Himmel, Yannick J. Bomble, and Janet Westpheling. Heterologous expression of family 10 xylanases from acidothermus cellulolyticus enhances the exoproteome of caldicellulosiruptor bescii and growth on xylan substrates. 9(1):176.

[32] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. 4:313–327.

[33] Corinna Kolarˇik, Roman Klinger, Christoph M Friedrich, Martin Hofmann-Apitius, and Juliane Fluck. Chemical names: Terminological resources and corpora annotation. page 8.

[34] Ilia Korvigo, Maxim Holmatov, Anatolii Zaikovskii, and Mikhail Skoblov. Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules. 10(1):28.

[35] Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. CHEMDNER: The drugs and chemical names extraction challenge. 7:S1.

[36] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, Roger A Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, Sv Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A Akhondi, Jan A Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. The CHEMDNER corpus of chemicals and drugs and its annotation principles. 7:S2.

[37] M. Kuhn, C. von Mering, M. Campillos, L. J. Jensen, and P. Bork. STITCH: interaction networks of chemicals and proteins. 36:D684–D688.

[38] Michael Kuhn, Damian Szklarczyk, Andrea Franceschini, Monica Campillos, Christian von Mering, Lars Juhl Jensen, Andreas Beyer, and Peer Bork. STITCH 2: an interaction network database for small molecules and proteins. 38:D552–D556.

[39] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

[40] Laura L. Lee, James R. Crosby, Gabriel M. Rubinstein, Tunyaboon Laemthong, Ryan G. Bing, Christopher T. Straub, Michael W.W. Adams, and Robert M. Kelly. The biology and biotechnology of the genus caldicellulosiruptor: recent developments in 'caldi world'. 24(1):1–15.

[41] Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. BioCreative v CDR task corpus: a resource for chemical disease relation extraction. 2016:baw068.

[42] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition.

[43] Qin Li, Yiyan Zhang, Hongyu Kang, Yi Xin, and Caicheng Shi. Mining association rules between stroke risk factors based on the apriori algorithm. 25:197–205.

[44] Mark Liberman, Mark Mandel, and GlaxoSmithKline Pharmaceuticals R\&D. PennBioIE CYP 1.0. Artwork Size: 167936 KB Pages: 167936 KB Type: dataset.

[45] Jiqiang Ling, Noah Reynolds, and Michael Ibba. Aminoacyl-tRNA synthesis and translational quality control. 63(1):61–78.

[46] Daniel M Lowe and Roger A Sayle. LeadMine: a grammar and dictionary driven approach to entity recognition. 7:S5.

[47] Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. 34(8):1381–1388.

[48] David McClosky and Eugene Charniak. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Short Papers - HLT '08*, page 101. Association for Computational Linguistics.

[49] Nancy Merino, Heidi S. Aronson, Diana P. Bojanova, Jayme Feyhl-Buska, Michael L. Wong, Shu Zhang, and Donato Giovannelli. Living at the extremes: Extremophiles and the limits of life in a planetary context. 10:780.

[50] Ines Montani, Matthew Honnibal, Matthew Honnibal, Sofie Van Landeghem, Adriane Boyd, Henning Peters, Maxim Samsonov, Paul O'Leary McCann, Jim Geovedi, Jim O'Regan, György Orosz, Duygu Altinok, Søren Lind Kristiansen, Roman, Explosion Bot, Leander Fiedler, Grégory Howard, Wannaphong Phatthiyaphaibun, Yohei Tamura, Sam Bozek, Murat, Mark Amery, Björn Böing, Pradeep Kumar Tippa, Leif Uwe Vogelsang, Ramanan Balakrishnan, Vadim Mazaev, GregDubbin, Jeannefukumaru, and Walter Henry. explosion/spaCy: v3.1.3: Bug fixes and UX updates.

[51] Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 97–109. Association for Computational Linguistics.

[52] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327. Association for Computational Linguistics.

[53] Prasad S. Nishtala, Te-yuan Chyou, Fabian Held, David G. Le Couteur, and Danijela Gnjidic. Association rules method and big data: Evaluating frequent medication combinations associated with fractures in older adults. 27(10):1123–1130.

[54] Chikashi Nobata, Paul D. Dobson, Syed A. Iqbal, Pedro Mendes, Jun'ichi Tsujii, Douglas B. Kell, and Sophia Ananiadou. Mining metabolites: extracting the yeast metabolome from the literature. 7(1):94–101.

[55] Peace Ossom Williamson and Christian I. J. Minter. Exploring PubMed as a reliable resource for scholarly communications services. 107(1).

[56] Evangelos Pafilis, Sune P. Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text. 8(6):e65390.

[57] Frederick Parente and John-Christopher Finley. Using association rules to measure subjective organization after acquired brain injury. 42(1):9–15.

[58] Xiaowei Peng, Hong Su, Shuofu Mi, and Yejun Han. A multifunctional thermophilic glycoside hydrolase from caldicellulosiruptor owensensis with potential applications in production of biofuels and biochemicals. 9(1):98.

[59] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Jun'ichi Tsujii, and Sophia Ananiadou. Overview of the cancer genetics and pathway curation tasks of BioNLP shared task 2013. 16:S2.

[60] Thomas C. Rindflesch, Lorraine Tanabe, John N. Weinstein, and Lawrence Hunter. EDGAR: Extraction of drugs, genes and relations from the biomedical literature. In *Biocomputing 2000*, pages 517–528. WORLD SCIENTIFIC.

[61] Tim Rocktäschel, Michael Weidlich, and Ulf Leser. ChemSpot: a hybrid system for chemical named entity recognition. 28(12):1633–1640.

[62] Neil R. Smalheiser and Don R. Swanson. Linking estrogen to alzheimer's disease: An informatics approach. 47(3):809–810.

[63] Neil R. Smalheiser, Vetle I. Torvik, and Wei Zhou. Arrowsmith two-node search interface: A tutorial on finding meaningful links between two disparate sets of articles in MEDLINE. 94(2):190–197.

[64] Filipa L. Sousa, Renato J. Alves, Miguel A. Ribeiro, José B. Pereira-Leal, Miguel Teixeira, and Manuela M. Pereira. The superfamily of heme–copper oxygen reductases: Types and evolutionary considerations. 1817(4):629–637.

[65] Christopher T. Straub, Piyum A. Khatibi, Jonathan K. Otten, Michael W. W. Adams, and Robert M. Kelly. Lignocellulose solubilization and conversion by extremely thermophilic *Caldicellulosiruptor bescii* improves by maintaining metabolic activity. 116(8):1901–1908.

[66] Don R. Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. 30(1):7–18.

[67] Don R. Swanson. Migraine and magnesium: Eleven neglected connections. 31(4):526–557.

[68] Don R. Swanson. Somatomedin c and arginine: Implicit connections between mutually isolated literatures. 33(2):157–186.

[69] Damian Szklarczyk, Alberto Santos, Christian von Mering, Lars Juhl Jensen, Peer Bork, and Michael Kuhn. STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data. 44:D380–D384.

[70] Anabel Usié, Rui Alves, Francesc Solsona, Miguel Vázquez, and Alfonso Valencia. CheNER: chemical named entity recognizer. 30(7):1039–1040.

[71] Anabel Usié, Joaquim Cruz, Jorge Comas, Francesc Solsona, and Rui Alves. CheNER: a tool for the identification of chemical entities and their classes in biomedical literature. 7:S15.

[72] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant. PubChem: a public information system for analyzing bioactivities of small molecules. 37:W623–W633.

[73] Taiki Watanabe, Akihiro Tamura, Takashi Ninomiya, Takuya Makino, and Tomoya Iwakura. Multi-task learning for chemical named entity recognition with chemical compound paraphrasing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6243–6248. Association for Computational Linguistics.

[74] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. OntoNotes release 5.0. Artwork Size: 2806280 KB Pages: 2806280 KB Type: dataset.

[75] Libin Ye, Xiaoyun Su, George E. Schmitz, Young Hwan Moon, Jing Zhang, Roderick I. Mackie, and Isaac K. O. Cann. Molecular and biochemical analyses of the GH44 module of CbMan5b/cel44a, a bifunctional enzyme from the hyperthermophilic bacterium caldicellulosiruptor bescii. 78(19):7048–7059.

[76] Benjamin M. Zeldes, Matthew W. Keller, Andrew J. Loder, Christopher T. Straub, Michael W. W. Adams, and Robert M. Kelly. Extremely thermophilic microorganisms as metabolic engineering platforms for production of fuels and industrial chemicals. 6.