INFORMATION EXTRACTION FROM PHARMACEUTICAL LITERATURE

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2014

By
Riza Theresa Batista-Navarro
School of Computer Science
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Chemical NER</td>
<td>72</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Drug NER</td>
<td>83</td>
</tr>
<tr>
<td>2.4</td>
<td>Applications</td>
<td>90</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Chemical Document Indexing</td>
<td>90</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Comparative Toxicogenomics Database chemical concept recognition</td>
<td>92</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Availability and Interoperability</td>
<td>95</td>
</tr>
<tr>
<td>2.5</td>
<td>Summary</td>
<td>97</td>
</tr>
<tr>
<td>3</td>
<td>Coreference Resolution</td>
<td>98</td>
</tr>
<tr>
<td>3.1</td>
<td>Literature Review</td>
<td>101</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Annotated Corpora</td>
<td>101</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Approaches to coreference resolution</td>
<td>110</td>
</tr>
<tr>
<td>3.2</td>
<td>Corpus Development</td>
<td>124</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Composition of Corpus Documents</td>
<td>124</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Annotation Scope</td>
<td>125</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Annotation Scheme and Procedure</td>
<td>127</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Results and Discussion</td>
<td>131</td>
</tr>
<tr>
<td>3.3</td>
<td>Methodology</td>
<td>133</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Support vector machines (SVMs)</td>
<td>133</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Feature set</td>
<td>136</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Implementation of paradigms</td>
<td>148</td>
</tr>
<tr>
<td>3.4</td>
<td>Evaluation and Discussion</td>
<td>152</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Scoring schemes</td>
<td>153</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Pre-processing pipeline</td>
<td>160</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Training and development</td>
<td>161</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Comparison</td>
<td>162</td>
</tr>
</tbody>
</table>
4 Interaction extraction

4.1 Literature Review

4.1.1 Annotated Corpora

4.1.2 Approaches to event extraction

4.2 Corpus Development

4.2.1 Composition of Corpus Documents

4.2.2 Annotation Tasks

4.2.3 Annotation Scheme and Procedure

4.2.4 Results and Discussion

4.3 Methodology

4.3.1 Trigger Recognition

4.3.2 Argument Recognition

4.3.3 Argument Recognition with Coreferring Expressions

4.4 Evaluation and Discussion

4.4.1 Trigger Recognition

4.4.2 Argument Recognition

4.5 Summary

5 Conclusion

5.1 Evaluation of Research Objectives

5.2 Future Work

A Chemical resources

A.1 List of chemical affixes

A.2 List of chemical element symbols
B  HANAPIN Coreference Corpus annotation resources 257

B.1 Annotation guidelines ........................................... 257
   B.1.1 Named entity annotation .................................. 257
   B.1.2 Introduction to the task and annotation types ........... 257
   B.1.3 Annotation Elements ....................................... 257
   B.1.4 Guidelines and Examples .................................. 263

B.2 Coreference annotation ........................................... 269
   B.2.1 Introduction to the task and annotation types ........... 269
   B.2.2 Guidelines and Examples .................................. 275

B.3 Document Type Definition (DTD) file ........................... 279

B.4 Instructions for configuring and using XConc Suite ............. 285
   B.4.1 Installation of XConc Suite ................................. 285
   B.4.2 Accessing the corpus ....................................... 286
   B.4.3 Named entity annotation .................................... 288
   B.4.4 Coreference Annotation ..................................... 292

C  Coreference resolution resources 294

C.1 List of pronouns .................................................. 294

C.2 List of chemical compound hyponyms from WordNet .......... 295

D  HANAPIN DTI Corpus annotation guidelines 296

D.1 Pre-annotated Entities .......................................... 296

D.2 Recognising Interactions ....................................... 297
   D.2.1 Pre-marked Interaction Phrases ............................ 297
   D.2.2 Annotation of Agents ....................................... 300
   D.2.3 Annotation of Targets ....................................... 301
   D.2.4 Annotation of Interaction Clues ............................ 302
   D.2.5 Negative, Neutral and Uncertain Interactions .......... 303
List of Tables

2.1 Comparison of lexical resources for chemistry (✓ = available; x = not available, • = available but limited) .......................... 41
2.2 Comparison of Corpora with Chemical Named Entity Annotations (F_1 defined in Section 2.3) .......................... 50
2.3 Comparison of Approaches to Chemical Named Entity Recognition ................................................................. 60
2.4 Orthographic features ........................................................................................................................................ 64
2.5 Example of a sentence tokenised and labelled with parts-of-speech tags .......................................................... 65
2.6 Example of a sentence tokenised and labelled with parts-of-speech and chunk tags .................................................. 66
2.7 Example of a token sequence tagged with matches against chemical dictionaries .................................................... 69
2.8 Example of a token sequence tagged with matches against our affix lists ............................................................ 69
2.9 Examples of chemical names with corresponding basic segments ........................................................................ 70
2.10 Example of BIO-tagged tokens converted into named entity character offsets .................................................... 75
2.11 Evaluation on CHEMDNER development set of model trained on CHEMD-NER training set ................................................. 75
2.12 Distribution of instances in the CHEMDNER data set incorrectly rejected (FN) by the Custom NERsuite model .............. 76
2.13 Chemical segment composition of some sample tokens ....................................................................................... 77
2.14 Comparative evaluation of chemical NERs on CHEMDNER Test Data and SCAI Pilot Corpus ................................................. 81

2.15 Recall scores for each chemical subtype in the CHEMDNER test data ................................................. 81

2.16 Comparison against Metaboliner on the Metabolites corpus. ......................................................... 82

2.17 Summary of overall drug NER results on the entire DDI test corpus ................................................. 87

2.18 Summary of drug NER type-level results on the DDI test corpus ................................................. 87

2.19 Summary of drug NER results on the PK and EU-ADR corpora. Indicated in parentheses are the drug subtypes included in the responses; d=generic drugs, b=brand, g=group. ................................................. 90

2.20 Official BioCreative IV CHEMDNER evaluation results for the chemical document indexing task (top 3 out of 18) ................................................. 92

2.21 Approximate string matching algorithm ................................................................. 94

2.22 Official BioCreative IV CTD evaluation results for the chemical concept recognition task ................................................. 95

3.1 Comparison of Corpora with Coreference Annotations ................................................. 108

3.2 Comparison of Approaches to Biomedical Coreference Resolution ................................................. 123

3.3 Coreference Types with Examples ................................................................. 125

3.4 Nominative form lookup table ................................................................. 138

3.5 Sample reference and response chains ................................................................. 153

3.6 BLANC Confusion matrix for example in Table 3.5 ................................................. 158

3.7 Summary of macro-averaged results for each paradigm using default (d) and enriched (e) feature sets ................................................. 163

4.1 Comparison of Corpora with Biomedical EventAnnotations ................................................. 175

4.2 Comparison of Approaches to Biomedical Event Extraction ......................................................... 182

4.3 Inter-annotator agreement on drug-target interaction annotations ................................................. 189

4.4 Frequencies in the HANAPIN DTI Corpus ......................................................... 189
4.5 Examples of forms retrieved from the BioLexicon .......................... 192
4.6 Example of a token sequence tagged with matches against our trigger dictionary ................................................................. 193
4.7 Examples of syntactic frames ....................................................... 199
4.8 Examples of semantic frames ...................................................... 200
4.9 Example of varying correspondences between syntactic and semantic arguments ................................................................. 201
4.10 Examples of correspondences between syntactic and semantic arguments in the BioLexicon ...................................................... 203
4.11 Transcription from our semantic types and roles to those in the BioLexicon ................................................................. 204
4.12 Examples of correspondences between syntactic and semantic arguments retrieved based on the HANAPIN DTI training corpus .... 204
4.13 Micro-averaged mention extraction results using 10-fold cross validation on the HANAPIN Coreference Corpus ............................. 209
4.14 Evaluation on the HANAPIN DTI test set of trigger recognition model trained on training set ...................................................... 212
4.15 Summary of micro-averaged event extraction results on the HANAPIN DTI test corpus ................................................................. 213
B.1 Named entity types ................................................................. 259
List of Figures
1.1

Total number of citations added to MEDLINE per fiscal year [1] . . .

2.1

Examples of the different ways in which a compound, e.g., penicillin

24

G, can be referred to in scientific literature. SMILES (Simplified MolecularInput Line-Entry System) and InChI (International Chemical Identifier) strings are notations for describing chemical structure. . . . . . .

34

2.2

Visualisation of comparative evaluation of chemical NERs . . . . . .

80

2.3

Visualisation of comparative evaluation of drug NERs . . . . . . . . .

89

2.4

Performance of our chemical concept recogniser web service (#184)
compared with that of other participants. Graph taken with permission
from the CTD task overview paper [2]. . . . . . . . . . . . . . . . . .

96

3.1

Sample annotations as shown in the XConc annotation tool. . . . . . . 127

3.2

XML code generated by XConc for the sample annotations in Figure 3.1.129

3.3

Distribution of semantic and coreference types in the HANAPIN corpus.131

3.4

Partial syntactic parse tree showing M. jannaschii as a nested noun
phrase. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 137

3.5

Syntactic parse tree showing The lead compound as the subject of the
sentence head inhibited. . . . . . . . . . . . . . . . . . . . . . . . . . 138

3.6

Partial syntactic parse tree showing extract as the head noun of the NP
cell extract. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
10


3.7 Partial syntactic parse tree showing the appositional relationship between the mentions Sarcotriol and a semi-synthetic derivative of sarcophine. 141

3.8 Partial syntactic parse tree showing the indefinite noun phrase beginning with a Lessepsian immigrant. 142

3.9 Comparative evaluation results using the MUC scoring scheme. 164

3.10 Comparative evaluation results using the B3 scoring scheme. 165

3.11 Comparative evaluation results using the BLANC scoring scheme. 165

4.1 An interaction where the patient is an entity: agent = Poly-APS, interaction clue = inhibitors and patient = AChE. 168

4.2 An interaction where the patient is a biological process: agent = Bisebromoamide, interaction clue = inhibits and patient = ERK phosphorylation. 168

4.3 Drug-target interactions represented as events (left-hand side) and as relations (right-hand side). 169

4.4 A drug-target interaction annotated using our scheme, as shown in the Manual Annotation Editor of Argo (inhibited as INTERACTION, bisebromoamide as AGENT, phosphorylation of ERK as TARGET and phosphorylation as BIOPROCESS). 188

4.5 The equivalent event representation of the interaction annotations in Figure 4.4 generated by our heuristic-based converter, as visualised in the brat annotation tool. 188

4.6 Example of an event centred on inhibited whose arguments correspond to the verb’s syntactic arguments. 196

4.7 Example of an event centred on inhibition which is an argument of the preposition of whose other argument is the event’s theme. 197
B.1 Setting up the DTD File ........................................... 286
B.2 Setting Up the CSS File ........................................... 287
B.3 Setting Up the Default XML editor ......................... 287
B.4 Structure of an Abstract ....................................... 288
B.5 Selecting the Type of Element for Insertion ............ 289
B.6 Selecting the named entity attributes ....................... 290
B.7 Annotating embedded terms .................................. 291
B.8 Annotating terms in a discontinuous text region ........ 291
B.9 Annotating the referring expression ......................... 293
B.10 Annotating the referring expressions - fragment ....... 293

D.1 Pre-annotated marine drugs (in turquoise) and enzymes (in salmon) . 296
D.2 A span of text that does not describe an interaction because of a missing element, the agent ........................................ 297
D.3 Pre-marked phrases in yellow .................................... 298
D.4 The PHRASE label for phrases which do not have interaction clues should be removed ........................................ 298
D.5 An entire sentence annotated by our automatic phrase extractor. The annotated text needs to be trimmed down ........................................ 298
D.6 An example of a minimal phrase that includes context .......... 299
D.7 References should be removed from the annotated phrase ........................................ 300
D.8 Entities acting as agents in brown ................................. 300
D.9 Only instances that are causing an interaction should be annotated as AGENTS ........................................ 301
D.10 Entities acting as targets in blue ............................... 301
D.11 Target bioprocess in blue ........................................ 302
D.12 Only instances that are at the receiving end of an interaction should be annotated as TARGETs. 302
D.13 Interaction clue in green. 303
D.14 A neutral interaction. 303
D.15 An uncertain interaction. 304
D.16 A negative interaction. 304
D.17 An example of inter-sentential interaction. 304
D.18 Annotation of inter-sentential interactions. 305
D.19 Mistakes committed by the authors should be ignored. 305
D.20 The comment box. 306
Abstract

**INFORMATION EXTRACTION FROM PHARMACEUTICAL LITERATURE**
Riza Theresa Batista-Navarro
A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy, 2014

With the constantly growing amount of biomedical literature, methods for automatically distilling information from unstructured data, collectively known as information extraction, have become indispensable. Whilst most biomedical information extraction efforts in the last decade have focussed on the identification of gene products and interactions between them, the biomedical text mining community has recently extended their scope to capture associations between biomedical and chemical entities with the aim of supporting applications in drug discovery. This thesis is the first comprehensive study focussing on information extraction from pharmaceutical chemistry literature. In this research, we describe our work on (1) recognising names of chemical compounds and drugs, facilitated by the incorporation of domain knowledge; (2) exploring different coreference resolution paradigms in order to recognise co-referring expressions given a full-text article; and (3) defining drug-target interactions as events and distilling them from pharmaceutical chemistry literature using event extraction methods.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property
and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Acknowledgements

I am immensely grateful to Prof. Sophia Ananiadou, not only for the supervision, research guidance and constructive criticism she has provided me as her student, but most especially for her understanding and encouragement during more challenging times. She has exceeded my expectations by opening several opportunities for me to learn and grow as a young researcher.

I am deeply indebted (quite literally) to the University of the Philippines who provided the financial support for this research through the Faculty Development Program of the Engineering Research and Development for Technology Consortium.

Pursuing a PhD degree in a country completely different from where I came from, I would have found it very difficult to cope with the challenges of postgraduate research if not for my colleagues at the National Centre for Text Mining. Special mention goes to Claudiu, George, Noha, Paul, Rafal and Tomoko, who have constantly given me their encouragement and who have all been like family to me. I would like to especially thank my collaborator and mentor, Rafal, who has generously provided me with his invaluable research input and support.

It definitely would not have been even possible for me to pursue this research without my family. I have no words to express how grateful I am to my ever-supportive mother who unselfishly encouraged me to pursue my dreams.

Lastly, words cannot describe how thankful I am to Jordan and Chip for their tremendous patience and understanding, and for sharing this entire journey with me.
Abbreviations

ACE  Automatic Content Extraction
ADE  Adverse drug effect
API  Application programming interface
BioNLP  Biomedical Natural language processing
CALBC  Collaborative Annotation of a Large Biomedical Corpus
ChEBI  Chemical Entities of Biological Interest
CHEMDNER  Chemical compound and drug name recognition
CRFs  Conditional random fields
CTD  Comparative Toxicogenomics Database
CTD  Comparative Toxicogenomics Database
DDI  Drug-drug interaction
DEI  Drug-target interaction
DSV  Delimiter-seperated values
EE  Event extraction
FN  False negatives
FP  False positives
FTP  File Transfer Protocol
GGPs  Genes and gene products
GREC  Gene Regulation Event Corpus
HPSG  Head-driven phrase structure grammar
i2b2  Informatics for Integrating Biology and the Bedside
IE  Information extraction
IR  Information Retrieval
JNLPBA  Joint Workshop on Natural Language Processing in Biomedicine and its Applications
Jochem  Joint Chemical Dictionary
LCS  Longest common subsequence
LDC  Linguistic Data Consortium
MCC  Mention-chain classification
MCR  Mention-chain ranking
MEMM  Maximum entropy Markov model
MeSH  Medical Subject Headings
ML  Machine learning
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC</td>
<td>Mention pair classification</td>
</tr>
<tr>
<td>MPR</td>
<td>Mention pair ranking</td>
</tr>
<tr>
<td>MUCSS</td>
<td>MUC scheme</td>
</tr>
<tr>
<td>NER</td>
<td>Named entity recognition</td>
</tr>
<tr>
<td>NIH</td>
<td>National Institutes of Health</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NP</td>
<td>Noun phrase</td>
</tr>
<tr>
<td>NYT</td>
<td>New York Times</td>
</tr>
<tr>
<td>OBO</td>
<td>Open Biomedical Ontology</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>PK</td>
<td>Pharmacokinetics</td>
</tr>
<tr>
<td>POS</td>
<td>Part-of-speech</td>
</tr>
<tr>
<td>PPI</td>
<td>Protein-protein Interactions</td>
</tr>
<tr>
<td>RE</td>
<td>Relation extraction</td>
</tr>
<tr>
<td>RSC</td>
<td>Royal Society of Chemistry</td>
</tr>
<tr>
<td>SBEP</td>
<td>Stanford Biomedical Event Parser</td>
</tr>
<tr>
<td>SD</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>SGML</td>
<td>Standard Generalized Markup Language</td>
</tr>
<tr>
<td>SVMs</td>
<td>Support vector machines</td>
</tr>
</tbody>
</table>
TEES  Turku Event Extraction System

TP    True positives

UIMA  Unstructured Information Management Architecture

WSJ   Wall Street Journal

XML   Extensible Markup Language
Chapter 1

Introduction

Keeping abreast of recently published developments in biomedicine is now considered a tedious task. For the last five years, an annual average of 713,946 citations have been added to MEDLINE, the US National Library of Medicine’s database of bibliographic references [1]. This average is bound to increase, as indicated by the generally consistent rise in the number of citations added annually for the last 15 years (Figure 1.1). MEDLINE currently contains over 21 million bibliographic entries which are mostly on subjects pertinent to biomedicine [3]. With the copious biomedical literature, the need for a means to automatically distill information of interest from unstructured data arises.

In response to this need, many researchers in the field of natural language processing (NLP) have taken interest in developing methods specifically for the biomedical domain – a research area which is now known as Biomedical NLP (BioNLP). As the goal is to ultimately eliminate the need for a reader to manually examine thousands of relevant documents, the majority of the methods developed were geared towards supporting information extraction (IE). Not to be confused with information retrieval (IR), IE is the umbrella term for any task that automatically extracts information from unstructured data. It is the process of selecting relevant information from text and
presenting it in a concise form that can be easily processed both by humans and machines. Unlike IR which finds documents relevant to a user-given search query, IE returns facts of interest contained in the documents, eliminating the need for full, manual examination by a human reader. Requiring the analysis of natural language, IE is one of the typical applications of NLP and includes tasks such as named entity recognition (NER) and relation extraction (RE). NER pertains to the automatic labelling of names of concepts whilst RE deals with the discovery of binary associations between them. Named entity recognisers for general domain documents typically recognise names of persons, organisations, geopolitical entities and geographical locations; the associations between them (e.g., Sonia Sotomayor_{person} as Justice of the US Supreme Court_{geopolitical_entity}) are then identified by relation extractors. In the last five years, there has also been growing interest in event extraction (EE) which is more powerful than RE as it can find and reconstruct facts of a more complicated structure (e.g., n-ary and nested typed associations called events).

Currently available BioNLP tools and resources can be broadly categorised according to domains of interest, specifically into genomic and clinical. Genomic NER
tools such as ABNER \[4\], BANNER \[5\] and NEMine \[6\] are capable of recognising names of genes or gene products (GGPs), e.g., proteins and complexes, whilst relation extractors find protein-protein interactions (PPIs) described in documents \[7\]–\[9\]. Event extraction systems like EventMine \[10\] and the Turku Event Extraction System \[11\], addressing the extraction of biomolecular events (e.g., phosphorylation, binding, regulation, gene expression), also fall under genomic BioNLP. To encourage the benchmarking of the methods proposed by different research groups, challenges such as the BioCreative \[12\] and the BioNLP Shared Task series \[13\] have been organised. The focus of BioCreative was on solutions for gene normalisation \[14\] and PPI extraction \[15\]–\[16\] whilst that of the BioNLP Shared Tasks is the extraction of events involving GGPs (e.g., gene regulation \[17\] and post-translational protein modifications \[18\]). Many of the genomic BioNLP tools were trained and evaluated on the data sets made available for these challenge evaluations as well as on popular annotated corpora (e.g., PennBioIE \[19\], GENETAG \[20\], BioInfer \[21\] and GENIA \[22\]) which consist of documents from the molecular biology subdomain.

Of a lesser prevalence are tools and resources for clinical BioNLP, which deals with the extraction of information from clinical narratives or medical records. Except for MetaMap \[23\], a tool that recognises UMLS Metathesaurus \[24\] concepts in text (e.g., anatomical structures, symptoms, age groups), most available clinical NLP resources and tools are outcomes of shared tasks such as the Informatics for Integrating Biology and the Bedside (i2b2) NLP challenges\(^1\). Examples of these are corpora and systems for the identification of concepts relevant to smoking \[25\]–\[26\] and obesity \[27\]–\[28\], as well as for the extraction of relations amongst medical problems, treatments and tests \[29\].

In this research, the focus will be shifted into what will be referred to as chemical BioNLP, i.e., the study of different NLP methods for the extraction of information

\(^1\)https://www.i2b2.org
revolving around biologically relevant chemical entities. Specifically, we have chosen the subdomain of pharmaceutical chemistry which is mostly concerned with interactions between potential drugs (i.e., chemical compounds) and drug targets (i.e., GGP$s such as enzymes). We focus on the development of methods for three information extraction tasks, namely, (1) chemical named entity recognition, (2) chemical coreference resolution and (3) the extraction of interactions between drugs and targets.

1.1 Motivation

The recognition of chemical entities and associated concepts, e.g., other chemicals, GGP$s and diseases in text supports the curation of chemoinformatic databases [30–33], the profiling and screening of compounds according to their biological activities [34, 35], the detection of drug adverse reactions [36, 37], drug repositioning [38–42] and the understanding of metabolic pathways [43], amongst many other applications to chemoinformatics, drug discovery and systems biology. Whilst most of the biomedical text mining efforts in the last decade have focussed on the identification of genes, their products and the interactions between them, there has been a recent surge in interest in extracting information from chemical literature. In the recently concluded Fourth BioCreative Challenge Evaluation Workshop, two tracks involved the recognition of chemical entities in abstracts [2, 44]. The sizable number of participating teams in these tracks, both academic and commercial, shows that members of the NLP community are now actively exploring research questions in chemical BioNLP.

The domain of chemistry is a specialised, scientific field whose literature is lexically and linguistically challenging. Chemical molecules, which are at the heart of this domain, are referred to in text using various conventions. These include systematic names, trivial names, brand names, company codes, abbreviations, database identifiers, and structures, not to mention anaphoric expressions [45, 46]. Although
1.1. MOTIVATION

a standard exists (i.e., the International Union of Pure and Applied Chemistry or IU-PAC nomenclature), different authors express their findings in different ways: biologists prefer simpler expressions such as trivial and brand names to improve readability, while chemists typically conform with nomenclature [35]. Spelling and orthographic variants are likely to emerge, even from authors who adhere to the use of only nomenclature-based names. The sublanguage used in the domain of chemistry exhibits such intricacies; comprehensively capturing mentions of chemical compounds and their interactions with other entity types is hence not straightforward and requires domain knowledge.

Previous studies have shown that different biomedical sublanguages exhibit semantic and linguistic variations [47–49]. We have especially demonstrated that pharmacology, the umbrella domain of pharmaceutical chemistry, is different from other subdomains. In our recent study on the analysis of variations between pharmacology and cell biology [50], we established that the two subdomains are semantically diverse, allowing a classifier which was trained solely on entity type frequencies as features, to discriminate between full-text articles from the two subdomains with highly satisfactory precision and recall. Extending our study to more subdomains [51], we also determined that pharmacology is the most discernible amongst 20 biomedical subdomains (i.e., the least alike every other subdomain) based on named entity type distributions. These findings suggest that existing tools developed and evaluated for one biomedical subdomain are not necessarily trivially adaptable to other subdomains. Careful adaptation, if not reformulation, of existing methods tuned for genomic BioNLP, for example, is needed when shifting the focus to chemical BioNLP.

Considering these challenges, it is not surprising that there has been no prior work focussing on information extraction tasks for the pharmaceutical chemistry domain. To the best of our knowledge, ours is the first effort to investigate chemical coreference resolution and drug-target interaction extraction.
1.2 Research Hypotheses and Objectives

To address the gaps mentioned above, we have raised the following research questions:

Q₁ What is the state-of-the-art in chemical and drug named entity recognition (NER)?

Q₂ What improvements can we introduce to existing approaches in order to achieve better performance in chemical and drug NER?

Q₃ What are the existing approaches to general and biomedical coreference resolution?

Q₄ Which amongst different approaches to coreference resolution is most suitable for the pharmaceutical chemistry domain?

Q₅ How can the task of extracting drug-target interactions be accomplished?

Q₆ Does the incorporation of coreference resolution into interaction extraction improve the performance of the latter?

We undertake this study with the overall aim of investigating approaches to named entity recognition, coreference resolution and interaction extraction for the domain of pharmaceutical chemistry. In embarking on this research, we formulated the following research hypotheses:

H₁ Existing methods in BioNLP can be adapted to information extraction from pharmaceutical chemistry literature with the incorporation of domain knowledge

H₂ Interactions between drugs and targets can be cast as events, and that they can be extracted using an event-based approach

Founded on these hypotheses, the following research objectives were established:

O₁ To conduct a comprehensive review of existing lexical resources, annotated corpora and approaches for chemical and drug named entity recognition
1.3. OVERVIEW OF CONTRIBUTIONS

O2 To develop chemical and drug named entity recognisers which are enabled by domain knowledge

O3 To conduct a comprehensive review of existing corpora and approaches for general and biomedical coreference resolution

O4 To develop implementations of different approaches to coreference resolution and to adapt them for the pharmaceutical chemistry domain

O5 To develop an event extraction methodology for capturing drug-target interactions

O6 To incorporate coreference resolution into our event extraction methodology and observe its impact on performance

We describe in the succeeding chapters how each of these objectives was achieved. Chapter 2 focusses on the accomplishment of objectives O1 and O2 whilst Chapter 3 provides details on how objectives O3 and O4 were fulfilled. Lastly, we show in Chapter 4 how we achieved objectives O5 and O6.

1.3 Overview of contributions

The contributions of this research are summarised in the following points.

C1 Development of chemical and drug named entity recognisers

C2 Development of the HANAPIN Coreference Corpus, a corpus annotated with coreference information

C3 Implementations of different coreference resolution approaches optimised for the pharmaceutical chemistry domain
C_4 The first comprehensive study on coreference resolution for a chemical subject domain

C_5 Development of the HANAPIN Drug-Target Interaction Corpus, a corpus annotated with event information

C_6 Development of a methodology for extracting drug-target interactions as events, with methods for coreference resolution incorporated

C_7 The first comprehensive study on drug-target interaction extraction from pharmaceutical chemistry literature
Chapter 2

Named Entity Recognition

Named entity recognition (NER), a subtask of information extraction, aims to determine the specific locations and semantic categories of named elements in text. Since information contained in text is usually centred on particular named entities, automatically recognising them is a fundamental task.

Having been considered as a yet unsolved important challenge in the 1990s, NER was included for the first time in the Sixth Message Understanding Conference (MUC-6) funded by the Defense Advanced Research Projects Agency (DARPA). This shared task fostered a competitive atmosphere for the development of pioneer work on named entity recognition, mostly focussed on the recognition of entities from news reports. The NER task called for the recognition of names of organisations, persons, locations, temporal expressions (i.e., dates, times) and numerical expressions (i.e., monetary, percentage values) from news reports on the subjects of negotiation of labour disputes and corporate management succession [52]. In the Seventh MUC, similar requirements were defined but used data sets focussed on the topics of airplane crashes and aircraft launches instead [53].

In the early 2000s, the Automatic Content Extraction (ACE) program was conceived with the goal of developing content extraction techniques, including NER, for
supporting automatic processing of text from general domain sources such as news reports, broadcast conversations and discussion fora [54]. The scope of the 2004 edition of the program included entity types such as persons, organisations, locations, geo-political entities, facilities, vehicles and weapons. The majority of general domain named entity recognisers which are currently available have adopted the same typology of named entities.

An NER for general domain documents is expected to annotate the sentence in Example 2.1 with the following information: (1) the specific boundaries of the text spans corresponding to named entities, indicated in the example in square brackets, and (2) the appropriate semantic types that should be assigned to them, indicated as subscripts. A method for NER must be able to automatically predict, for instance, that the text spanning the first two tokens of the sentence \((\text{United Nations})\) is a named entity, and that it is the name of an organisation.

**Example 2.1.** [United Nations]_{\text{Org}} humanitarian chief [Valerie Amos]_{\text{Per}}, visiting the devastated city of [Tacloban]_{\text{GPE}} entity, said the situation was desperate with residents left without food or fresh water for five days.

Biomedical named entity recognisers capture similar information but for entity types which include, but are not limited to, genes, proteins, chemical compounds, anatomical entities, diseases and species [55]. The majority of the biomedical NERs have been directed towards the recognition of genes and gene products, e.g., ABNER [4], BANNER [5] and GENIA Tagger [56]. Challenge evaluations such as BioCreative [12], TREC Genomics [57] and JNLPBA [58] also focussed on genomic NER. Furthermore, a systematic review of NER for biomedical literature [59] has established that the majority of the published biomedical NERs were developed and evaluated using the the GENIA corpus which is composed of 1,999 abstracts from the domain of molecular biology.
Chemical NER, concerned with the recognition of names of chemical compounds and/or drugs, is an interesting and challenging problem for a number of reasons, the first being the significant number of forms in which each chemical entity may appear in text. These can range from nomenclature-based expressions (i.e., IUPAC names, SMILES and InChI strings) to the more commonly used trivial and brand names, to the more obscure abbreviations and identifiers (Figure 2.1). Moreover, due to on-going drug discovery efforts, novel lead compounds are continuously being reported in the literature, leading to the constantly growing number of chemical compound names.

The recognition of named entities in scientific discourse can also be complicated by ambiguous tokens in text. Ambiguity arises when a token has several meanings or can be interpreted in more than one way. Looking at the context in which it is used in discourse usually allows a reader to disambiguate a token.

**Example 2.2.** The mechanisms by which early chronic [lead]Chemical ([Pb]Chemical) exposure alter [brain]AnatomicalEntity development have not been identified.

In Example 2.2, the token *lead* pertains to the chemical element with symbol *Pb*. Within the context of this sentence, *lead* will be ideally captured by a chemical named entity recogniser that has periodic table elements within its scope. The same token, however, has a completely different meaning in Example 2.3, where it signifies leadership. A chemical NER tool will not capture any named entities in this example.

**Example 2.3.** The [United States]GPEntity took the lead in extending aid to the victims of the typhoon which swept through the [Philippines]Location.

In the next section, we present a review of the state-of-the-art. Since the focus of this research is on information extraction methods for documents from pharmaceutical chemistry, the following review of resources and techniques focusses on notable work relevant to the domain of chemistry.
### Penicillin G

<table>
<thead>
<tr>
<th>Generic name</th>
<th>Benzylpenicillin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand name</td>
<td>Pentids®, Pfizerpen®</td>
</tr>
<tr>
<td>Systematic or IUPAC name</td>
<td>(2S,5R,6R)-3,3-dimethyl-7-oxo-6-(2-phenylacetamido)-4-thia-1-azabicyclo[3.2.0]heptane-2-carboxylic acid</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>PCG</td>
</tr>
<tr>
<td>CAS Registry number</td>
<td>61-33-6</td>
</tr>
<tr>
<td>SMILES string</td>
<td>[H][C@]12SC(C)(C)<a href="N1C(=O)%5BC@@H%5D2NC(=O)CC1=CC=CC=C1">C@@H</a>C(O)=O</td>
</tr>
<tr>
<td>InChI string</td>
<td>InChI=1S/C16H18N2O4S/c1-16(2)12(15(21)22)18-13(20)11(14(18)23-16)17-10(19)8-9-6-4-3-5-7-9/h3-7,11-12,14H,8H2,1-2H3,(H,17,19) (H,21,22)/t11-,12+,14-/m1/s1</td>
</tr>
<tr>
<td>Anaphoric expression</td>
<td>It is usually administered intravenously.</td>
</tr>
</tbody>
</table>

Figure 2.1: Examples of the different ways in which a compound, e.g., penicillin G, can be referred to in scientific literature. SMILES (Simplified Molecular-Input Line-Entry System) and InChI (International Chemical Identifier) strings are notations for describing chemical structure.

## 2.1 Literature Review

### 2.1.1 Lexical Resources

Lexical resources such as domain-specific databases and ontologies serve as sources of semantic knowledge. Many of them are publicly available and provided in computable formats, making it possible for developers of NER methods (as well as corpora) to leverage the semantic information they contain. We review some of the chemical resources with emphasis on those which are relevant to this study.

#### 2.1.1.1 Chemical Databases

**PubChem.** Hosted by the US National Institutes of Health (NIH) since September 2004, PubChem is an open repository of information on small molecules and their
biological properties [60]. It consists of three databases, namely, PubChem Compound, PubChem Substance and PubChem BioAssay [61] [60]. Currently containing 48 million entries, PubChem Compound stores information on names and structures of chemical compounds. It is complemented by PubChem Substance which serves as a repository of information on 126 million specific samples of compounds. This, in turn, is complemented by PubChem BioAssay which stores the results of biological activity testing of substances, currently containing about 730,000 records. Most of the data in PubChem come from submissions of various contributors; data deposited to PubChem is processed automatically to eliminate the costs of manual curation. All of the information is downloadable via their FTP (File Transfer Protocol) site[^1].

**ChemBank.** Similar in terms of data content to PubChem is ChemBank, a knowledge and analysis environment for small molecules and assays developed jointly by the Chemical Biology Program and Chemical Biology Platform at the Broad Institute of Harvard and MIT [62]. It contains chemical structures and names, molecular descriptors and information on biological activities of small molecules. It also contains other elements which make it unique and different from other small molecule databases, namely, raw screening results from high-throughput biological assays and hierarchical metadata for describing screening experiments. Over 1.2 million unique chemical structures of small molecules are contained in ChemBank (as of August 2007). Not all of the information, however, is available in an easily downloadable format.

**Comparative Toxicogenomics Database (CTD).** CTD is a publicly available resource that integrates information on chemicals, genes and diseases curated from scientific literature, aiming to foster understanding of the means by which drugs and chemicals affect human health [32]. Information on entities and relationships between them[^32].

(e.g., chemical-gene, chemical-disease and gene-disease) are stored in the database by means of text mining-assisted manual curation. Its chemical vocabulary, currently containing more than 150,000 unique entities, was largely derived from the US National Library of Medicine’s Medical Subject Headings (MeSH), albeit structured differently (i.e., as a polyhierarchic tree in which a node may belong to more than one branch). All of the vocabularies are downloadable in formats such as XML and delimiter-separated values (DSV)².

**ChemSpider.** ChemSpider, developed by the Royal Society of Chemistry, is a freely accessible database that holds chemical information on almost 25 million compounds [63]. Serving as a chemical information aggregator, it automatically collates data from over 450 various sources. Data quality is ensured by means of expert curation, realised as crowdsourced tasks to which practising chemists contribute their domain knowledge and expertise. Full download of ChemSpider is currently not possible; however, apart from a web interface, a suite of web services has been made available to enable users to query the database and obtain specific information³.

**ChemIDplus.** Developed by the US National Library of Medicine, ChemIDplus is an online search engine covering over 390,000 records of chemical substances [64]. It forms the basis for some of the NLM’s other resources, such as the Toxicology Data Network. None of the information, however, is available for download; users can access the data only through the web interface⁴.

**DrugBank.** As a resource focused on the storage of information on drugs and their respective targets, DrugBank provides fine-grained and highly structured knowledge

---

²http://ctdbase.org/downloads  
⁴http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp
on pharmacological substances, clinically approved or otherwise [65]. Currently containing almost 7,000 unique entities, it stores comprehensive information on each compound in a DrugCard which holds the following attributes: identification (chemical structure), taxonomy (chemical families), pharmacology (mechanism of action), pharmacoeconomics (manufacturers), physical properties and interactions (drug targets). Although automatic tools assist the DrugBank team in populating the DrugCards with information, a thorough manual validation process follows to ensure high quality of the data. The database is available for public access\footnote{http://www.drugbank.ca/downloads}.

Other databases containing compounds such as LIGAND (a database of chemical compounds and reactions in biological pathways) [66] and the Human Metabolome Database (HMDB) [67] are also publicly available, although these are even more specialised in terms of content. The Kyoto Encyclopedia of Genes and Genomes (KEGG) [68] contains information on chemical compounds and drugs but is available for download only to paying subscribers.

\subsection{Ontologies and Dictionaries}

\textbf{Chemical Entities of Biological Interest (ChEBI).} Another resource is Chemical Entities of Biological Interest (ChEBI), a publicly available repository of chemical compounds whose focus is on both natural and synthetic small molecules which intervene in the processes of living organisms [69, 70]. It has been constantly populated since 2002 by the European Bioinformatics Institute (EBI) and currently contains entries for 36,000 compounds\footnote{http://www.ebi.ac.uk/chebi/statisticsForward.do}. A unique feature of CheBI is its organisation of the entities according to an ontological classification, by which the relationships among them are specified.

An ontology is a specification of a vocabulary for a shared domain of discourse,
including machine-interpretable definitions of classes, relationships and other objects [71]. It is used as a means of sharing and annotating information in specific domains. The ChEBI Ontology consists of three sub-ontologies, namely, chemical entity, subatomic particle and role. Whilst chemical entity pertains to chemical molecules classified according to structure, subatomic particle covers only particles which are smaller than atoms. The role sub-ontology contains the classification of entities according to their biological and chemical roles, as well as their intended use by humans. Aside from standard ontological relationships such as is a and is part of, ChEBI relationships include domain-specific ones such as is conjugate acid of, is conjugate base of, is tautomer of, is enantiomer of, has functional parent, has parent hydride and is substituent group from. Both the ChEBI database and ontology can be downloaded from the EBI FTP site\textsuperscript{7} in several formats (e.g., database dumps for the database and OBO and OWL for the ontology).

**ChemAxiom.** ChemAxiom is another ontology for chemistry which aims, among other things, to describe chemical objects and data. This formal ontology consists of sub-ontologies such as ChemAxiomChemDomain (fundamental concepts), ChemAxiomProp (properties), ChemAxiomMetrology (measurements), ChemAxiomPoly and ChemAxiomPolyClass (polymers). ChemAxiom was the first ontology developed for describing concepts in general chemistry [72].

**Joint Chemical Dictionary (Jochem)** Initially called Chemlist, the Joint Chemical Dictionary (Jochem) was developed by the BioSemantics group of the Erasmus University Medical Center in Rotterdam, Netherlands [73]. Compiled for the purpose of identifying small molecules and drugs in text, Jochem was formed by a systematic combination of information from several sources and databases such as UMLS, MeSH,
2.1. LITERATURE REVIEW

ChEBI, DrugBank, KEGG, HMDB and ChemIDplus. The combined dictionary contains a total of more than 2 million terms. Its contribution to a term identification task was evaluated by means of dictionary-based matching against the SCAI pilot corpus (reviewed in Section 2.1.2). Results show that Jochem obtained a precision of 67% and a recall of 40% (defined in Section 2.3) and that using it as a lexicon yields better performance for OSCAR 3, a recogniser for chemical entities (reviewed in Section 2.1.2). According to their experiments, the best performance is achieved when the entries from PubChem were left out. This was attributed to the fact that PubChem is the only data source whose information is not manually curated.

2.1.1.3 Comparison

Presented in Table 2.1 is a comparison in terms of availability and granularity of the chemical information provided by the different resources reviewed above. Although the ChemAxiom ontology was designed to describe information on chemical entities, it is currently not utilised as a data store (unlike the ChEBI ontology which is tightly coupled with the ChEBI database) and has not been included in this comparison.

We signify restricted information accessibility with bullet points, such as the case with CAS Registry number for PubChem, for instance. Whilst the CAS Registry number is available in PubChem for each compound, the lack of metadata prevents users to non-trivially distinguish this specific detail from the other types of information. In a similar manner, CTD has bullet points for all of abbreviation, systematic, trivial and brand names. This is due to the absence of a fine-grained structure according to which the known names of a chemical could have been provided. Instead, all synonyms were given in one flat list, with no differentiation between the various chemical name types.

It can be observed that ChEBI and DrugBank provide the most fine-grained and the most types of information on chemical compound and/or drug identification. We also take note that across all resources, chemical name abbreviation is the information type
which, although consistently stored, was never indicated to be the short form of another name. This, in our opinion, merits attention and is something that should be addressed by developers of lexical resources, given that chemical abbreviations are some of the most difficult cases for named entity recognisers (discussed in Section 2.3). Five of the resources can be downloaded by the public in full, i.e., PubChem, CTD, DrugBank, ChEBI and Jochem.
Table 2.1: Comparison of lexical resources for chemistry (✓ = available; ✗ = not available, • = available but limited)

<table>
<thead>
<tr>
<th>Resource</th>
<th>Systematic names</th>
<th>Trivial names</th>
<th>Brand</th>
<th>Abbreviation</th>
<th>CAS Registry number</th>
<th>Structure</th>
<th>Molecular formula</th>
<th>External links</th>
<th>Fully downloadable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PubChem</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ChemBank</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>•</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>CTD</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>✓</td>
<td>•</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ChemSpider</td>
<td>✓</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ChemIDplus</td>
<td>✓</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>DrugBank</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>•</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ChEBI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>•</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Jochem</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
2.1.2 Annotated Corpora

The development of publicly available, gold-standard corpora for chemical named entity recognition has received relatively less attention [74] compared to that for gene or protein recognition. Until recently, there were only very few corpora containing chemical named entity annotations, namely, SciBorg, SCAI and the Patents corpora. However, with the recent rise of interest in chemical NLP, more have emerged. Details on how each of these corpora were developed and evaluated are summarised below.

2.1.2.1 Chemical compound annotations

SciBorg Corpus. Corbett et al. [75] developed an annotated corpus for chemistry consisting of 42 full-text journal papers from the Royal Society of Chemistry (RSC). The annotation schema required an annotator to classify a span of text as one of the following: chemical compound (CM), chemical reaction (RN), chemical adjective (CJ), enzyme (ASE) and chemical prefix (CPR). The resulting annotations show that the majority (94%) of the annotated entities fall under the CM category. The corpus is not publicly available due to copyright restrictions on the full articles.

SCAI Corpora. Kolářík et al. [76] built a corpus consisting of MEDLINE abstracts which was developed in two steps. First, a pilot set of 100 abstracts were annotated to gain insight on the types of chemical names found in MEDLINE abstracts and their respective frequencies. For this, the annotators were asked to mark up chemical entities, classifying each as any of IUPAC or IUPAC-like name (IUPAC), partial IUPAC class name (PART), trivial name (TRIVIAL), abbreviation (ABB), sum formula (SUM) and chemical family name (FAMILY). A total of 1,206 chemical names were annotated in the pilot corpus, with the TRIVIAL and IUPAC types having the highest frequencies. Satisfied with the results on the pilot corpus, the authors proceeded to the annotation of a set of abstracts intended to support the recognition of IUPAC and IUPAC-like
names. This set was divided into training and test sets. In the 463 abstracts in the training set, a total of 3,712 IUPAC and PART entities were annotated. The test set, in contrast, consisting of 1,000 abstracts which were carefully sampled so as to represent the entirety of MEDLINE, contains only 151 IUPAC and PART annotated entities. Each of the pilot, training and test sets is publicly available.\(^8\)

**Patents Corpus.** A corpus consisting of 40 patent documents\(^7\) from the chemical domain was annotated by a collaborative team of curators from the European Patent Office (EPO) and the Chemical Entities of Biological Interest (ChEBI) database. The annotation task involved the tagging of anything that falls under the notion of a chemical entity in the documents. A total of 11,162 such entities were annotated in the corpus, which is available for public download\(^9\). Details of the inter-annotator agreement on this corpus as well as its availability have not been disclosed.

**BioCreative IV CHEMDNER Corpus.** The CHEMDNER corpus\(^4\), developed as a resource for the chemical compound and drug name recognition (CHEMDNER) task of BioCreative IV\(^10\) consists of 10,000 MEDLINE abstracts with a total of 84,355 chemical or drug name annotations corresponding to 19,805 unique entities. Comprising the corpus are abstracts coming from a wide range of disciplines relevant to chemistry (e.g., medicinal chemistry, organic chemistry, toxicology, pharmacology and pharmacy). The annotation guidelines called for the categorisation of each chemical or drug name mention into one of the following eight classes: systematic, trivial, family, abbreviation, formula, identifier, multiple (i.e., coordinated mentions) and a catch-all category for unclassifiable ones. The annotation task was carried out by experienced

---

\(^8\)http://www.scai.fraunhofer.de/en/business-research-areas/bioinformatics/research-development/information-extraction-semantic-text-analysis/named-entity-recognition/chemcorpora.html


\(^10\)http://www.biocreative.org/news/biocreative-iv
domain experts with the use of the AnnotateIt tool.\footnote{http://annotateit.org} The corpus, split into training (3,500), development (3,500) and test (3,000) sets, is available upon registration with BioCreative.

**CALBC Corpus.** As a result of the Collaborative Annotation of a Large Biomedical Corpus (CALBC) project\footnote{http://www.biocreative.org/resources/corpora/bc-iv-chemdner-corpus}, a large-scale biomedical silver standard corpus has been produced. It contains annotations resulting from the harmonisation of named entities automatically recognised by five different tools, namely, Whatizit\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html}, Peregrine\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html}, GeNO\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html}, MetaMap\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html} and I2E\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html}. Aside from chemical/drug names, protein/gene, disease and species names were also tagged by these tools in the 174,999 MEDLINE abstracts comprising the corpus. The resulting automatically generated annotations were then harmonised employing a voting scheme based on some heuristics. For example, an annotation is accepted only if at least two tools are in agreement, i.e., the semantic types assigned are the same, and the boundaries given by one are subsumed if not matched by those from the other tool. Around half a million named entity annotations for each semantic category are contained in the resulting harmonised corpus which is publicly available\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html}.

**Metabolites Corpus.** Metabolites, small molecules produced during reactions catalysed by enzymes,\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html} form a subset of chemical compounds. As part of a study on yeast metabolic pathway reconstruction, a corpus of 296 MEDLINE abstracts containing metabolite names was developed, which we will refer to as the Metabolites corpus.\footnote{http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/resources.html} Two annotators, both domain experts, were asked to annotate names of metabolites which appear as participants in metabolic pathways. The annotations from the two curators were harmonised to form the gold standard data, resulting in a total of 1,853
annotated metabolite names. Released in MEDLINE XML format, the Metabolites corpus is available for public download\textsuperscript{14}.

**HANAPIN Coreference Corpus.** We ourselves developed our own corpus of documents from the chemistry domain, as a resource for the coreference resolution task. Called HANAPIN, this corpus consists of 20 randomly selected full papers from the *Marine Drugs* journal. Three graduate students, all with degrees in chemistry, were asked to annotate names of chemical entities (as well as other entity types, as described in Section 3.2). They were asked to mark-up any expression by which a chemical entity can be referred to (e.g., systematic/trivial/group name, abbreviation, molecular formula). After harmonisation of the annotations from the three curators, a total of 1,360 chemical mentions were retained in the corpus. It will be made publicly available upon completion of this research.

2.1.2.2 Drug annotations

**DDI Corpus.** Developed as a resource for the SemEval 2013 DDI Extraction track, the Drug-Drug Interaction (DDI) corpus\textsuperscript{85} contains annotations for drug names as well as the interactions between them. It is an extension of the corpus used in the previous DDI Extraction challenge held in 2011 which had 579 textual descriptions from the DrugBank database\textsuperscript{65}. The current version consists of a total of 792 such descriptions, as well as an additional 233 MEDLINE abstracts. Four drug subtypes were defined in the annotation scheme: DRUG for generic names, BRAND for brand names, GROUP for the general group (e.g., antibiotic) and DRUG_N for substances not approved for human use (e.g., pesticide). The corpus was pre-annotated with the use of the MetaMap Transfer (MMTx) tool which is capable of finding within text the locations of concepts contained in the UMLS Metathesaurus\textsuperscript{86}. A total of 18,502 drug names

\textsuperscript{14}http://www.nactem.ac.uk/metabolite-corpus
were annotated in the corpus, with DRUG (11,646) and GROUP (4,225) being the most frequent. Two pharmacists were tasked to perform manual validation of the automatically generated annotations by changing the assigned category when necessary; the boundaries or locations given by MMTx were retained. The corpus has been released publicly.\footnote{http://labda.inf.uc3m.es/doku.php?id=en:labda_ddicorpus}

**ADE Corpus.** The Adverse Drug Effect (ADE) corpus \footnote{https://sites.google.com/site/adecorpus} is a data set consisting of 2,972 MEDLINE medical case reports on topics relevant to drug adverse events. Three concept types were defined in the annotation schema, namely, drug, adverse effect and dosage. The drug category captures all types of drug names (i.e., systematic, trivial, brand, abbreviation) which are mentioned in a therapeutic context. Adverse effects cover diseases or disorders described as reactions to drugs whilst dosage corresponds to characteristics of the drug intake, e.g., quantity and frequency. As a preliminary annotation task, the three annotators were asked to annotate a seed corpus consisting of 100 documents which do not overlap with any of those in the main corpus. After refinement of the annotation guidelines based on insights gained from the preliminary task, the annotators proceeded with the annotation of the main corpus. All tasks were performed with the use of the Knowtator annotation tool \footnote{[88]} . A harmonisation phase was then introduced to reconcile any disparate annotations. The final corpus contains 5,063 annotations for drugs, 5,776 for adverse effects and 231 for dosage. It has been made available for public access.\footnote{https://sites.google.com/site/adecorpus}

**PK Corpus.** The Pharmacokinetics (PK) corpus \footnote{[89]} consists of four subsets corresponding to studies on clinical pharmacokinetics (56 abstracts), clinical pharmacogenetics (57 abstracts), in vivo drug-drug interactions (218 abstracts) and in vitro drug-drug interactions (210 abstracts) . As part of the multi-level annotation scheme, drug
and enzyme names were manually annotated by three curators, aside from PK parameters, numbers, mechanisms and change. The drug category corresponds to names which can be found in the DrugBank database, as well as names of drug metabolites. The enzyme type covers a limited set of enzymes, i.e., the cytochrome P450 (CYP450) family. A total of 8,633 drugs and 3,801 CYP450 enzymes were annotated in the corpus, which is publicly available\(^1\)

**PK DDI Corpus.** Consisting of 64 US Food and Drug Administration-approved drug package labels (also known as package inserts), the Pharmacokinetic Drug-Drug Interaction (PK DDI) corpus \[90\] contains annotations for drugs acting either as precipitants or objects in drug-drug interactions. Each drug is categorised into any of active ingredient, drug product or metabolite. After the retrieval of package inserts from the DailyMed website\(^2\), a pre-annotation stage was performed. For this, the dictionary-based NCBO Annotator tool \[91\] was utilised to find concepts in RxNorm \[92\]. Using the Knowtator annotation tool, two domain experts validated these annotations and created new ones where necessary. After a review and correction phase, full consensus was reached on the annotations. The final corpus, which has been made publicly available\(^3\) contains annotations for 3,351 active ingredients, 234 drug products and 201 metabolites.

**EU-ADR Corpus.** The EU-ADR corpus \[93\], consisting of 300 MEDLINE abstracts, contains named entity annotations for drugs as well as targets and diseases. Annotators were presented with abstracts containing entities and relationships which were automatically pre-annotated by the dictionary-based Peregrine concept recognition system \[80\]. With a web-based annotation tool\(^4\) at their disposal, the annotators validated

---

\(^1\)http://rweb.compbio.iupui.edu/corpus/downloads.html
\(^2\)http://dailymed.nlm.nih.gov
\(^3\)http://dbmi-icode-01.dbmi.pitt.edu/dikb-evidence/package-insert-DDI-NLP-corpus.html
\(^4\)http://euadr.erasmusmc.nl/sda/annotate.py
the automatically generated annotations and created new ones where necessary. The corpus is publicly available.21

2.1.2.3 Comparison

Table 2.2 summarises the details of each corpus reviewed. The following discussion highlights the most notable corpora according to some of the dimensions presented in the table.

The types of documents comprising these chemical corpora range from documents in which the use of natural language hardly conforms with any convention or form (i.e., MEDLINE abstracts and case reports), to relatively more structured texts (i.e., package inserts and DrugBank textual descriptions), to lengthy documents such as patents and full articles. Scientific abstracts are the most popular choice, perhaps due to the greater ease with which they can be obtained, compared to the other document types. Access to full articles, for example, is highly restricted by copyright, whilst sources of specialised documents such as patents and package inserts are not very common. Since this research focusses on information extraction from scientific literature, our interest is in abstracts or full-text articles. They pose more challenges since their authors tend to use natural language more freely, i.e., without following any structure or adhering to any convention. The three largest corpora consisting of abstracts are CALBC, CHEMDNER and SCAI. To date, the SciBorg and HANAPIN corpora are the only chemical document collections consisting of full articles.

We observe that the use of automatic taggers has been integrated in the annotation pipelines of some of the more recently released corpora. CALBC was annotated solely by five automatic tools, making it a purely silver-standard corpus. In contrast, the DDI, PK DDI and EU-ADR corpora underwent a manual correction phase during which human curators validated the automatically generated annotations. The inter-annotator

21http://euadr.erasmusmc.nl/sda/euadr_corpus.tgz
agreement (IAA) on these three corpora was satisfactory. This implies that the development of a corpus can be accelerated without compromising quality by automatic pre-annotation succeeded by a manual correction stage.

It is noticeable that more corpora, especially the recently released ones, have the annotations stored in a stand-off format (i.e., separate from the text). This way, the actual texts are left unmodified, unlike in the case of corpora with in-line annotations which are embedded within the text. Corpora with in-line annotations impose on developers the problem of having to reconstruct the original text, hence requiring more pre-processing tasks such as removal of XML tags and computation of locations (i.e., character offsets) where necessary. To date, there is no one recognised standard format in which named entity annotations are stored. This emphasises the need for tools or solutions to address the issue of interoperability, i.e., (de)serialisation from/into different formats.

All corpora consisting of MEDLINE abstracts have been made publicly available. The PK DDI and Patents corpora, containing package inserts and patent application documents, respectively, are also freely downloadable. Due to copyright restrictions on full-text articles, however, SciBorg is available only upon request.
Table 2.2: Comparison of Corpora with Chemical Named Entity Annotations ($F_1$ defined in Section 2.3)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Document Type/Size</th>
<th>Chemical Subcategories</th>
<th>Auto. Tagger(s)</th>
<th>Annotation Tool</th>
<th>Encoding/Format</th>
<th># Chem. Mentions</th>
<th>IAA</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SciBorg</td>
<td>42 full articles</td>
<td>chemical molecule</td>
<td>N/A</td>
<td>in-house software</td>
<td>in-line XML</td>
<td>7,297</td>
<td>56-93% $F_1$</td>
<td>upon request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>chemical adjective</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>chemical prefix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>chemical reaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>enzyme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCAI</td>
<td>100 abstracts (pilot), 463 abstracts (training), 1K abstracts (test)</td>
<td>IUPAC partial IUPAC abbreviation sum formula trivial family</td>
<td>N/A</td>
<td>undisclosed</td>
<td>BIO</td>
<td>3,712</td>
<td>94% $F_1$</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patents</td>
<td>40 patent documents</td>
<td>N/A</td>
<td>N/A</td>
<td>undisclosed</td>
<td>undisclosed</td>
<td>11,162</td>
<td>undisclosed</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHEMDNER</td>
<td>3.5K abstracts (training), 3.5K abstracts (dev.), 3K (test)</td>
<td>systematic trivial family abbreviation formula identifier multiple</td>
<td>N/A</td>
<td>AnnotateIt</td>
<td>stand-off DSV</td>
<td>29,478</td>
<td>91% $F_1$</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALBC</td>
<td>174,999 abstracts</td>
<td>N/A</td>
<td>Whatizit</td>
<td>N/A</td>
<td>in-line XML</td>
<td>~0.5M</td>
<td>N/A</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Peregrine GeNO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MetaMap I2E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Comparison of Corpora with Chemical Named Entity Annotations (Continued from previous page)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Document Type/Size</th>
<th>Chemical Subcategories</th>
<th>Auto. Tagger(s)</th>
<th>Annotation Tool</th>
<th>Encoding/Format</th>
<th># Chem. Mentions</th>
<th>IAA</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metabolites</td>
<td>296 abstracts</td>
<td>N/A</td>
<td>in-house software</td>
<td>in-line XML</td>
<td>1,853</td>
<td>78.35-88.49%</td>
<td>public</td>
<td></td>
</tr>
<tr>
<td>HANAPIN</td>
<td>20 full papers</td>
<td>N/A</td>
<td>XConc Suite</td>
<td>in-line XML</td>
<td>1,360</td>
<td>76.43-83.77%</td>
<td>to be released</td>
<td></td>
</tr>
<tr>
<td>DDI</td>
<td>792 Drug-Bank texts, 233 abstracts</td>
<td>drug brand group other uses</td>
<td>MMTx</td>
<td>stand-off XML</td>
<td>18,502</td>
<td>79.62-91.04%</td>
<td>public</td>
<td></td>
</tr>
<tr>
<td>ADE</td>
<td>2,972 case reports</td>
<td>N/A</td>
<td>Knowtator</td>
<td>stand-off DSV</td>
<td>5,063</td>
<td>75-80%</td>
<td>public</td>
<td></td>
</tr>
<tr>
<td>PK</td>
<td>541 abstracts</td>
<td>N/A</td>
<td>undisclosed</td>
<td>in-line XML</td>
<td>8,633</td>
<td>95.30%</td>
<td>public</td>
<td></td>
</tr>
<tr>
<td>PK DDI</td>
<td>64 package inserts</td>
<td>active ingredient drug product metabolite</td>
<td>NCBO</td>
<td>stand-off DSV</td>
<td>3,786</td>
<td>60.8-99.50%</td>
<td>public</td>
<td></td>
</tr>
<tr>
<td>EU-ADR</td>
<td>300 abstracts</td>
<td>N/A</td>
<td>Peregrine</td>
<td>stand-off DSV</td>
<td>1,753</td>
<td>71.2-77.5%</td>
<td>public</td>
<td></td>
</tr>
</tbody>
</table>
2.1.3 Approaches to chemical named entity recognition

Methods proposed for the task of recognising chemical and drug names are reviewed in this section. They can be grouped into three categories: heuristics-based, machine learning-based and hybrid approaches. For each proposed technique, the main ideas and contributions are summarised.

2.1.3.1 Heuristics-based approaches: Dictionaries and Rules

Early work on the recognition of chemical entities employed approaches which are based on dictionaries and hand-crafted rules. Kemp and Lynch [94] designed an algorithm for isolating chemical substance names from English patents. Three different dictionaries were compiled: a stopword list, a stopstring list and a 23,000-token dictionary consisting of fragments from chemical names found in databases. The stopword list is comprised of words which are definitely non-chemical whilst the stopstring list contains tokens which would exclude words containing them from being candidate chemical names (e.g., *phobic*). In recognising chemical names in text, rules for suffix matching were first applied to remove plurals and non-nouns which are, based on their intuition, definitely non-chemical names. Thereafter, the remaining words were processed against each of the stopword and stopstring lists. Those matching either were discarded whilst the remaining ones were further processed against the dictionary of tokens.

Similar to this is one of the approaches investigated by Wilbur et al. [95] who developed tools for identifying chemical names in text for phrase extraction and indexing purposes. Using the systematic chemical tokens in the Registry File Basic Name Segment Dictionary [96], the system finds the longest leftmost identical segment and performs this matching on the entire term from left to right. The idea is to divide the term into different tokens found in the token dictionary. A term qualifies as a chemical name
only if it can be segmented into known chemical segment morphemes. To recognise generic and brand names, the token dictionary was extended with additional tokens. Furthermore, the authors applied regular expressions to match numerical expressions pertaining to locants (e.g., $3',4'$ in $3', 4'$-dimethoxyflavone), dosage and measurement. Each term, after segmentation, was assigned a score based on provenance (the number of known chemical segments), cohesiveness (the maximum number of contiguous segments) and coverage (the number of classifiable segments). To evaluate their system, development and test sets were built, consisting of strings from the UMLS Metathesaurus which were categorised into chemical and non-chemical entities.

Narayanaswamy et al. [97] developed a purely rule-based biological named entity recogniser that handles both chemical and protein names. Two types of terms were extracted from text: core terms (c-terms) and functional terms (f-terms). General core terms are characterised by surface features such as capitalisation and the inclusion of numeric or special symbols. However, to differentiate between protein and chemical names, more specific surface features such as the presence of chemical roots or suffixes were additionally considered. In contrast, functional terms are domain-specific terms (e.g. steroid in tertbutyldimethyl silyoxyandrost-4-ene steroid) which were used to distinguish between protein and chemical names. In handling names consisting of multiple terms, rules for extension and concatenation were applied. The method was evaluated against a manually annotated corpus of 55 MEDLINE abstracts on the topic of acetylation.

2.1.3.2 Machine learning-based approaches

Wilbur et al. [95] who have initially used a dictionary-based technique for recognising chemical names, also explored a machine learning approach to improve their results. They implemented a naïve Bayes classifier based on character $n$-gram features in two variants: one where attributes of both positive and negative samples are
of equal weights (which they called the TOTAL method) and another where only the attributes of the positive samples are given weights (called POS method). The TOTAL method achieved the best performance on their test set, beating both the POS and dictionary-based methods. Although their technique was shown to perform well in the recognition of names contained in lists of terms from the UMLS Metathesaurus, it was not evaluated on actual documents containing text.

Friedrich et. al. [98] experimented with different machine learning methods such as support vector machines (SVMs), maximum entropy Markov model (MEMM), naïve Bayes and conditional random fields (CRF) to recognise systematic or IUPAC names. The methods were developed using the version of the GENIA corpus which was subdivided into benchmark data sets in the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004) challenge. In these data sets, the protein names were replaced by the authors with IUPAC names obtained from publicly available chemical databases. For all techniques, the same set of morphological (e.g., capitalisation), lexical (e.g., surface form), statistical (e.g., frequency of the article the) and dictionary (e.g., matches in certain dictionaries) features was used. The CRF model produced by learning on the training set was evaluated on the test set.

Klinger et al. [99] extended this work when they developed a classifier for IUPAC and IUPAC-like names also using conditional random fields. In addition to morphological and lexical features, dictionary features that check for matching IUPAC suffixes and prefixes from PubChem [61] were extracted and included in the feature set for training the classifier. They used as initial training data the corpus from the BioCreative II challenge with protein names replaced with IUPAC names from PubChem, following the methods of Friedrich et al. [98]. Evaluating the classifier trained on this tweaked data set, however, gave very poor results on the SCAI test corpus (which was developed by the authors themselves, described in Section 2.1.2). This poor performance was attributed to two problems: first, only correct and complete IUPAC names
2.1. LITERATURE REVIEW

occur in the tweaked data set which was used for training when in reality, fragments are very common in real texts; and second, the tweaked training set does not contain any negative (non-IUPAC or non-IUPAC-like) examples. After retraining the CRF model on the manually annotated SCAI training corpus, and identifying the best feature set and learning parameters by experimentation, the authors obtained significantly better performance on the test set.

In developing MetaboliNER, a named entity recogniser for metabolites, Nobata et al. also employed a conditional random fields model trained on features such as word n-grams, orthography, part-of-speech tags and dictionary matches against ChEBI and HMDB. MetaboliNER was evaluated using ten-fold cross validation on the Metabolites corpus which we have previously reviewed.

Wren [100] developed a machine learning-based chemical name recogniser using first-order Markov models based on two-character transition frequencies. Two models, one for chemical terms and another for non-chemical terms, were implemented. Given a sequence of characters \( S \), for each model \( M \), the product of each transition from one character to another (i.e., state transition) was computed, corresponding to the probability of observing sequence \( S \) within model \( M \). The confidence score was then calculated as the \( \log_{10} \) ratio between the probabilities. The models were trained on primary names and synonyms from the first half of the ChemIDplus database [64], forming a total of 194,000 entries of chemical terms, and on non-scientific literature (e.g. Charles Darwin’s Descent of Man and Origin of Species, and Leo Tolstoy’s War and Peace). To evaluate its scalability, the algorithm was evaluated on 13.1 million MEDLINE records containing approximately 7.4 million abstracts; an overall precision of 82.7% was achieved.
2.1.3.3 Hybrid approaches

Some proposed approaches to chemical named entity recognition are based on a combination of different methods.

OSCAR 3, developed by Corbett et al. [101], is a publicly available tool for the automatic annotation of chemical entities in journal articles. Its main component is the chemical name recogniser, implemented as a pipeline of different approaches and cascading classifiers. Internal dictionaries were formed by extracting words from the ChEBI database [69] and from a Linux system’s standard English word list. A pre-classifier based on Markov models was trained on tokens from these dictionaries and other manually annotated data. Four-gram character models were used, refined by a modified Kneser-Ney smoothing algorithm [102]. Given an input token, the pre-classifier performs character \( n \)-gram analysis in which a score is calculated based on the probability of the input token pertaining to a chemical/non-chemical entity. Thereafter, maximum entropy Markov models trained on the SciBorg corpus using different features detected potential named entities based on the conditional probability of tag sequences. Some filtering rules for removing spurious candidates were then applied. A unique trait of OSCAR 3 is its capability to disambiguate terms into five different types (corresponding to those introduced in the annotation scheme for the SciBorg corpus), unlike other chemical recognisers which only distinguish between chemical and general non-chemical terms, or between chemical names and protein names. To realise this capability, a re-scoring system was implemented by the construction of a maximum entropy classifier for each type. The entire system was tested both on PubMed abstracts and the SciBorg corpus of full chemistry articles. OSCAR 3 was released as a server intended to run locally on a user’s machine. A refactored version of it, one that utilised a decoupled tokeniser, allowed its integration into configurable text mining workflows and obtained a 2\% increase in \( F_1 \)-score on a SciBorg subset of 14
papers [103]. In a more current release, OSCAR 4 [104], a more flexible and modular API for invoking the core chemical entity recogniser, was introduced.

ChemSpot [105] employs both a dictionary matcher and a conditional random fields model to recognise chemical named entities. This combination is based on the authors’ observation that brand names do not follow any nomenclature nor pattern, whereas systematic names conform with the IUPAC nomenclature albeit susceptible to morphological variations. Following this intuition, the authors employed a dictionary-matching method for capturing brand names, and a classifier for recognising systematic names. For the former, the dictionary-matching engine of LINNAEUS [106] was used, with the Joint Chemical Dictionary (Jochem) [73] as its lexicon. In contrast, the classifier was realised with a conditional random fields model which was trained on the SCAI training corpus and queried using the application programming interface (API) of BANNER [5]. Features used by the model include morphology, size-two prefixes and suffixes and bag of words. Post-processing rules were implemented for merging annotations from the two distinct recognisers. Based on the comparative evaluation results reported by the ChemSpot developers [105], the tool outperforms OSCAR 4 by a margin of 10.8 percentage points on the SCAI pilot corpus.

2.1.3.4 Comparison

Presented in Table 2.3 are details of some of the various reported approaches to chemical named entity recognition, categorised into heuristics-based, machine learning-based and hybrid types. As can be observed in the table, methods based on heuristics require the construction of domain-specific resources. In the case of dictionary-based approaches, term lists (e.g., stopwords, segments, functional terms) are crucial, whilst rule-based ones require manually constructed rules or patterns (e.g., regular expressions). A primary problem with the use of purely dictionary- and/or rule-based approaches arises from the fact that not all names of chemical entities can possibly be
covered by one or even a combination of resources. Chemical entities are mentioned in free text in numerous ways (e.g., systematic, semi-systematic, trivial, brand and family names, abbreviations), and to build a resource that would cover all of these names or even just the patterns for them is nearly impossible. This is true especially in some domains of chemistry (i.e., natural products chemistry) where the interest is in novel compounds whose names have not been previously curated in any resource. Furthermore, even if these compounds are assigned names which conform with the IUPAC nomenclature, different interpretations of the nomenclature give rise to morphological variations. Hence, it is infeasible to create dictionaries or rules which are exhaustive enough to cover all cases.

Another weakness, particularly of dictionary-based approaches, is the inability to handle ambiguous surface forms. As discussed in the beginning of this chapter, the token lead, for example, might be recognised as the chemical element even if in text it was used as a verb or adjective in the leadership sense of the word. Because of such cases, context surrounding a given token must be taken into account.

In this respect, machine learning techniques are more robust as they can learn hidden properties of named entities (e.g., surrounding context), without requiring the construction of exhaustive hand-crafted pattern-matching rules. The drawback, however, is the need for annotated corpora from which the technique can learn the features that would characterise the entity types of interest. With machine learning-based approaches, the feature set employed for model training and prediction is key. The most commonly used types are character n-grams, lexical (e.g., surface form), morphological (e.g., capitalisation) and dictionary features. It is also worth noting that many of the machine learning-based approaches made use of the conditional random fields algorithm.

Other approaches are based on the hybrid of different algorithms, combining the
strengths of various methods. ChemSpot, for example, takes advantage of the robustness of CRF models to handle morphological variations in IUPAC names but leverages a dictionary to capture name types in which patterns can be hardly observed.

Unfortunately, the approaches in Table 2.3 are not comparable in terms of their performance as they were evaluated on different data sets or corpora; we tabulated the performance only for the reader’s reference. It is also not possible to benchmark all of them against one corpus since only OSCAR 3/4 and ChemSpot are publicly available. In Section 2.3, however, the performance of OSCAR 3/4 and ChemSpot will be presented and compared with our own proposed method on several corpora.
### Table 2.3: Comparison of Approaches to Chemical Named Entity Recognition

<table>
<thead>
<tr>
<th>Approach</th>
<th>Proponents or System</th>
<th>Key Ideas and Resources</th>
<th>Eval. Corpus</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristics</td>
<td>Kemp &amp; Lynch</td>
<td>matching against token dictionary, stopword list and stopstring list</td>
<td>patents</td>
<td>undisclosed</td>
<td>97.4%</td>
<td>undisclosed</td>
</tr>
<tr>
<td></td>
<td>Wilbur et al.</td>
<td>chemical segment dictionary, regular expression matching, scoring</td>
<td>UMLS strings</td>
<td>80%</td>
<td>84%</td>
<td>81.95%</td>
</tr>
<tr>
<td></td>
<td>Narayanaswamy et al.</td>
<td>rules, core terms, functional terms, help terms</td>
<td>55 abstracts</td>
<td>93.15%</td>
<td>86.08%</td>
<td>89.48%</td>
</tr>
<tr>
<td>ML</td>
<td>Wilbur et al.</td>
<td>naïve Bayes, character n-grams as features</td>
<td>UMLS strings</td>
<td>95%</td>
<td>97%</td>
<td>96%</td>
</tr>
<tr>
<td></td>
<td>Friedrich et al.</td>
<td>CRFs, morphological, statistical, lexical, dictionary features</td>
<td>tweaked GENIA</td>
<td>97.22%</td>
<td>97.27%</td>
<td>97.25%</td>
</tr>
<tr>
<td></td>
<td>Klinger et al.</td>
<td>CRFs, morphological, lexical features, IUPAC infixes as dictionary features</td>
<td>SCAI test set</td>
<td>86.5%</td>
<td>84.8%</td>
<td>85.6%</td>
</tr>
<tr>
<td></td>
<td>Nobata et al.</td>
<td>CRFs, orthography, lexical features, ChEBI and HMDB dictionary features</td>
<td>Metabolites</td>
<td>83.02%</td>
<td>74.42%</td>
<td>78.49%</td>
</tr>
<tr>
<td></td>
<td>Wren</td>
<td>first-order Markov models, ChemID</td>
<td>7.4M abstracts</td>
<td>82.7%</td>
<td>undisclosed</td>
<td>undisclosed</td>
</tr>
<tr>
<td>Hybrid</td>
<td>OSCAR 3/4</td>
<td>Markov token pre-classifier, maximum entropy Markov models</td>
<td>SciBorg</td>
<td>78.7%</td>
<td>82.9%</td>
<td>80.7%</td>
</tr>
<tr>
<td></td>
<td>ChemSpot</td>
<td>LINNAEUS matching against Jochem, CRFs using BANNER</td>
<td>SCAI pilot set</td>
<td>67.3%</td>
<td>68.9%</td>
<td>68.1%</td>
</tr>
</tbody>
</table>


2.2. Methodology

We cast the problem of named entity recognition as a sequence labelling task, i.e., the automatic assignment of labels to a sequence of items. In this case, the sequence of items corresponds to the ordered tokens in an input sentence. Example 2.4 shows the same sentence in the example at the beginning of this chapter, but this time treated as a space-delimited sequence of tokens with the desired labels.

Example 2.4. United\textsubscript{B−Org} Nations\textsubscript{I−Org} humanitarian\textsubscript{O} chief\textsubscript{O} Valerie\textsubscript{B−Per} Amos\textsubscript{I−Per} \textsubscript{O} visiting\textsubscript{O} the\textsubscript{O} devastated\textsubscript{O} city\textsubscript{O} of\textsubscript{O} Tacloban\textsubscript{B−GPEntity} said\textsubscript{O} the\textsubscript{O} situation\textsubscript{O} was\textsubscript{O} desperate\textsubscript{O} with\textsubscript{O} residents\textsubscript{O} left\textsubscript{O} without\textsubscript{O} food\textsubscript{O} or\textsubscript{O} fresh\textsubscript{O} water\textsubscript{O} for\textsubscript{O} five\textsubscript{O} days\textsubscript{O}.

The set of possible labels is defined by a chosen representation or encoding. The inside-outside (IO) encoding, for instance, differentiates only between tokens within named entities (inside or I) and those which are not (outside or O). Another option is the much more complex begin-middle-end-word-outside (BMEWO) encoding which additionally distinguishes amongst tokens at the beginning (B), end (E) and middle (M) of a named entity, and tokens which correspond to single-word named entities (W). We chose the more commonly used begin-inside-outside (BIO) encoding, which is also what has been applied to Example 2.4.

The machine learning-based conditional random fields (CRF) algorithm has been shown to perform well in sequence labelling tasks, particularly in biomedical named entity recognition. It has also been employed in purely machine learning-based chemical named entity recognition \cite{98, 99} as well as in ChemSpot \cite{105} whose engine is based on both a dictionary matcher and a statistical model. In this study, we propose the combination of machine learning and rules in attempting to solve the problem, specifically:

• training of a CRF model on an enriched, chemistry-specific feature set
• token relabelling and abbreviation detection as post-processing steps

2.2.1 Conditional random fields

The conditional random fields algorithm [107] finds the most probable label sequence \( y \) given an observation sequence \( x \) according to Equation 2.1.

\[
y = \text{argmax}_y \ p_\lambda(y|x) \tag{2.1}
\]

where \( x \) consists of the sequence of tokens from input text. The probability \( p_\lambda(y|x) \) is computed by Equation 2.2.

\[
p_\lambda(y|x) = \frac{1}{Z_x} \cdot \exp\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y_{i-1}, y_i, x, i)\right) \tag{2.2}
\]

Feature functions define the result of this equation. Each feature function \( f_j(y_{i-1}, y_i, x, i) \) is multiplied by a learned weight \( \lambda_j \) specific to that feature. The summation of all \( m \) weighted feature functions is obtained, which in turn, is accumulated over all \( n \) items in the sequence. The exponential function of this accumulated value is then multiplied by the reciprocal of \( Z_x \) which is a normalisation factor for all state sequences [4]. Details of the features employed in this study are discussed in more detail in the next section.

The input to the CRF algorithm are sequences of tokens from text. This type of information is provided by a pre-processing pipeline that takes raw text as input. We describe our specific pre-processing pipeline in Section 2.3.

Instead of developing our own implementation of the conditional random fields algorithm, we leveraged an already existing package specifically built for biomedical NER called NERsuite [108].
2.2. METHODOLOGY

2.2.2 Feature set

In this section, we discuss in detail the enriched feature set employed in training (and querying) the CRF model. The features are grouped into two general categories to distinguish between those which are typically used in biomedical NER which come with NERsuite by default, and our proposed features.

2.2.2.1 Default features

We first describe what can be considered internal features or those which describe only the active token (i.e., the token currently in consideration) and none of the context surrounding it.

Character $n$-grams. A token’s character $n$-grams corresponds to the set of all possible size $n$ combinations of consecutive characters. The token cocaine, for example, will have co, oc, ca, ai, in and ne as its 2-grams and coc, oca, cai, ain and ine as its 3-grams. Size two, three and four $n$-grams are being extracted by NERsuite as features.

Orthography. Features capturing the symbol-level composition of a token are known as orthographic features. NERsuite extracts a wide range of such features, mostly adopted from the work on biomedical named entity recognition of Lee et al. \[109\]. These are enumerated in Table 2.4 together with examples.

In contrast to internal features, the following word $n$-gram features capture characteristics of the context surrounding the active token by taking into account information from its neighbouring tokens. A maximum distance $d$ relative to the current active token at index $i$ of the sequence is initially set. Unigrams and bigrams within the window defined by $d$ are extracted as follows:

- Unigrams: $\{w_{i-d}\}, \{w_{i-d+1}\}, \{w_{i-d+2}\}, \ldots, \{w_{i+d-2}\}, \{w_{i+d-1}\}, \{w_{i+d}\}$
### Table 2.4: Orthographic features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial letter is in uppercase</td>
<td>Boc-L-leucine</td>
</tr>
<tr>
<td>Contains only digits</td>
<td>206553</td>
</tr>
<tr>
<td>Contains digits</td>
<td>5-HTP</td>
</tr>
<tr>
<td>Contains only alphanumeric characters</td>
<td>HClO4</td>
</tr>
<tr>
<td>Contains only uppercase letters and digits</td>
<td>AFB1</td>
</tr>
<tr>
<td>Contains only uppercase letters</td>
<td>NO</td>
</tr>
<tr>
<td>Does not contain any lowercase letters</td>
<td>SKF81297</td>
</tr>
<tr>
<td>Contains non-initial uppercase letters</td>
<td>PbS</td>
</tr>
<tr>
<td>Contains two consecutive uppercase letters</td>
<td>PAHs</td>
</tr>
<tr>
<td>Has a Greek letter name as a substring</td>
<td>alpha-ketoacid</td>
</tr>
<tr>
<td>Contains a comma</td>
<td>3,14-dibromo</td>
</tr>
<tr>
<td>Contains a full stop</td>
<td>In(0.2)Ga(0.8)As</td>
</tr>
<tr>
<td>Contains a hyphen</td>
<td>HP-β-CD</td>
</tr>
<tr>
<td>Contains a forward slash</td>
<td>(E/Z)-Goniothalamin</td>
</tr>
<tr>
<td>Contains an opening square bracket</td>
<td>[(14)C]pazopanib</td>
</tr>
<tr>
<td>Contains a closing square bracket</td>
<td>pyrido[3,2-d]pyrimidines</td>
</tr>
<tr>
<td>Contains an opening parenthesis</td>
<td>I3 (-)</td>
</tr>
<tr>
<td>Contains a closing parenthesis</td>
<td>Fe(C10 H15 )2</td>
</tr>
<tr>
<td>Contains a semi-colon</td>
<td>R=Me, Et; X=O, S;</td>
</tr>
<tr>
<td>Contains a percentage symbol</td>
<td>85%</td>
</tr>
<tr>
<td>Contains an apostrophe</td>
<td>5-methyl-2′-deoxycytidine</td>
</tr>
</tbody>
</table>

- **Bigrams:** \{w_{i-d}, w_{i-d+1}\}, \{w_{i-d+1}, w_{i-d+2}\}, \ldots, \{w_{i+d-2}, w_{i+d-1}\}, \{w_{i+d-1}, w_{i+d}\}

**Token n-grams.** Using a maximum distance of size two from the active token, NERsuite extracts unigrams and bigrams based on the surface forms. From the sequence of tokens in Table 2.5 with GSK221149A as the active token, the following token n-grams will be extracted: \{found\}, \{that\}, \{GSK221149A\}, \{affects\}, \{gestation\} as unigrams, and \{found, that\}, \{that, GSK221149A\}, \{GSK221149A, affects\}, \{affects, gestation\} as bigrams.

Additionally, NERsuite extracts unigrams and bigrams of normalised surface forms, where numbers have been converted to ‘0’ and compressed if appearing consecutively. The unigrams and bigrams formed from the normalised surface forms in the same example include \{found\}, \{that\}, \{GSK0A\}, \{affects\}, \{gestation\}, and \{found, that\},
2.2. METHODOLOGY

Table 2.5: Example of a sentence tokenised and labelled with parts-of-speech tags.

<table>
<thead>
<tr>
<th>Surface form</th>
<th>Lemma</th>
<th>Part-of-speech tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>We found that GSK221149A affects gestation in rats</td>
<td>We find that GSK221149A affect gestation in rat</td>
<td>PRP VBD IN NN VBG IN NN NNS</td>
</tr>
</tbody>
</table>

\{that, GSK0A\}, \{GSK0A, affects\}, \{affects, gestation\}.

**Lemma n-grams.** In the same manner in which it collects surface form n-grams, NERsuite extracts unigrams and bigrams of tokens’ lemmatised forms. Given the same example in Table 2.5, it will obtain \{find\}, \{that\}, \{GSK221149A\}, \{affect\}, \{gestation\} as unigrams, and \{find, that\}, \{that, GSK221149A\}, \{GSK221149A, affect\}, \{affect, gestation\} as bigrams. Unigrams and bigrams formed from normalised lemmatised forms are also being extracted, e.g., \{GSK0A\} and \{GSK0A, affect\}.

**Part-of-speech (POS) tags.** Similarly, unigrams and bigrams of part-of-speech tags are extracted from within a distance of two from the active token. From the same example, still using GSK221149A as the active token, the unigrams \{VBD\}, \{IN\}, \{NN\}, \{VBZ\}, \{NN\} and bigrams \{VBD, IN\}, \{IN, NN\}, \{NN, VBZ\}, \{VBZ, NN\} will be extracted.

**Lemma and POS tag n-grams.** These n-grams are formed based on the combination of lemmatised forms and part-of-speech tags. Examples of unigrams and bigrams from the same example include \{find:VBD\}, \{that:IN\}, \{GSK221149A:NN\}, \{affect:VBZ\}, \{gestation:NN\}, and \{find:VBD, that:IN\}, \{that:IN, GSK221149A:NN\}, \{GSK221149A:NN, affect:VBZ\}, \{affect:VBZ, gestation:NN\}, respectively.
Table 2.6: Example of a sentence tokenised and labelled with parts-of-speech and chunk tags.

<table>
<thead>
<tr>
<th>Surface form</th>
<th>Lemma</th>
<th>Part-of-speech tag</th>
<th>Chunk tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>It attenuated MPTP</td>
<td>It</td>
<td>PRP</td>
<td>B-NP</td>
</tr>
<tr>
<td>-induced depletion of dopamine</td>
<td>attenuate MPTP -induced depletion of dopamine</td>
<td>VBD NN IN</td>
<td>B-VP B-NP B-PP</td>
</tr>
<tr>
<td>.</td>
<td></td>
<td>.</td>
<td>O</td>
</tr>
</tbody>
</table>

Chunks. Information pertaining to chunks or phrases is also incorporated into the feature set. This includes the chunk tag of the active token and the surface form of the last token in the current chunk. Given the example in Table 2.6, NERsuite will extract as chunk features of the active token *MPTP* the corresponding chunk tag {B-NP} and the surface form {*depletion*}.

2.2.2.2 Proposed features

The following were added to the feature set in order to capture certain characteristics, some of which are intrinsically chemical.

Greek characters. Several chemical names contain Greek characters, e.g., *17beta-oestradiol* and *(S)-α,ε-diaminohexanoic acid*. The former example has the Greek letter name *beta* in the surface form whilst the latter contains the characters *α* and *ε*. Although the default feature set already includes a feature for catching the presence of full Greek letter names (e.g., *beta*) in the surface form, it does not take into account the occurrence of Greek characters (e.g., *α* and *ε*). In order to capture this, we added an additional orthographic feature that checks if an active mention’s surface form contains any of the 24 Greek characters.
Word shape. Quite similar to the normalised surface form employed as a token n-gram feature, a token’s word shape is also a canonical representation of its surface form, albeit a more fine-grained one. Whilst the normalised surface form canonicalises only numerical characters by converting them to ‘0’ and compressing consecutive instances (e.g., from GSK221149A to GSK0A), the word shape feature that we introduced into the feature set additionally accounts for alphabetic characters, punctuation and special characters. Two variants were used: (1) the full word shape, which converts all uppercase letters to ‘A’, lowercase letters to ‘a’, numbers to ‘0’ and everything else (e.g., punctuation and special characters) to ‘_’; and (2) the brief word shape, which is equivalent to the full word shape with its consecutive character types squeezed into one. The name 10-amino-20(S)-camptothecin, for instance, would have 00_aaaaa_00_A__aaaaaaaaaaaa as its full word shape and 0_a_0_A_a as the brief variant.

Dictionaries. The first chemistry-specific feature set that was introduced captures matches between token surface forms and names recorded in chemical lexica and databases. This feature was added based on the intuition that a token whose surface form exists in any domain expert-curated chemical dictionary has a high probability of being a chemical name. As an initial step, five chemical dictionaries were compiled based on the following databases/lexica, taking into account both preferred names and all available synonyms.

- Chemical Entities of Biological Interest (ChEBI) [110]
- DrugBank [65]
- Comparative Toxicogenomics Database (CTD) [32]
- PubChem Compound [61]
- Joint Chemical Dictionary (Jochem) [73]
During the compilation of these dictionaries, entries were normalised by converting all alphabetic characters to lowercase, all numbers to ‘0’ and all punctuation/special characters to ‘_’. The surface forms of tokens were also being converted to a normalised form in the same manner. We then utilised the dictionary matcher bundled with NERsuite to capture the longest possible matches between a sequence of normalised surface forms and entries in the compiled dictionaries. Token sequences are tagged with dictionary matches in the BIO format, examples of which are shown in Table 2.7. Unigrams and bigrams were then extracted for each dictionary, both from the BIO labels as well as from the combination of the BIO labels with the corresponding surface form. With starch as the active token and two as the maximum distance, the following features will be extracted by matching the token sequence in the first column of Table 2.7 against CTD:

- Surface form and dictionary match n-grams:
  - Unigrams: \{on:O\}, \{hydroxyethyl:B\}, \{starch:I\}, \{-:O\}, \{hydroxyethyl:B\}

- Dictionary match n-grams:
  - Unigrams: \{O\}, \{B\}, \{I\}, \{O\}, \{B\}
  - Bigrams: \{O, B\}, \{B, I\}, \{I, O\}, \{O, B\}

**Affixes.** A significant proportion of chemical names, especially those which were formed based on nomenclature, often begin and/or end with any of commonly used chemical prefixes/suffixes. In order to capture this information, lists containing the most popular size-two, -three and -four chemical prefixes and suffixes were compiled, against which the surface form of each token is checked. Table 2.8 shows the result.
of matching the tokens in the first column against our affix lists which are provided in Appendix A.1. Tags resulting from this matching process are incorporated into the feature set.

Table 2.8: Example of a token sequence tagged with matches against our affix lists.

<table>
<thead>
<tr>
<th>Token</th>
<th>Prefixes</th>
<th>Suffixes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>size 2</td>
<td>size 3</td>
</tr>
<tr>
<td>Incubation with diisopropyl fluorophosphate and bis-(4-nitrophenyl) phosphate</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>flou</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
Segment Dictionary which contains 3,307 chemical name segments corresponding to expressions for any of the following: chemical functionality or moiety, ring, multiplicative prefix and common natural product name [96]. Also described in this resource is an algorithm for dividing a full chemical name into its natural segments, each of which is further broken down into its basic segments. The reader is referred to the said document for details of the algorithm.

Based on the suggested algorithm, we extract the number of basic segments in the surface form of each token and employ this as a feature. Provided in Table 2.9 are some examples. It is worth noting that in counting the number of basic segments, the algorithm also includes fragments which were not successfully matched against the dictionary. For instance, in the second example in the table Methylergonovine, only the segments methyl, ergo and novi exist in the dictionary; nevertheless, the unmatched segment ne was still included in the counting of basic segments.

Table 2.9: Examples of chemical names with corresponding basic segments.

<table>
<thead>
<tr>
<th>Token</th>
<th>Basic segments</th>
<th>No. of basic segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-acetoxyactinidine</td>
<td>10, acet, oxy, actin, idine</td>
<td>5</td>
</tr>
<tr>
<td>methylergonovine</td>
<td>methyl, ergo, novi, ne</td>
<td>4</td>
</tr>
<tr>
<td>interleukin-2</td>
<td>interleukin, 2</td>
<td>2</td>
</tr>
</tbody>
</table>

**Chemical symbols.** Lastly, we additionally check tokens which have a maximum length of two against a list of chemical element symbols since such short tokens can be hardly described by features based on chemical segments and affixes. Furthermore, not all of the chemical elements are contained in the five dictionaries we have compiled, especially those whose applications are not relevant to biomedicine. Since they are, nevertheless, still considered chemical named entities, it was deemed necessary to incorporate a boolean attribute into the feature set, signifying a match between a token’s surface form and a chemical element symbol. The complete list of chemical element symbols is provided in Appendix A.2.
2.3 Evaluation and Discussion

Described in this section is our experimental setup for named entity recognition. We divided the problem into two subtasks, i.e., chemical NER and drug NER. The former involves the recognition of names of chemical entities in general, regardless of use or application. In contrast, the latter focusses only on the recognition of chemical entities mentioned within a pharmacological context.

The standard metrics for measuring classification performance, namely, precision, recall and F₁-score, were used in our evaluation. Precision measures the ratio of identified instances which are relevant whilst recall is the ratio of relevant instances which were identified. These two metrics can be computed in terms of the number of true positives (TP), false positives (FP) and false negatives (FN) which are computed between the output of an automatic method (response) and gold standard data (reference). These are defined as follows.

- TP is the number of instances in the reference which were correctly identified in the response.
- FP is the number of instances in the reference which were incorrectly identified in the response.
- FN is the number of instances in the reference which were incorrectly rejected in the response.

Precision and recall can then be computed using Equations 2.3 and 2.4, respectively.

\[
P = \frac{TP}{TP + FP}
\]  
(2.3)
\[ R = \frac{TP}{TP + FN} \]  

(2.4)

The combination of these two metrics is often expressed as the F\(_1\) score (or balanced F-score) which is the harmonic mean of precision and recall (Equation 2.5).

\[ F_1 = 2 \cdot \frac{P \cdot R}{P + R} \]  

(2.5)

In reporting the values of these three metrics over a set of documents, there are two ways by which they are computed, namely, by micro-averaging and macro-averaging. In the former, the individual TP, FP and FN are accumulated over all documents. Micro-averaged precision and recall are computed based on the sums; micro-averaged F\(_1\) is the harmonic mean of the micro-averaged precision and recall. Macro-averaging differs from this in that precision and recall are computed for each document. By taking the average of the precision and recall values over all documents, we obtain the macro-averaged precision and recall, respectively. The macro-averaged F\(_1\) is then calculated by taking the harmonic mean of the two.

It is worth noting that when comparing evaluation results across various datasets (i.e., corpora), micro-averaging is more suitable as it is not sensitive to the number of documents in the dataset.

### 2.3.1 Chemical NER

In evaluating our proposed methodology, we used three of the gold standard corpora reviewed in Section 2.1. The CRF model for chemical NER was developed based on the CHEMDNER training and development sets consisting of a total of 7,000 MEDLINE abstracts. Testing was carried out on the following corpora: the CHEMDNER test set (3,000 MEDLINE abstracts), the SCAI pilot corpus (100 MEDLINE abstracts),
2.3. EVALUATION AND DISCUSSION

as well as the Metabolites corpus. All documents were pre-processed with the pipeline described below.

2.3.1.1 Pre-processing pipeline

Sentence splitting. The LingPipe MEDLINE sentence model [111] was our sentence splitter of choice. Based on a set of heuristics, this sentence model was specifically designed for processing biomedical text. It employs rules which take into account possible stops (e.g., full stop, question mark), impossible penultimates (e.g., personal titles such as Dr.) and impossible starts (e.g., the percent sign). This model was especially configured to allow a word beginning with a lowercase letter to start a sentence as long as an uppercase letter or digit is encountered before any white space character. For example, each of the words p70 and rIFN-alpha-2b may begin a sentence but not the word antibiotic.

Tokenisation. Each sentence is segmented into tokens using OSCAR 4’s tokenisation tool. This tokeniser, having been tuned for chemical texts, can keep intact long chemical names as opposed to other general- or biomedical-domain tokenisers. For instance, given a sentence with 4,9-Diazadodecane-1,12-diamine, the biomedical tokeniser GENIA Tagger [56] will return five tokens, having treated the commas as delimiters. Having a chemical name split into several tokens is not preferable as it leads to loss of information during subsequent feature extraction, especially in the case of features such as number of chemical segments.

Part-of-speech and chunk tagging. Based on a maximum entropy model trained on both general- and biomedical-domain documents, the GENIA tagger has demonstrated robustness in tagging biomedical text with syntactic information [56]. It can take as input already tokenised sentences and provides as output for each token the lemmatised
CHAPTER 2. NAMED ENTITY RECOGNITION

form as well as the part-of-speech and chunk tags.

2.3.1.2 Training and development

Given the gold standard labels in the training corpus combined with information resulting from the pre-processed documents such as token boundaries, lemmata, POS and chunk tags, the NERsuite package extracts all features described in the previous section. We initially trained two CRF models using the CHEMDNER training set of 3,500 MEDLINE abstracts, one with the default features and the other with our enriched feature set. The resulting models were tested on the CHEMDNER development set, also consisting of 3,500 abstracts. Aside from a label in the BIO format, our CRF model provides as output a probability value between 0 and 1 signifying the confidence with which the model has assigned a label to a token. In computing the values of the performance metrics, we utilised the official BioCreative evaluation library\textsuperscript{22} that outputs both macro- and micro-averaged precision, recall and F\textsubscript{1} scores. As this library requires the responses to be encoded in character offset format, the BIO labels were combined with token boundary information to generate the named entity offsets, illustrated in Table 2.10. The first three columns show the token boundary information provided by our pre-processing pipeline, whilst the fourth column contains the BIO labels given by the CRF model. By combining these two types of information, the named entity boundaries (fifth column) are obtained in a straightforward manner.

Presented in Table 2.11, preliminary evaluation results indicate that the model trained on the enriched feature set (Custom NERsuite) outperforms the other (Default NERsuite) in all metrics by the indicated margin. As the models obtain better precision than recall, we attempted to increase the latter by introducing two post-processing steps which aim to recognise instances which were incorrectly rejected by the CRF models. Recalling that the annotations in the CHEMDNER training and development sets are

\textsuperscript{22}http://www.biocreative.org/resources/biocreative-ii5/evaluation-library
2.3. EVALUATION AND DISCUSSION

Table 2.10: Example of BIO-tagged tokens converted into named entity character offsets.

<table>
<thead>
<tr>
<th>Begin offset</th>
<th>End offset</th>
<th>Token</th>
<th>BIO label</th>
<th>Named entity character offsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>The</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>chemopreventive</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>role</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>27</td>
<td>of</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>43</td>
<td>ursodeoxycholic</td>
<td>B</td>
<td>(28-48): [ursodeoxycholic acid]</td>
</tr>
<tr>
<td>44</td>
<td>48</td>
<td>acid</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>51</td>
<td>in</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>64</td>
<td>azoxymethane</td>
<td>B</td>
<td>(52-64): [azoxymethane]</td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>-</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>72</td>
<td>treated</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>77</td>
<td>rats</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.11: Evaluation on CHEMDNER development set of model trained on CHEMDNER training set

<table>
<thead>
<tr>
<th></th>
<th>Macro</th>
<th></th>
<th>Micro</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
</tr>
<tr>
<td>Default NERsuite</td>
<td>86.663</td>
<td>79.007</td>
<td>80.887</td>
<td>88.55</td>
</tr>
<tr>
<td>Custom NERsuite</td>
<td>88.255</td>
<td>81.111</td>
<td>82.863</td>
<td>89.868</td>
</tr>
<tr>
<td>Margin</td>
<td>+1.592</td>
<td>+2.104</td>
<td>+1.976</td>
<td>+1.318</td>
</tr>
</tbody>
</table>

categorised into chemical subtypes, we looked at the distribution of false negatives according to these subtypes (Table 2.12).

Detection of abbreviations. Based on our error analysis, abbreviations account for 30% of the instances most frequently missed by the the CRF model trained on our enriched features. As a post-processing step, our chemical NER iterates through the set of response chemical named entities given by the model for each document. With $i$ as the index of the last token in any named entity $e$, our chemical NER checks if:

- the token at $i + 1$ is the opening parenthesis,
- the token at $i + 3$ is the closing parenthesis, and
Table 2.12: Distribution of instances in the CHEMDNER data set incorrectly rejected (FN) by the Custom NERsuite model

<table>
<thead>
<tr>
<th>Subtype</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbreviation</td>
<td>1,882</td>
<td>30.32%</td>
</tr>
<tr>
<td>Formula</td>
<td>1,291</td>
<td>20.80%</td>
</tr>
<tr>
<td>Family</td>
<td>979</td>
<td>15.77%</td>
</tr>
<tr>
<td>Trivial</td>
<td>926</td>
<td>14.92%</td>
</tr>
<tr>
<td>Systematic</td>
<td>693</td>
<td>11.16%</td>
</tr>
<tr>
<td>Identifier</td>
<td>293</td>
<td>4.72%</td>
</tr>
<tr>
<td>Multiple</td>
<td>118</td>
<td>1.90%</td>
</tr>
<tr>
<td>No class</td>
<td>25</td>
<td>0.40%</td>
</tr>
</tbody>
</table>

- the token at $i + 2$ was assigned the label $O$ by the CRF model.

If all three conditions hold true, the NER verifies that the named entity $e$ is a possible expanded form of the token at $i + 2$. This is done using a simple abbreviation detection algorithm \cite{112} that checks if each character in the short form occurs in the candidate expanded form. If named entity $e$ was confirmed to be an expanded form of token $i + 2$, the label of all instances of the said token is changed from $O$ to $B$. Furthermore, these instances will be added to the set of response chemical named entities for the current document.

**Computation of chemical segment composition.** To alleviate the problem of partial recognition of family, trivial and systematic names (altogether accounting for almost 42% of the false negatives), a score-based token relabelling algorithm measuring chemical segment composition was introduced. As a first step, the probability values given by the CRF model to outside ($O$) tokens were examined. Those tokens whose probabilities were less than a chosen threshold $t_1$ are subjected to a chemical composition analysis step. This thresholding step is based on the intuition that those tokens which were classified as non-chemical tokens by the CRF model with low confidence are
likely to be parts of a chemical name. Leveraging the chemical basic segments resulting from the extraction of features, our chemical NER filters out those segments in a given token which do not exist in the segment dictionary. Chemical segment composition is then calculated as the ratio of the number of characters comprising the remaining basic segments to the total number of characters in the token. If this ratio is greater than a chosen threshold $t_2$, the token was relabelled as part of a chemical name. A few examples of tokens and their corresponding chemical segment composition ratios are provided in Table 2.13.

Table 2.13: Chemical segment composition of some sample tokens

<table>
<thead>
<tr>
<th>token</th>
<th>Chemical basic segments (in dictionary)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>polycalcium</td>
<td>poly, calcium</td>
<td>1.0</td>
</tr>
<tr>
<td>2-methoxyestradiol</td>
<td>meth, oxy, estra, di, ol</td>
<td>0.89</td>
</tr>
<tr>
<td>palytoxin</td>
<td>toxin</td>
<td>0.56</td>
</tr>
</tbody>
</table>

In determining the appropriate thresholds, the performance of the chemical NER on the development set was calculated using different settings of $t_1$ and $t_2$. The first threshold $t_1$ defines how low the CRF model’s confidence in assigning an $\circ$ label to a token should be for it to be considered a probable part of a chemical. Threshold $t_2$ specifies how high the chemical segment composition ratio should be for a token initially labelled as $\circ$ by the CRF model to qualify for reclassification into a chemical name part. For $t_1$, we experimented with values within the range of $[0.91, 0.99]$ in increments of 0.01; for $t_2$, values within the range of $[0.5, 0.9]$ in increments of 0.1 were explored. Our experiments showed that optimal precision and $F_1$ scores are obtained when $t_1 = 0.93$ and $t_2 = 0.9$; for optimal recall, the threshold values $t_1 = 0.96$ and $t_2 = 0.5$ are more suitable.

Although some regular expressions were formulated in an attempt to increase recall for chemical formulas, we have decided to not employ them since their application led to a significant decrease in precision. This implies that not all formula-like token
surface forms actually pertain to chemical entities.

2.3.1.3 Comparison

Having completed the fine-tuning experiments on the CHEMDNER development data set, we trained a final CRF model using the merged CHEMDNER training and development data sets (7,000 MEDLINE abstracts). Different variants of our chemical NER were evaluated on the CHEMDNER test data set of 3,000 abstracts and the SCAI pilot corpus of 100 abstracts. We further compare our results with those of the publicly available OSCAR 4 and ChemSpot by evaluating them on the same datasets. It is worth noting that these two tools are the currently recognised state-of-the-art chemical named entity recognisers.

Presented in Table 2.14 are the micro-averaged evaluation results for OSCAR 4 and ChemSpot alongside the following:

- Default NERsuite: based on a CRF model trained on the default feature set
- Custom NERsuite: based on a CRF model trained on the enriched feature set
- Custom NERsuite+ChemSeg: based on the same CRF model as in Custom NERsuite but with computation of chemical segment composition as a post-processing step; uses $t_1 = 0.93$ and $t_2 = 0.9$ in the chemical composition computation step to optimise precision and F$_1$ score
- Custom NERsuite+Abbr: based on the same CRF model as in Custom NERsuite but with abbreviation detection as a post-processing step
- Custom NERsuite+ChemSeg+Abbr-1: based on the same CRF model as in Custom NERsuite but with both post-processing steps; uses $t_1 = 0.93$ and $t_2 = 0.9$ in the chemical composition computation step to optimise precision and F$_1$ score
2.3. **EVALUATION AND DISCUSSION**

- Custom NERsuite+ChemSeg+Abbr-2: based on the same CRF model as in Custom NERsuite but with both post-processing steps; uses $t_1 = 0.96$ and $t_2 = 0.5$ in the chemical composition computation step to optimise recall.

A consistent pattern can be observed across both datasets, i.e., optimal precision and $F_1$ scores are achieved with Custom NERsuite+Abbr whilst the best recall is obtained with Custom NERsuite+ChemSeg+Abbr-2. Furthermore, Custom NERsuite+Abbr, our best performing chemical NER in terms of $F_1$ score, outperforms OSCAR 4 and ChemSpot by a margin of at least 22 percentage points on the CHEMDNER test data, and at least 5 percentage points on the SCAI pilot corpus.

Having observed that the recall of our chemical NERs on the CHEMDNER test data is significantly lower than precision, we tabulated their recall scores for each chemical subtype (Table 2.15). This allowed us to identify the subtypes which are most difficult for our NERs and, hence, are pulling down the overall recall. It can be observed that the most challenging subtypes for our NERs are the coordinated (Multiple) and uncategorised (Undefined) chemical mentions. This can be attributed to the fact that these two chemical subtypes have the least number of annotated instances in our training data, altogether accounting only for 0.82% and 0.75% of the annotated chemical entities in the CHEMDNER training and development sets, respectively. The sparsity of annotations under these two categories prevented our CRF models from learning their features, leading to the low recall for these types.

Figure 2.2 shows the performance of Custom NERsuite (without any post-processing) visualised and compared against that of the state-of-the-art (OSCAR 4 and ChemSpot) and our baseline (Default NERsuite). It illustrates that the addition of our proposed features into the CRF model improves the performance across all metrics, in both data sets. We performed a test to validate that this performance gain is statistically significant. In order to determine which test is most appropriate, we test if our samples (i.e.,
Figure 2.2: Visualisation of comparative evaluation of chemical NERs

the F₁ score for each document) come from a normal distribution using the Shapiro-Wilk test \([113]\). A \(p\)-value < 2.2\(\times\)10⁻¹⁶ was obtained, hence we reject the null hypothesis that our samples are normally distributed. A suitable significance test, therefore, is the Wilcoxon signed-rank test, a paired \(t\)-test which does not assume that the samples come from a normal distribution. Setting the F₁ scores obtained by Default NERsuite on the CHEMDNER and SCAI documents as one population and those obtained by Custom NERsuite as the other, we performed the one-sided version of this test and obtained a \(p\)-value < 1.002\(\times\)10⁻¹⁴. As the \(p\)-value is very small, we can reject the null hypothesis that the introduction of our additional features does not improve the performance of chemical NER.
Table 2.14: Comparative evaluation of chemical NERs on CHEMDNER Test Data and SCAI Pilot Corpus

<table>
<thead>
<tr>
<th>Chemical NER</th>
<th>CHEMDNER Test Data</th>
<th></th>
<th></th>
<th>SCAI Pilot Corpus</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>OSCAR 4</td>
<td>45.075</td>
<td>76.999</td>
<td>56.862</td>
<td>50.882</td>
<td>81.343</td>
<td>62.604</td>
</tr>
<tr>
<td>ChemSpot</td>
<td>72.633</td>
<td>57.256</td>
<td>64.034</td>
<td>76.068</td>
<td>70.896</td>
<td>73.391</td>
</tr>
<tr>
<td>Default NERsuite</td>
<td>91.596</td>
<td>79.318</td>
<td>85.016</td>
<td>77.657</td>
<td>76.949</td>
<td>77.301</td>
</tr>
<tr>
<td>Custom NERsuite</td>
<td>92.664</td>
<td>80.813</td>
<td>86.334</td>
<td>78.464</td>
<td>77.944</td>
<td>78.203</td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg</td>
<td>92.023</td>
<td>81.232</td>
<td>86.291</td>
<td>78.109</td>
<td>78.109</td>
<td>78.109</td>
</tr>
<tr>
<td>Custom NERsuite+Abbr</td>
<td><strong>92.674</strong></td>
<td>81.235</td>
<td><strong>86.579</strong></td>
<td><strong>78.607</strong></td>
<td>78.607</td>
<td><strong>78.607</strong></td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg+Abbr-1</td>
<td>92.036</td>
<td>81.650</td>
<td>86.532</td>
<td>78.254</td>
<td>78.773</td>
<td>78.512</td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg+Abbr-2</td>
<td>90.622</td>
<td><strong>82.407</strong></td>
<td>86.319</td>
<td>77.163</td>
<td><strong>79.851</strong></td>
<td>78.484</td>
</tr>
</tbody>
</table>

Table 2.15: Recall scores for each chemical subtype in the CHEMDNER test data

<table>
<thead>
<tr>
<th>Chemical NER</th>
<th>Abbreviation</th>
<th>Family</th>
<th>Formula</th>
<th>Identifier</th>
<th>Multiple</th>
<th>Systematic</th>
<th>Trivial</th>
<th>Undefined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom NERsuite</td>
<td>62.404</td>
<td>81.225</td>
<td><strong>71.507</strong></td>
<td>72.709</td>
<td>41.708</td>
<td>90.028</td>
<td>89.280</td>
<td>53.658</td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg</td>
<td>62.601</td>
<td>82.192</td>
<td>71.478</td>
<td>72.709</td>
<td>41.708</td>
<td>90.398</td>
<td>89.830</td>
<td>53.658</td>
</tr>
<tr>
<td>Custom NERsuite+Abbr</td>
<td>64.720</td>
<td>81.225</td>
<td><strong>71.507</strong></td>
<td>72.709</td>
<td>41.708</td>
<td>90.257</td>
<td>89.280</td>
<td>53.658</td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg+Abbr-1</td>
<td>64.892</td>
<td>82.192</td>
<td>71.478</td>
<td>72.709</td>
<td>41.708</td>
<td>90.628</td>
<td>89.830</td>
<td>53.658</td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg+Abbr-2</td>
<td><strong>66.050</strong></td>
<td><strong>84.152</strong></td>
<td>71.420</td>
<td>72.709</td>
<td><strong>43.216</strong></td>
<td><strong>90.945</strong></td>
<td><strong>90.535</strong></td>
<td>53.658</td>
</tr>
</tbody>
</table>
Table 2.16: Comparison against MetaboliNER on the Metabolites corpus.

<table>
<thead>
<tr>
<th>Chemical NER</th>
<th>Metabolites Corpus</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaboliNER</td>
<td>83.02</td>
<td>74.42</td>
<td>78.49</td>
<td></td>
</tr>
<tr>
<td>Default NERsuite</td>
<td>82.773</td>
<td>71.300</td>
<td>76.610</td>
<td></td>
</tr>
<tr>
<td>Custom NERsuite</td>
<td>83.800</td>
<td>75.703</td>
<td>79.546</td>
<td></td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg</td>
<td>81.336</td>
<td>79.943</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom NERsuite+Abbr</td>
<td><strong>83.822</strong></td>
<td>75.826</td>
<td>79.623</td>
<td></td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg+Abbr-1</td>
<td>81.191</td>
<td>80.065</td>
<td>80.624</td>
<td></td>
</tr>
<tr>
<td>Custom NERsuite+ChemSeg+Abbr-2</td>
<td>77.838</td>
<td><strong>81.614</strong></td>
<td>79.682</td>
<td></td>
</tr>
</tbody>
</table>

It is also of interest to compare our method with that of Nobata et al. which was applied to the recognition of metabolite names. Following the same steps for evaluation that they did, we performed ten-fold cross validation on the Metabolites corpus. This was carried out by splitting the corpus into ten non-overlapping subsets containing approximately the same number of documents. For each fold $f$ where $1 \leq f \leq 10$, a CRF model was trained on the combination of subsets from the other folds. Training was carried out in the same manner as with our chemical NER, except that the Human Metabolome Database (HMDB) and ChEBI were leveraged in the generation of dictionary features, as Nobata et al. did. The model trained for fold $f$ is then used to generate responses for subset $f$. Responses from each of the ten folds were then collated and evaluated against the gold standard annotations. Presented in Table 2.16 are our results, together with those reported for MetaboliNER. Each variant of our proposed method performed better than MetaboliNER, with the combination of the CRF model and the chemical segment composition computation achieving the best $F_1$ score of 80.633%.
2.3. EVALUATION AND DISCUSSION

2.3.2 Drug NER

Drugs can be considered a subset of chemical entities, i.e., those which possess pharmacological properties or have been used in pharmacological applications. Recognising them with a general chemical named entity recogniser, therefore, will expectedly yield satisfactory results in terms of recall but not precision. For this reason, we developed a drug-specific named entity recogniser which is still founded on our proposed methodology but with a few modifications.

2.3.2.1 Training and development

Amongst the corpora containing drug named entity annotations, the DDI corpus stands out for having the most gold standard mentions. Provided as a resource for the SemEval 2013 DDI Extraction task [114], the corpus contains a total of 714 training and 112 test documents. We used the training set in the development of our CRF models.

In training a model for recognising drugs, the same pre-processing pipeline and enriched feature set for our chemical NER were employed. The only necessary modification was the elimination of dictionary features which are not drug-specific. Of the five databases/lexica we leveraged during dictionary features generation, DrugBank is the only resource focussed specifically on drugs. Matching the tokens against the four others (i.e., ChEBI, CTD, PubChem Compound and Jochem) was therefore skipped during the generation of features for our drug NER.

As it is desirable to compare the performance of our drug named entity recogniser with that of the state-of-the-art (i.e., those reported in the DDI Extraction DrugNER task), we conformed with the shared task’s requirement for the capability to disambiguate between four different drug subtypes: generic names, brands, groups and non-human consumption substances. In order to meet this requirement, the following two approaches were explored:
training of four different models, one for each subtype, and

training of only one joint model that can assign specific subtype labels whilst
recognising all four

The first approach required model training to be performed as many times as the
number of subclasses of interest (i.e., four). Each model was trained in a one-vs-all
fashion, i.e., by taking the annotations of the currently considered subtype as positive
instances whilst treating the rest as negative, including those falling under the three
other drug subtypes. The final set of responses is then formed by merging together the
individual annotations from each model.

In contrast, model training in the second approach was done only once. The sub-
type labels of all drug name annotations are kept and presented to the CRF model
during training. In this way, the drug NER utilises a single model trained to disam-
biguate between the four different subtypes in one step, instead of performing a series
of one-vs-all decisions like in our other approach.

2.3.2.2 Comparison

The DDI test corpus, consisting of 54 DrugBank documents and 58 MEDLINE ab-
stracts, was used in benchmarking our drug named entity recogniser. In evaluating
our NER, we utilised the official evaluation software provided by the organisers of the
SemEval DDI Extraction DrugNER task, to make the evaluation comparable to the
similarly reported state-of-the-art. The evaluation software measures the performance
of a drug NER by comparing its responses against gold standard annotations according
to the following four criteria [114]:

- Strict matching: requires both boundaries and drug subtypes to match
- Exact boundary matching: requires only the boundaries to match
• Partial boundary matching: requires boundaries to at least be partially overlapping but does not require the subtypes to match

• Type matching: requires boundaries to at least be partially overlapping and subtypes to match.

Presented in Table 2.17 are the results of our two approaches on the entire DDI test corpus, alongside those of the other research groups as reported in the DDI Extraction 2013 workshop [114]. The entries appear in order of increasing $F_1$ according to the strict matching criterion, the most stringent of the four. It can be observed that our first approach which utilises multiple models (Custom NERsuite-Multiple) consistently achieves better precision than our second approach (CustomNERsuite-Joint). However, the joint model consistently outperformed our first approach in terms of recall and $F_1$ score. Also, with the employment of our enriched feature set, an increase of 3.8 percentage points in $F_1$ score (measured using strict matching) was achieved, both in the multiple model and joint approaches.

When compared with the official results of other research groups’ drug NER systems, the scores of the Custom NERsuite implementations ranked the highest, based on each of the strict and type matching criteria. Whilst an increase of at least 3 percentage points is obtained by using any of the criteria less stringent than strict matching, the increase obtained by the NER system of the WBI group from exact and partial boundary matching is significantly greater, leading to our results landing the second place in these two criteria.

It is worth noting that in both of our approaches, there was no difference between the results from exact boundary matching and those from partial boundary matching. This implies that there were no instances of partially overlapping annotations; our drug NERs were either exactly matching the boundaries of the gold standard annotations or

\footnote{a more drug-specific variant of the CRF-based chemical NER tool ChemSpot}
completely missing them.

In Table 2.18, we present the results of strict matching for each drug subtype, in order of increasing macro-averaged F\textsubscript{1} score. Whilst our drug NER, Custom NERsuite-Joint, performed poorly in the recognition of non-human use substances, it obtained superior performance in both generic and group name categories. It also came a close second in recognising brand names, behind UTurku’s NER by only 0.5 percentage points. Again, it is noticeable that the joint model approach consistently outperformed the use of multiple models in terms of recall and F\textsubscript{1} score.

The poor performance in the recognition of non-human consumption substances can be attributed to the sparsity of annotations for this subtype in the corpus. In the training set, substances for non-human use account for only 3.41\% of the drug name annotations (or 504 out of 14,765). The small set of instances proved insufficient for training the CRF model to recognise entities of this subtype.
### Table 2.17: Summary of overall drug NER results on the entire DDI test corpus

<table>
<thead>
<tr>
<th>Drug NER</th>
<th>Strict</th>
<th>Exact boundary</th>
<th>Partial boundary</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>UEM UC3M</td>
<td>51.7</td>
<td>54.2</td>
<td>52.9</td>
<td>59.4</td>
</tr>
<tr>
<td>UTurku</td>
<td>73.7</td>
<td>57.9</td>
<td>64.8</td>
<td>75.7</td>
</tr>
<tr>
<td>NLM_LHC</td>
<td>73.2</td>
<td>67.9</td>
<td>70.4</td>
<td>82.3</td>
</tr>
<tr>
<td>WBI</td>
<td>73.4</td>
<td>69.8</td>
<td>71.5</td>
<td>85.5</td>
</tr>
<tr>
<td>Default NERsuite-Multiple</td>
<td>81.8</td>
<td>65.0</td>
<td>72.5</td>
<td>89.7</td>
</tr>
<tr>
<td>Default NERsuite-Joint</td>
<td>77.9</td>
<td>67.9</td>
<td>72.6</td>
<td>88.6</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple</td>
<td>87.7</td>
<td>67.5</td>
<td>76.3</td>
<td>90.9</td>
</tr>
<tr>
<td>Custom NERsuite-Joint</td>
<td>84.7</td>
<td>69.5</td>
<td>76.4</td>
<td>90.2</td>
</tr>
</tbody>
</table>

### Table 2.18: Summary of drug NER type-level results on the DDI test corpus

<table>
<thead>
<tr>
<th>Drug NER</th>
<th>Generic name</th>
<th>Brand</th>
<th>Group</th>
<th>Drug_n</th>
<th>Macro Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>UEM UC3M</td>
<td>68.2</td>
<td>83.8</td>
<td>75.2</td>
<td>15.4</td>
<td>6.8</td>
</tr>
<tr>
<td>NLM_LHC</td>
<td>72.5</td>
<td>91.7</td>
<td>81.0</td>
<td>77.5</td>
<td>93.2</td>
</tr>
<tr>
<td>UTurku</td>
<td>80.1</td>
<td>76.6</td>
<td>78.3</td>
<td>94.5</td>
<td>88.1</td>
</tr>
<tr>
<td>Default NERsuite-Multiple</td>
<td>89.5</td>
<td>80.1</td>
<td>84.5</td>
<td>100.0</td>
<td>74.6</td>
</tr>
<tr>
<td>WBI</td>
<td>73.6</td>
<td>85.2</td>
<td>79.0</td>
<td>81.0</td>
<td>86.4</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple</td>
<td>94.3</td>
<td>85.2</td>
<td>89.5</td>
<td>100.0</td>
<td>74.6</td>
</tr>
<tr>
<td>Default NERsuite-Joint</td>
<td>87.8</td>
<td>84.1</td>
<td>85.9</td>
<td>100.0</td>
<td>81.4</td>
</tr>
<tr>
<td>Custom NERsuite-Joint</td>
<td>93.3</td>
<td>86.6</td>
<td>89.8</td>
<td>100.0</td>
<td>83.1</td>
</tr>
</tbody>
</table>
Shown in Figure 2.3 is a graph illustrating the overall performance of our top-performing drug NER, Custom NERsuite-Joint, alongside that of the best system in the DDI Extraction DrugNER task (WBI). Also included is the performance of our baseline method, Default NERsuite-Joint, which employs a CRF model trained on default built-in features. It can be observed that by both strict matching and macro-averaging over drug subtypes, we gain an improvement in F$_1$ scores by the introduction of our additional features. To evaluate the significance of these features, a one-tailed paired Wilcoxon signed-rank test was performed on the F$_1$ scores obtained on the DDI test set by strict matching. A p-value of 0.001233 was obtained, signifying that the improvement gained by employing our enriched feature set is statistically significant.

As an additional evaluation step, we considered running our drug NER on other corpora containing drug named entity annotations, namely, ADE, PK-DDI, PK and EU-ADR, which have all been reviewed in Section 2.1. Whilst they both contain annotated drug names, ADE and PK-DDI are not suitable as gold standard corpora for the NER task since not all instances of drug names were annotated in these two. Mentions of drugs were marked up only if they appeared in text within the context of drug-drug interactions. Consequently, mentions pertaining to a drug were annotated only in certain sentences but not in others. Shown in Examples 2.5 and 2.6, for instance, are two sentences coming from the same package insert in the PK-DDI corpus. The drug name AVANDIA was annotated in the sentence in Example 2.5 which describes a DDI, but not in 2.6. As the detection of the presence of DDIs is not within the scope of the drug NER task, these two corpora are not suitable for evaluating our drug NER.

Example 2.5. Repeat oral dosing of [AVANDIA] for 14 days did not alter the steady-state pharmacokinetics of [digoxin] in healthy volunteers.

Example 2.6. Patients with lipid abnormalities were not excluded from clinical trials of AVANDIA.
2.3. **EVALUATION AND DISCUSSION**

![Figure 2.3: Visualisation of comparative evaluation of drug NERs](image)

We nevertheless proceeded with the evaluation of our drug NER on the PK and EU-ADR corpora. Shown in Table 2.19 are the micro-averaged scores of different variants of our two approaches on these two corpora. With the use of the model trained solely for generic drug names, we obtained optimal precision on both corpora. Optimal recall is achieved by the joint model, specifically when none of the drug subtypes recognised were filtered out. Whilst the performance of our drug NER is more than satisfactory on the PK corpus, it exhibited weak performance on the EU-ADR corpus. Upon inspection of the false negatives, we observed that there is a significant number of annotated drug names which correspond to pharmacological roles. For example, mentions such as *IL-1 receptor antagonist* and *acid ceramidase inhibitors* were marked up as drugs in the EU-ADR corpus. Such cases particularly caused difficulty for our drug NER as expressions pertaining to roles are not considered as drug entities, as far as the definition that we adopted is concerned.
Table 2.19: Summary of drug NER results on the PK and EU-ADR corpora. Indicated in parentheses are the drug subtypes included in the responses; d=generic drugs, b=brand, g=group.

<table>
<thead>
<tr>
<th>Drug NER</th>
<th>PK Corpus</th>
<th>EU-ADR Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple (all)</td>
<td>84.889</td>
<td>81.614</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple (d)</td>
<td>92.078</td>
<td>80.376</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple (d+b)</td>
<td>91.824</td>
<td>80.470</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple (d+g)</td>
<td>87.508</td>
<td>80.574</td>
</tr>
<tr>
<td>Custom NERsuite-Multiple (d+b+g)</td>
<td>87.285</td>
<td>80.668</td>
</tr>
<tr>
<td>Custom NERsuite-Joint (all)</td>
<td>81.976</td>
<td><strong>83.860</strong></td>
</tr>
<tr>
<td>Custom NERsuite-Joint (d)</td>
<td>90.209</td>
<td>81.437</td>
</tr>
<tr>
<td>Custom NERsuite-Joint (d+b)</td>
<td>89.632</td>
<td>81.812</td>
</tr>
<tr>
<td>Custom NERsuite-Joint (d+g)</td>
<td>85.492</td>
<td>81.624</td>
</tr>
<tr>
<td>Custom NERsuite-Joint (d+b+g)</td>
<td>84.995</td>
<td>81.999</td>
</tr>
</tbody>
</table>

2.4 Applications

In this section, applications of our methods to other chemical text mining tasks are described.

2.4.1 Chemical Document Indexing

The BioCreative IV CHEMDNER track was organised to encourage NLP researchers to implement tools for recognising names of chemical compounds and drugs in text. One of the tasks defined by the organisers is chemical document indexing (CDI), which required participating systems to return a list of unique chemical entities described in a given document. It is different from named entity recognition in that it does not require the locations or boundaries of the entity mentions. Additionally, it was necessary for the unique mentions to be ranked according to how confident the chemical NER is in returning each. The CHEMDNER corpus was provided to the participants by the organisers as a development resource.

\[\text{defined at the surface level}\]
The work on chemical NER described previously was directly applied to this sub-task. Exploiting the probability values returned by our CRF model for each token, our approach calculates its confidence in recognising a chemical named entity by taking the average of the probabilities over the tokens composing a name. In cases where there are multiple instances of a chemical name, the highest confidence is the one retained. The list of unique chemical name mentions is returned in decreasing order of the confidence scores.

Presented in Table 2.20 are the official evaluation results on our method (University of Manchester), reported in terms of micro-averaged scores. To put into perspective how our method compares with that of a few other participants, we also provide the results of their best performing methods alongside ours. In reporting the rankings, the organisers grouped together participants whose results do not have any statistically significant difference. This was done by calculating the micro-averaged \( F_1 \) score of each participating method on each document in a set of 1,000 randomly selected CHEMDNER test documents. The standard deviation (SD) over each participant’s F-scores is then calculated and used to define ranges over participants. The range of participant \( P_1 \) consists of itself and other participants whose F\(_1\) scores vary from that of \( P_1 \) by at most twice its own SD. For example, the range of Wuhan includes Manchester and Humboldt but not NCBI/NLM/NIH. Subtracting twice of Wuhan’s SD from its F\(_1\) score, we reach an overlap with the F\(_1\) scores of Manchester and Humboldt, but not with that of NCBI/NLM/NIH. A range represents a grouping of participant results whose differences from each other are not statistically significant; participants which were given the same range, therefore, were assigned the same ranking. Our method was ranked first, together with that of Wuhan University. The variant of the chemical NER which gave the best performance on the CHEMDNER test set is the one that utilises a CRF model trained on our enriched features and abbreviation detection as a post-processing step (Custom NERsuite+Abbr).
Table 2.20: Official BioCreative IV CHEMDNER evaluation results for the chemical document indexing task (top 3 out of 18)

<table>
<thead>
<tr>
<th>Row</th>
<th>Team</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
<th>SD</th>
<th>Range</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Wuhan University</td>
<td>87.02</td>
<td>89.41</td>
<td>88.20</td>
<td>0.30</td>
<td>A-C</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>University of Manchester</td>
<td>91.28</td>
<td>85.24</td>
<td>88.15</td>
<td>0.34</td>
<td>A-C</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>Humboldt University</td>
<td>89.34</td>
<td>86.51</td>
<td>87.90</td>
<td>0.33</td>
<td>A-D</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>NCBI/NLM/NIH</td>
<td>87.81</td>
<td>87.24</td>
<td>87.52</td>
<td>0.33</td>
<td>C-D</td>
<td>3</td>
</tr>
</tbody>
</table>

2.4.2 Comparative Toxicogenomics Database chemical concept recognition

In the Fourth BioCreative Challenge Evaluation Workshop, a track was organised specifically for encouraging the text mining community to develop interoperable automatic tools that can possibly assist in the curation of the CTD database. It required the preparation of RESTful web services capable of accepting abstracts encoded in BioC [115], a new data interchange format advocated by BioCreative, and returning, within a minimal amount of time, a version enriched with unique entities found in the document. The entity types of interest are chemicals, genes, diseases and action terms. We cast the task as a named entity recognition problem, as we did the previously described CHEMDNER chemical document indexing task. In the following, we describe our approach for the automatic annotation of chemical entities.

At our disposal were the publicly available CTD vocabularies containing the preferred name and corresponding alternate names or synonyms for each concept curated in the database. Additionally, the organisers provided a learning corpus of 1,112 MEDLINE abstracts encoded in the BioC XML format. Each abstract contains a list of unique chemicals, genes, diseases and action terms which were manually identified by domain experts. In each concept annotation, only the preferred name as it appears in the CTD vocabularies is indicated. Since the training corpus does not contain the locations of entities nor the exact form in which they appear in the documents, the first
challenge we addressed was the generation of a silver-annotated corpus suitable for named entity recognition.

Leveraging the CTD chemical vocabulary, we determined the locations of chemical mentions in the abstracts in the provided training corpus using case-insensitive exact string matching. This, however, introduced a considerable amount of noise due to the ambiguity of certain names. In mitigating this problem, we exploited the on-line testing facility provided by the CTD track organisers in order to identify the false positives returned for each document and to filter them out. The remaining entities (i.e., the true positives) were then used in silver-annotating the documents in the corpus with their specific locations in text.

The silver-annotated corpus was split into subsets for training and testing. Following the same approach to chemical NER described in the preceding section, we employed the NERsuite package to train a CRF model for chemical names on the training subset. The BIO-labelled tokens given by the model were converted into names in the same manner described in the previous section. Around the time of preparing the web services, the heuristics-based post-processing steps were still under development; hence, they were not applied to our solution to the CTD task.

There are two items concerning evaluation which are worth noting:

- Although normalisation of entities to CTD was not a requirement, the testing facility, serving as the evaluation library for this track, calculated the number of successful matches between the responses and the gold standard annotations by attempting to map the former to the CTD preferred names in the latter using case-insensitive exact string matching.

- The task organisers have communicated to the participants that whilst a balance between precision and recall is desirable, optimal recall is more preferable than directly or indirectly through synonyms.

\(^{25}\)http://bc.ctdbase.org/ws
\(^{26}\)directly or indirectly through synonyms
Table 2.21: Approximate string matching algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Response named entity</th>
<th>CTD chemical entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>synthetic conjugated estrogens A</td>
<td>Duramed brand of synthetic conjugated estrogens, A</td>
</tr>
<tr>
<td></td>
<td>Lowercasing</td>
<td>synthetic conjugated estrogens a</td>
</tr>
<tr>
<td></td>
<td>Stop word removal</td>
<td>synthetic conjugated estrogens a</td>
</tr>
<tr>
<td></td>
<td>Stemming</td>
<td>synthetic conjugated estrogen a</td>
</tr>
<tr>
<td></td>
<td>Reordering</td>
<td>a conjug estrogen synthet</td>
</tr>
</tbody>
</table>

precision as far as actual CTD curation is concerned.

Considering these points, we incorporated exact string matching between our responses and CTD names as an intermediate step towards isolating the responses for which no preferred names were found. This allowed us to check against our test set, examples of such responses and to observe that some of them are morphological variations of CTD names or synonyms. As the aim is to optimise recall by increasing the number of matches, an approximate string matching algorithm was introduced to identify and return the most similar CTD name or synonym for each of such responses. This algorithm is based on the steps outlined in Table [2.21] performed on both the response annotation and the entries in the CTD chemical vocabulary.

The resulting canonical form of the response annotation $C_r$ is paired with that of each of the CTD entries $C_e$. The ratio between the number of common tokens in $C_r$ and the number of tokens in $C_e$ is calculated. For the given example in the table, the ratio will be $4/6$ or 0.67 between a conjug estrogen synthet ($C_r$) and a brand conjug duram estrogen synthet ($C_e$). We then take as a candidate match the $C_e$ of the pair with the highest score. If this score is greater than an established threshold of 0.20, the CTD entry corresponding to $C_e$ is returned.
Table 2.22: Official BioCreative IV CTD evaluation results for the chemical concept recognition task

<table>
<thead>
<tr>
<th></th>
<th>Micro-averaged</th>
<th>Macro-averaged</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>75.236</td>
<td>73.406</td>
</tr>
</tbody>
</table>

The official evaluation was carried out by the task organisers on 510 test MEDLINE abstracts. Shown in Table 2.22 are the performance scores of our chemical concept recogniser. Our method (#184) achieved the best balanced F-score, compared with those of the nine other participants (Figure 2.4). Whilst six other participating methods obtained better recall than ours did, their approaches (except for one) achieved below-average precision which, in turn, led to below-average F-scores. Our web service was also capable of returning results within reasonable time, at an average of 0.77 seconds for each abstract.

Our web services for recognising chemicals and genes achieved the best performance (in terms of $F_1$ score) in the respective categories. The task organisers also calculated the combined F-score average over all four concept types for each participant. Comparison of the combined averages showed that our web services delivered the strongest overall performance.

### 2.4.3 Availability and Interoperability

The chemical NER has been deployed as an available processing component in Argo\(^\text{27}\), a web-based text mining platform that facilitates the creation and execution of user-defined NLP workflows \(^\text{116}\). Conforming with the Unstructured Information Management Architecture (UIMA), Argo alleviates the issue of interoperability by providing data serialisers/deserialisers which can convert into/from various data formats. The

\(^{27}\)http://argo.nactem.ac.uk
Figure 2.4: Performance of our chemical concept recogniser web service (#184) compared with that of other participants. Graph taken with permission from the CTD task overview paper [2].
annotation formats it currently supports include: CHEMDNER DSV, BioC, XML\textsuperscript{28} and the BioNLP Shared Task formats. By deploying our chemical NER in Argo, we are enabling users to incorporate it into their own workflows without imposing any restrictions on the input/output formats. Furthermore, as Argo also allows the creation of multi-branched evaluation workflows, our chemical NER can be benchmarked against a user-selected corpus and/or compared with other NERs.

2.5 Summary

We have presented in this chapter our study on chemical named entity recognition. After providing our review and comparison of existing lexical resources, annotated corpora and published approaches, we introduced our own approach to the problem which involved the incorporation of chemistry-specific features into the CRF model as well as post-processing rules for the recognition of abbreviations and computation of chemical segment composition. A thorough evaluation of our proposed method has been performed. Our chemical NER consistently outperformed OSCAR 4 and ChemSpot, the current state-of-the-art tools, on the CHEMDNER test and SCAI pilot corpora. Similarly, our drug NER outperformed the best performing system in the DDI Extraction DrugNER shared task, evaluated on the DDI test data using strict matching. Moreover, results of our chemical NER on the Metabolites corpus are better than those reported for the MetaboliNER tool. As the difference between our method and that employed in MetaboliNER lies only in our proposed chemistry-specific features and post-processing rules, we can conclude that our extensions improve the performance of chemical NER.

\textsuperscript{28}via a generic XML parser
Chapter 3

Coreference Resolution

One of the means by which authors increase readability and informativeness of their writing is through the use of various linguistic elements. On the one hand, readability is improved when the writing style avoids excessive repetition of the same names within the same discourse. Hence, in writing, we often substitute previously mentioned names with pronouns (e.g., he, she and it). Informativeness, on the other hand, is increased when noun phrases introducing new information into discourse (e.g., appositives, definite noun phrases) are included in the text.

In interpreting text which was written using such elements, one of the challenges lies in determining which particular entity is being referred to (referent) by which expression (referring expression), a task known as reference resolution [117]. It covers two subtasks, namely, coreference resolution and anaphora resolution.

Coreference is a linguistic phenomenon characterised by one or more expressions referring to a unique referent, also known as coreferring expressions [117]. Often associated with coreference is anaphora which is characterised by an expression (anaphor) whose interpretation depends on a previous mention in the same discourse (antecedent).

Coreferring mentions in text form a set known as a coreference chain which can be treated as an equivalence class. By automatically grouping all coreferring mentions
in text into respective coreference chains, we are performing a task known as coreference resolution. If, in contrast, we are attempting to determine the antecedent of an anaphor, we are realising a task called anaphora resolution. This study is focussed on the coreference resolution task.

Whilst the output of anaphora resolution is a set of anaphor-antecedent pairs, that of coreference resolution is a set of coreference chains. Despite this difference, an overlap between them may be observed in several cases. Often, an anaphor and its antecedent are coreferential (i.e., having the same referent) and hence may be placed in the same coreference chain, as in the following example:

Example 3.1. John gave his wife a necklace for her birthday. She thanked him for it.

Example 3.1 contains the following coreference chains: {John, his, him}, {his wife, her, She} and {necklace, it}. The anaphor and antecedent in each of the anaphoric pairs {his, John}, {her, his wife}, {She, his wife}, {him, John} and {it, necklace} fall within the same chain. In some scenarios, however, this does not hold. Each of the following two examples illustrates the case where an anaphor and its antecedent are not coreferential.

Example 3.2. Peter received his paycheque yesterday but John didn’t get one.

Example 3.3. The structure of compound 3 was elucidated through NMR spectroscopic analysis while that of 2 was elucidated by mass spectroscopic analysis.

Example 3.2 includes the anaphoric pairs {his, Peter} and {one, paycheque}. Whilst his and Peter refer to the same person and thus belong to the same coreference chain, one and paycheque do not corefer since the paycheque that John is expecting is a different entity from Peter’s paycheque. The coreference chains in this example, therefore, are: {Peter, his}, {paycheque (Peter’s)}, {yesterday}, {John} and {one (John’s paycheque)}. Similarly, whilst that and structure in Example 3.3 are in an anaphoric
relation, they do not corefer since the chemical structure of the compound denoted by 2 is a different entity from the structure of compound 3.

The converse also holds; two expressions may be coreferential but not belonging to an anaphor-antecedent relation. This is true when we consider coreferring expressions across multiple documents. Two expressions from two different documents may refer to the same entity in the real world, but an anaphoric link between them cannot be established as the documents are considered as different works of discourse. In this research, however, there is no intention to investigate cross-document coreference resolution, i.e., coreferring expressions will be resolved only on a per-document basis. Hence, we can treat coreferent expressions as anaphoric, i.e., by taking any two expressions from a coreference chain and linking them through an anaphoric relation (with the one occurring later in the discourse as the anaphor and the earlier one as its antecedent). For this reason, we will also summarise in the following review of relevant work on coreference resolution the state-of-the-art in anaphora resolution.

In the 6th Message Understanding Conference (MUC-6) held in 1995, automatic annotation of coreferring expressions was included as one of the tasks, with the objective of supporting information extraction systems. The annotations were done on news reports from the subject of labour dispute negotiation and corporate management succession [52]. The same task was included in the succeeding conference MUC-7, but this time on news reports on airplane crashes and aircraft launches [53]. Pioneering work on coreference resolution commenced around this time and has been carried out mostly on documents from the general domain.

Nevertheless, a handful of studies on coreference resolution in the biomedical domain have emerged, mostly on resolving expressions pertaining to gene or gene products. Recently, there has also been interest in resolving anaphora to support drug-drug interaction extraction in pharmacological text [118].

Both coreference and anaphora resolution tasks are non-trivial and still considered
unsolved, especially that for scientific literature. It is an especially interesting and challenging problem in the domain of chemistry where various types of expressions are used to refer to the same chemical compound, including, but not limited to, its systematic name, trivial name, brand name, chemical structure, abbreviation as well as anaphoric expressions such as pronouns and author-assigned numerical labels. Currently, however, solutions to this problem have not been fully explored, motivating us to investigate methods for coreference resolution in this specialised domain. In the following review of relevant corpora and proposed approaches, we start by describing efforts towards general-domain coreference resolution, and proceed to summarising prior art in the biomedical domain.

### 3.1 Literature Review

#### 3.1.1 Annotated Corpora

The phenomenon of coreference has diverse representation formats: in some corpora, coreferring mentions are represented as coreference chains; in most of them, however, they are linked pairwise, i.e., as anaphor-antecedent pairs.

##### 3.1.1.1 Corpora from the general domain

**MUC Corpora.** The MUC-6 and MUC-7 general-domain corpora were developed following the pairwise linking scheme [52][53]. They consist of news articles in which expressions pertaining to nouns, noun phrases and pronouns (called markables) were annotated. Where necessary, links between them were created by annotators to signify coreference [52][53]. Covering personal, demonstrative and possessive pronouns, the annotation scheme was implemented in the Standard Generalized Markup Language (SGML) format. Each markable is contained in a coref element which has
attributes such as id (a unique identifier), ref (for creating links) and min (the minimum string that should be given by a system being evaluated for scoring). The link between coreferring expressions is established by setting the value of the ref attribute of the anaphoric expression to the value of the id attribute of the antecedent. This scheme, which accounts only for identity relations, was eventually called the MUC scheme (MUCSS). Both of the MUC-6 and MUC-7 corpora can be obtained by purchase of a license from the Linguistic Data Consortium (LDC).[1]

**GNOME Corpus.** The GNOME Corpus [119] is a document collection developed by Poesio et al. as a resource for studying discourse salience and for evaluating anaphora resolution systems. It consists of three subcorpora, namely, the museum subcorpus which contains descriptions of museum objects, the pharmaceutical corpus which contains leaflets with legally mandatory information about medicines, and the tutorial dialogues corpus. A simplified version of the MATE annotation scheme [120] was employed in the annotation of the documents. Implemented as a markup language, it defines an ante element for representing anaphor-antecedent pairs. First, all markables are annotated in-line using the ne element. A stand-off instance of the ante element is then created and assigned the identifier of the anaphor. Embedded inside this instance is an anchor element having the identifier of the antecedent. Multiple anchor elements may be embedded into an ante element, in cases where the anaphor is ambiguous (i.e., refers to multiple antecedents). Unlike the MUC scheme which supports only anaphor-antecedent relations of the identity type, the variant of MATE which was used in the GNOME corpus allows for the annotation of part-whole, set-subset and possessive relation types. As some of the texts contained in the corpus are copyrighted, the GNOME corpus is not publicly available.

---

[1]https://www.ldc.upenn.edu
3.1. LITERATURE REVIEW

ACE 2005 Corpus. The Automatic Content Extraction (ACE) 2005 corpus was developed to support the 2005 edition of the ACE program [121]. One of the five tasks defined as part of the program are the entity, relation and event mention detection tasks which required that all mentions of each entity/relation/event (i.e., coreferring expressions) be recognised and grouped together. To support this task, the documents in the corpus (coming from broadcast news, broadcast conversation, newsgroups and weblogs) were annotated with coreference information. The annotation scheme, implemented in XML, specifies a representation different from that of the MUCSS. Instead of being treated as a set of anaphor-antecedent pairs, coreferential expressions were grouped into one coreference chain. Entities are categorised into the following types: person, organisation, location, geo-political entity, facility, vehicle and weapon. All mentions of such entities were marked up, including proper names, common nouns and pronouns. Each is annotated using a stand-off entity_mention element. Mentions referring to the same entity are then grouped together using an entity element. Consisting of document sets for each of Arabic (403 documents), Chinese (633 documents) and English (599 documents), the ACE 2005 corpus can be purchased from the Linguistic Data Consortium.

3.1.1.2 Corpora from the biomedical domain

MEDSTRACT Corpus. The MEDSTRACT Corpus [122] was developed as part of the MEDSTRACT project [123] which aims to build tools for extracting information from biomedical text. Consisting of MEDLINE abstracts (54 in the development set and 46 in the test set), this corpus is the first biomedical document collection to be annotated with anaphoric information. Similar to the MUCSS, the scheme employed in the annotation of this corpus represents coreferences in a pairwise manner, i.e., using a link from an anaphoric expression to its antecedent. The documents were encoded in
XML format, with the markables annotated in-line using entity elements. To mark-up anaphoric relations, an antecedent attribute is embedded into the entity element encapsulating the anaphor; its value is set to the identifier of the antecedent. More than one antecedent can be linked, in the case of ambiguous anaphors. A domain expert was asked to annotate all anaphoric relations having pronouns and lexical noun phrases. The corpus is available for public download\(^2\).

**MEDCo Corpus.** The MEDCo Corpus [124] is the result of adding coreference annotations to the GENIA Corpus which consists of documents from the domain of molecular biology (1,999 abstracts and 43 full papers from MEDLINE). The annotation scheme employed is very much similar to MUCSS, but covering relation types other than identity, namely, pronominal, part-whole, whole-part, appositive and relative. The subset of abstracts is publicly accessible\(^3\) the full articles, however, have yet to be made available. The annotations in the abstracts subset form the basis of the data sets for the BioNLP 2011 Protein Coreference Resolution task [125].

**FlySlip Corpus.** Gasperin et al. developed a corpus of five full papers on fruit fly genes which were annotated with coreference information. A specialised annotation scheme which was tailored for the domain was developed. With this scheme, only noun phrases corresponding to gene mentions and other gene-related mentions were marked. The links between such mentions were then created and categorised into coreferent and associative relations. Coreferent relations (similar to MUCSS’ identity relation) pertain to relations between two mentions referring to the same entity, whilst associative relations are applicable to mentions which are related not by identity but by some other relation such as what they call the biotype relation (mentions are of different biotypes), homolog relation (mentions are of the same biotype but different homologs), or the

\(^2\)http://www.medstract.org
\(^3\)http://nlp.i2r.a-star.edu.sg/medco.html
set-member relation. By their definition, biotypes are entities which are gene-related, further classified into gene, gene product, gene subtype, gene part, sequence supertype and gene variant. Before manual annotation, the corpus was first automatically pre-annotated by a gene name recogniser. Using the MMAX tool, annotators validated the automatically generated annotations which entailed correcting the entity type given by the recogniser as well as tagging missed gene names. Thereafter, names of other gene-related entities were tagged and assigned the correct biotype. Finally, links between anaphors and antecedents were created and assigned the appropriate relation type. The corpus is publicly accessible.

**CRAFT Corpus.** The Colorado Richly Annotated Full Text (CRAFT) corpus, a joint project between the University of Colorado School of Medicine and the Linguistics Department of the University of Colorado at Boulder, was developed as a resource to support text mining research geared towards biomedical curation. It consists of 97 full papers on topics relevant to laboratory mice. Coreference annotation has been recently added to the corpus, on top of linguistic and term annotations. After reviewing several annotation schemes such as MUCSS, OntoNotes, that of MEDCo and that of FlySlip, the CRAFT team adapted the OntoNotes annotation guidelines. The types of relations covered by the scheme include identity and appositive relations. Aside from nominal and pronominal phrases, verbs and events (nominalised verbs) were also considered as markables. A notable feature of the OntoNotes annotation scheme is the way by which annotators establish relationships amongst coreferring expressions. Whereas with the other schemes the annotator links an anaphor to its antecedent, with OntoNotes, the annotator places coreferring expressions into the same coreference chain. After annotation, therefore, there are a number of stand-off coreference chains, each of which is a set containing coreferring expressions. The version

\[\text{http://www.wiki.cl.cam.ac.uk/rowiki/NaturalLanguage/FlySlip/Flyslip-resources}\]
of the CRAFT corpus which has been released does not include yet coreference annotations; the annotation task has yet to be completed [128].

**DDI Corpus.** For the design and evaluation of DrugNerAr [118], a drug anaphora resolution system, a corpus on the domain of drug interactions was developed [129]. Consisting of 49 documents derived from the DrugBank database, the corpus was initially automatically processed by the MetaMap Transfer (MMTx) tool and the authors’ own NER tool for linking of phrases to UMLS concepts and for the classification of drug entities into drug families, respectively. Manual annotation of anaphoric relations whose anaphors are either nouns or pronouns was then carried out by a linguist. The scheme specifies that an XML element corresponding to an anaphoric expression is linked to its antecedent by setting the value of its id_antecedent attribute to the identifier of the antecedent element. The corpus is available for public download[^5].

### 3.1.1.3 Comparison

Presented in Table 3.1 are the coreference-annotated corpora reviewed above. We identify the types of coreferences annotated in each corpus, adapting Mitkov’s classification of anaphora [130] which is also applicable to coreference. Nominal coreference is characterised by the use of a noun. It is further divided into pronominal and sortal coreference which are signified by a pronoun and a lexical noun phrase, respectively. Unlike nominal coreference, verbal coreference is indicated by verbs. Both nominal and verbal coreference can be broadly categorised by relation type as direct or indirect. In direct coreference, coreferring expressions are related by identity, synonymy or specialisation; in indirect coreference, they are related by associative relations such as meronymy or holonymy for nouns, and troponymy or entailment for verbs. Annotation of indirect coreference typically requires specialised domain knowledge. This

[^5]: http://labda.inf.uc3m.es/DrugDDI/DrugNerAr.html
categorisation allowed us to identify direct nominal coreference as the type of coreference that we prefer to focus on for this study. The discussions in the succeeding sections, therefore, will focus on this particular coreference type.

The coreference-annotated corpora reviewed can be grouped into four according to domain: general (MUC, GNOME and ACE 2005), molecular biology (MEDSTRACT and MEDCo), genomics (FlySlip and CRAFT) and pharmacology (DrugNerAr). In MUC and GNOME, all noun phrases in anaphoric relations were marked up, regardless of semantic type. In the ACE 2005 corpus, on the contrary, only mentions of certain types (person, organisation, location, geo-political entity, facility, vehicle and weapon) were annotated. MEDSTRACT and MEDCo have annotations for semantic types from the UMLS and GENIA ontologies, respectively. Each of FlySlip and DrugNerAr, in contrast, has annotations for only one semantic type: GGP and drug, respectively. CRAFT is unique amongst the biomedical corpora in this respect as its developers sought to annotate all coreferring expressions regardless of type.
<table>
<thead>
<tr>
<th>Corpus</th>
<th>Domain</th>
<th>Document Type/Size</th>
<th>Scheme</th>
<th>Coreference Types</th>
<th>Encoding/Format</th>
<th>IAA</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC 6/7</td>
<td>labour disputes, corporate succession/airplane crashes, aircraft launches</td>
<td>WSJ/30 NYT news articles</td>
<td>anaphor-antecedent pairs</td>
<td>direct nominal</td>
<td>in-line SGML</td>
<td>undisclosed</td>
<td>licensed</td>
</tr>
<tr>
<td>GNOME</td>
<td>museums, medicines, tutorial dialogues</td>
<td>descriptions, transcripts, leaflets</td>
<td>anaphor-antecedent pairs</td>
<td>direct &amp; indirect nominal</td>
<td>stand-off XML</td>
<td>79.4% overlap</td>
<td>licensed</td>
</tr>
<tr>
<td>ACE 2005</td>
<td>broadcast conversation, broadcast news, newsgroups, weblogs</td>
<td>599 articles, transcripts (English)</td>
<td>coreference chains</td>
<td>direct nominal &amp; verbal</td>
<td>stand-off XML</td>
<td>undisclosed</td>
<td>licensed</td>
</tr>
<tr>
<td>MEDSTRACT</td>
<td>molecular biology</td>
<td>100 abstracts</td>
<td>anaphor-antecedent pairs</td>
<td>direct nominal</td>
<td>in-line XML</td>
<td>undisclosed</td>
<td>public</td>
</tr>
<tr>
<td>MEDCo-A</td>
<td>human blood cell transcription factors</td>
<td>1,999 abstracts</td>
<td>anaphor-antecedent pairs</td>
<td>direct nominal</td>
<td>in-line XML</td>
<td>83% α</td>
<td>public</td>
</tr>
<tr>
<td>MEDCo-B</td>
<td>human blood cell transcription factors</td>
<td>43 full articles</td>
<td>anaphor-antecedent pairs</td>
<td>direct nominal</td>
<td>in-line XML</td>
<td>80.7% α</td>
<td>currently unavailable</td>
</tr>
<tr>
<td>FlySlip</td>
<td>fruit fly genomics</td>
<td>5 full articles</td>
<td>anaphor-antecedent pairs</td>
<td>direct &amp; indirect sortal</td>
<td>in-line XML</td>
<td>83% κ</td>
<td>public</td>
</tr>
<tr>
<td>CRAFT</td>
<td>mouse genomics</td>
<td>97 full articles</td>
<td>coreference chains</td>
<td>direct nominal &amp; verbal</td>
<td>stand-off XML</td>
<td>61.9% α</td>
<td>currently unavailable</td>
</tr>
<tr>
<td>DrugNerAr</td>
<td>drug-drug interactions</td>
<td>49 DrugBank texts</td>
<td>anaphor-antecedent pairs</td>
<td>direct nominal</td>
<td>stand-off XML</td>
<td>undisclosed</td>
<td>public</td>
</tr>
</tbody>
</table>
3.1. LITERATURE REVIEW

Previous studies have shown the advantages of utilising full articles instead of abstracts in the development of information extraction systems [131–133]. Furthermore, one study [134] demonstrated the need for processing full-text articles when identifying coreferent expressions pertaining to biomedical entities. The development of systems which can handle full articles, however, is largely reliant on the availability of learning and evaluation corpora comprised of coreference-annotated full texts. Currently, only one such corpus, FlySlip, is available for download.

CRAFT and ACE 2005 are the only corpora with annotations for verbal coreference; all the rest have annotations only for pronominal and/or sortal coreference. With respect to coreference types according to relation, GNOME and FlySlip are the only corpora with annotations for indirect coreference.

The annotations in most of the corpora are represented as anaphor-antecedent pairs, rather than as coreference chains (as in the case of ACE 2005 and CRAFT). This is because some of these corpora ( GNOME, MEDSTRACT, FlySlip, DrugNerAr) were developed with the intention of supporting anaphora resolution, rather than coreference resolution. However, they can still be adapted to support the latter, by the addition of links between unconnected pairs which refer to the same real-world entity. The MUC and MEDCo corpora, although developed as resources for coreference resolution tasks, were based on annotation schemes using pairwise linking. We believe that this is a reasonable approach, the first reason being the intuitiveness of linking several expressions two at a time (i.e., by looking at the closest preceding coreferring expression). Second is the ease with which most annotation tools can be adapted to support pairwise linking, whereas the formation of coreference chains during annotation would probably require a tool with a more complex user interface (e.g., one with drag-and-drop mechanisms). The only drawback with this approach is the requirement that pairs referring to the same entity are linked, to eventually allow their merger into one coreference chain. Even with such a pairwise linking scheme, storing coreferring expressions as
CHAPTER 3. COREFERENCE RESOLUTION

3.1.2 Approaches to coreference resolution

3.1.2.1 Classical Pronoun Resolution

Most of the classical approaches to coreference and anaphora resolution were focussed on the pronominal type. Hobbs’ method for anaphora resolution \[135\] is based on an algorithm that searches for an antecedent by traversing the surface parse tree of a sentence. In searching for noun phrases with the same number and gender as the anaphoric expression, a particular traversal order is followed.

Lappin and Leass formulated the Resolution of Anaphora Procedure (RAP) \[136\]. This algorithm is based on the concept of salience which can be computed from the full syntactic parse of a sentence. For each candidate antecedent in the discourse, salience weighting is done based on the salience factors which are satisfied. Although ranking based on grammatical roles was imposed on the salience factors, their values were arbitrarily assigned. An example of a salience factor is sentence recency which has the highest value amongst the factors and is satisfied only by all candidate antecedents in the current sentence. Subject and head noun emphases are the next highest-valued salience factors whilst indirect object emphasis has the lowest. RAP also includes an algorithm for detecting pleonastic pronouns which are considered as semantically empty. A sentence with a pleonastic \textit{it} is shown in Example 3.4, where the pronoun \textit{it} does not refer to any entity. To determine if a pronoun is pleonastic, certain rules which check for the appearance of the pronoun with modal adjectives (e.g., desirable, convenient, important) or cognitive verbs (e.g., past tense forms such as recommended, anticipated and assumed) are applied. Pronouns captured by these rules were not
considered anaphoric expressions.

**Example 3.4.** It is advisable to have a timeline.

Kennedy and Boguraev [137] formulated a modified version of Lappin and Le- ass’ RAP in which only part-of-speech tags enriched with annotations of grammatical function are required instead of full syntactic parses.

Mitkov’s algorithm [138] provided an even less computationally expensive method for pronoun resolution. It required only part-of-speech tagging of sentences, rather than syntactic parses. The algorithm is largely based on indicators which were used for tracking antecedents. Each candidate antecedent is assigned a score for each of the indicators which include, among others, salience features such as definiteness, lexical reiteration and givenness.

### 3.1.2.2 Pronoun Resolution Based on the Centering Theory

Centering Theory [139] is a theory of local coherence of discourse which has been used by several research groups for anaphora and coreference resolution. Most of the ideas on which it was based on had been laid out as early as 1977, when Barbara Grosz introduced the concept of centre of attention. This, according to her, is a set of entities on which the discourse participants’ attention is centred at a given time. The set of centred entities can be useful in anaphora or coreference resolution as it is in this set where the antecedent is expectedly located. However, Grosz did not describe in detail how the centre of attention is determined. In 1979, Candace Sidner defined specific ways by which the centre can be identified. She defined structures such as discourse focus, actor focus and a set called potential foci, as well as algorithms on how these structures are assigned values. She stipulated that the antecedent corresponds to either the discourse focus or the actor focus; in cases where both have been eliminated as the possible antecedent by one of the criteria, an element from the set of potential foci is
In 1986, Grosz and Sidner defined discourse structure as consisting of three components: a linguistic structure, an intentional structure and an attentional state. Linguistic structure accounts for the division of the discourse into discourse segments, each of which has a discourse segment purpose. Intentional structure, the second component, corresponds to the discourse participant’s intentions. Finally, attentional state represents the centre of attention of the discourse participant as the discourse unfolds.

Centering Theory was then formalised as a framework for modelling the dynamic attentional state within a discourse segment. The theory presented the relationships amongst three elements: the focus of attention, the choice of expression and the degree of coherence. It is based on the concept of centres which are realised in an utterance. For each utterance, there is exactly one backward-looking centre and a set of forward-looking centres; the backward-looking centre of the current utterance is the highest-ranked forward-looking centre of the previous utterance, where ranking is usually based on grammatical role (e.g., a centre acting as subject has higher ranking than one acting as object).

Although it was developed as a theory of local coherence, the Centering Theory has been used by some research groups as a framework for anaphora or coreference resolution. Brennan et al. of the Hewlett-Packard Laboratories, for example, extended the theory by adding more transitions (the links between utterances), and applied it to pronoun resolution. The theory can be used for anaphora or coreference resolution in the context of the pronoun realisation rule which implies that if an utterance contains a pronoun, its backward-looking centre must also be a pronoun; matching this backward-looking center with the highest-ranked forward looking centre of the previous utterance results in identifying the antecedent.
3.1.2.3 Coreference Resolution Paradigms

In this section, we describe various notable approaches to both general- and biomedical-domain coreference resolution, categorised according to paradigm. We follow the paradigms identified by Rahman and Ng [143] but refer to them with names which we believe are more intuitive. Example 3.5 will be used to illustrate how the mention These will be resolved according to each paradigm.

**Example 3.5.** Indeed, we have recently identified several cyanobacterial compounds that are potent [protease] inhibitors, such as [grassystatins A-C] that selectively inhibit [cathepsin E] and [lyngbyastatins 4-7] [which] inhibit [porcine pancreatic elastase]. [The latter group] are part of a class prolifically produced by marine and aquatic cyanobacteria, containing [3-amino-6-hydroxy-2-piperidone] ([Ahp]) as part of a six-unit cyclic core with a pendant side chain. [These] generally inhibit certain serine proteases.

All mentions appear in square brackets. Those which are coreferent with each other are underlined in the same style (i.e., those with single underlines refer to one entity, and those with double underlines refer to another). Except for These which is our active mention (i.e., the mention that we currently want to resolve), mentions which are not underlined are not coreferent with any other mention.

**Mention pair classification (MPC).** Under this paradigm, pairs of mentions are classified as coreferent or not. In resolving the mention These in Example 3.5, a system will first form mention pairs between the active mention and the mentions preceding it, e.g., <These, Ahp>, <These, 3-amino-6-hydroxy-2-piperidone>, <These, The latter group>. Each pair will then be analysed by a binary classifier, which will predict whether the two mentions are coreferent or not. Ideally, for instance, only the pair <These, The latter group> will be classified as positive (coreferent) by a coreference
resolution system. If none of the mention pairs was detected as coreferent, the active mention is considered to be discourse-new. Otherwise, the active mention is linked to the preceding mention in the positively identified pair. A post-processing step of merging the linked pairs together into a chain is then performed.

Soon et al. applied this approach to nominal coreference resolution, employing a machine learning-based method \[144\]. Based on the annotations in the MUC-6 and -7 corpora, positive training instances were generated by forming pairs between the noun phrases (NP) of a gold standard coreference chain. Negative examples were formed by taking the markables between two coreferring noun phrases and then pairing each with the latter noun phrase. A model was trained on these noun phrase pairs with the C5 decision tree algorithm, based on a feature set consisting of the following: sentence distance, pronoun match, string match, definite NP match, demonstrative NP match, number agreement, semantic type agreement, gender agreement, being both proper names, one being an appositive of the other and one being an alias of the other. To identify coreference chains in unseen text, pairs of markables (i.e., phrases captured by their noun phrase extraction module) are formed in the following manner: (1) a candidate anaphor (i.e., a noun phrase in the document except the first occurring one) is paired with each noun phrase preceding it, starting from the closest one; (2) each pair is presented to the decision tree which will classify the mentions in the pair as coreferent or not; (3) if they are classified as coreferent, the algorithm proceeds to the next candidate anaphor; otherwise, a new pair is formed with the next preceding noun phrase, and presented to the classifier. After processing all candidate anaphors, the procedure merges positively classified pairs with common member noun phrases.

Most of the reported approaches to biomedical coreference resolution were based on mention pair classification. Gasperin and Briscoe, for example, used a probabilistic model based on the naïve Bayes approach to locate the antecedents of non-pronominal
3.1. LITERATURE REVIEW

anaphors and to further classify the anaphoric relations as either coreferent or associative [145]. A coreferent relation exists if the anaphor and its antecedent are identical entities; on the contrary, an associative relation holds if the two are not identical but the latter helps define the former. The associative types are further divided into set-member relations and biotype relations which hold between biomedical entities of interest (e.g., gene, gene product, gene subtype, gene part, product part, sequence supertype and gene variant). Aside from the basic features such as string similarity, semantic type agreement, number agreement and distance, a biotype feature determining whether the expression corresponds to a gene or product was utilised in training the model.

Other approaches based on mention pair classification employed heuristics. Castaño et. al. [146] developed an anaphora resolution engine for pronominal and sortal anaphors. Candidate anaphors were detected using hand-crafted rules based on the syntactic and semantic types of an anaphor. They were restricted, for example, to third-person pronouns and expressions which correspond to any of the target UMLS semantic types (e.g., amino acid, protein, cell, enzyme). The antecedent of each anaphor is then selected based on a salience score which measures the compatibility of a candidate antecedent with the anaphor based on features such as string similarity, and person, number and semantic type agreement.

Liang and Lin [147] also computed for the value of salience in locating antecedents, with semantic features enhanced by exploiting external resources such as UMLS, PubMed, WordNet and the GENIA corpus. The UMLS was used offline to generate key lexicons for each semantic type available; these lexicons were used to predict the semantic types of chunks during sortal anaphora recognition and antecedent selection. If these key lexicons fail to predict a chunk’s semantic type, a lookup in PubMed is made. Furthermore, the other types of relationships curated in WordNet (e.g., hypernymy, hyponymy, holonymy, meronymy) are checked to determine semantic type agreement. In resolving pronominal anaphora, the semantic type is determined by
matching against a set of subject-action/action-object (SA/OA) patterns which were collected from the GENIA corpus. Evaluation was done on two data sets: the MED-STRACT corpus and a set of 100 randomly selected abstracts from MEDLINE.

Kim and Park [148] used a set of rules for resolving pronominal, sortal and biological interaction anaphoric expressions to their antecedents. After locating pronouns, definite noun phrases and biological interactions with missing arguments, rules and patterns were applied to locate their corresponding antecedents. These patterns were manually defined based on observations on how pronouns appear relative to their antecedents, as well as on the Centering Theory. For sortal anaphora resolution, a scoring mechanism similar to Castaño’s approach [146] was used. Their system, called BioAR, was evaluated using 120 biological interactions extracted from MEDLINE.

DrugNerAr [118] was developed for use in a drug-drug interaction extraction system. Hand-crafted patterns and rules were used to detect anaphoric expressions pertaining to drugs, both pronominal and sortal. Similar to BioAR, DrugNerAr identifies the antecedent of each anaphor by applying the Centering Theory, except for relative, reflexive and possessive pronominal anaphora whose antecedents were always found to be the closest noun phrase that matches the number of the pronoun. Evaluation of the approach was done on a manually annotated corpus consisting of 49 drug interaction documents from DrugBank.

**Mention pair ranking (MPR).** This paradigm is similar to mention pair classification in that mention pairs between the active mention and its candidate antecedents are also initially formed. However, instead of performing classification of each pair as coreferent or not, a system will induce a ranking over the pairs, with the top-ranked one having the highest probability of being a coreferring mention. As a first step, a system will determine whether the active mention is discourse-new or anaphoric (e.g.,
with the use of a binary classifier). Whilst the former case does not require any further step, the latter implies that the interpretation of the active mention is dependent on another mention and hence requires resolution. In the case of the latter, the active mention *These*, for instance, is paired with preceding mentions one at a time, e.g., *<These, Ahp>***, *<These, 3-amino-6-hydroxy-2-piperidone>***, *<These, The latter group>*. The pairs will be then be ranked. Given our running example, an ideal coreference resolution system would give the pair *<These, The latter group>* the highest ranking. The active mention is thus linked to the preceding mention in the pair with the highest ranking. Like in mention pair classification, a post-processing step for merging linked pairs together into a chain is necessary.

The approach by Denis and Baldridge [149] is an implementation of mention ranking. After the generation of positive and negative training pairs, a statistical model was trained using a log-linear ranking loss function. The function incorporates features of the candidate antecedent (e.g., being a proper name or pronoun, subjecthood), agreement between the anaphor and candidate antecedent (e.g., gender, number, person, semantic type agreement) and string similarity. Presented with anaphor-antecedent pairs in test documents, the ranking model selects the pair with the highest probability of being coreferent. The method was evaluated on the ACE 2005 corpus.

Nguyen and Kim [150] used a maximum entropy ranker model in their biomedical pronoun resolution system. A combination of morphological, syntactic and semantic features was extracted from three data sets from different domains: MUC-7, ACE 2003 and the MEDCo corpus of abstracts. The contribution of each feature was systematically measured; it was observed that the semantic features contributed the most to the performance on the MEDCo abstracts corpus. Their system was evaluated on MEDCo using a criterion that only requires that an anaphor be matched to any expression in its antecedent’s coreference chain (rather than strictly to the antecedent).

Similarly, Torii and Vijay-Shanker [151] took an approach based on mention pair
ranking for the resolution of sortal anaphora. Specifically, they trained a maximum entropy model on features such as basic syntactic, morphological and semantic features, with the addition of their proposed highlighting features. These highlighting features were measured based on the observation that there are only a few entities which authors of biomedical articles focus on, and that these entities will most likely be repeated in the discourse with the use of anaphors. Included under this new type of features are the study and we features which are enabled when phrases such as In this study... or We investigate... are found in the same sentence as a term, implying that the author will most likely be focussing on the term in the discourse. A corpus consisting of MEDLINE abstracts (309 for training and 297 for testing) was manually annotated and used for the evaluation of their method.

**Mention-chain classification (MCC).** Similar to mention pair classification, this paradigm also employs a binary classifier to predict whether an instance is characterised by coreference or not. Instead of a mention pair, however, an instance is composed of the active mention and a set of preceding mentions which form a partial coreference chain. To illustrate the process, we take note of the partial coreference chains which will have already been formed from Example 3.5:

\[
\begin{align*}
  c_1 : & \{\text{protease}\} \\
  c_2 : & \{\text{grassystatins A-C}\} \\
  c_3 : & \{\text{cathepsin E}\} \\
  c_4 : & \{\text{Iyngbystatins 4-7}, \text{which}, \text{The latter group}\} \\
  c_5 : & \{\text{porcine pancreatic elastase}\} \\
  c_6 : & \{\text{3-amino-6-hydroxy-2-piperidone}, \text{Ahp}\}
\end{align*}
\]

First, the active mention will be paired with partial chains, giving, for example, <These, c_6>, <These, c_5> and <These, c_4>. For each pair, a binary classifier will detect
whether the active mention is coreferent with the partial chain, based on attributes of its member mentions. Given our running example, an ideal system will classify $<\text{These}, c_4>$ as coreferent and the other two as non-coreferent. After classifying a pair as coreferent, the active mention will be added to the partial chain. If each of the mention-chain pairs was classified as non-coreferent, the active mention is not resolved to any chain.

The approach taken by Yang et al. [152] is based on mention-chain classification. An instance is composed of the active mention and a partial coreference chain. A C5 decision tree-based classifier was trained on learning instances from 70 of the MEDCo abstracts. It was then evaluated on 30 annotated abstracts, also from MEDCo. The F-score they obtained is higher by a margin of almost 4% over a baseline system implemented as mention pair classification.

**Mention-chain ranking (MCR).** Mention-chain ranking combines characteristics of mention pair ranking and mention-chain classification. It is similar to mention pair ranking in that it requires the preliminary step of anaphoricity detection and it induces a ranking amongst instances. However, rather than forming a pair consisting of the active mention and a preceding mention, it creates instances by pairing the active mention with a partial coreference chain, like in mention-chain classification. After determining, for example, that the active mention *These* is anaphoric (as opposed to discourse-new) and hence should be resolved, a system will form the following mention-chain pairs: $<\text{These}, c_6>$, $<\text{These}, c_5>$ and $<\text{These}, c_4>$. These pairs will be ordered according to a ranking algorithm, with the “most coreferent” mention-chain pair getting the highest rank. The active mention will be added to the chain in the top-ranked instance.

The solution to coreference resolution proposed by Rahman and Ng [143] is based on this paradigm. Instances were created by pairing the active mention with each of the partial chains which have been formed so far. Instead of giving positive or
negative labels to instances, the most preferable (coreferent) instance is assigned a higher rank than the rest of the instances. By pairing the active mention with each of the partial clusters, a set of instances is generated and ranked by a model trained using support vector machines. The active mention is added to the chain of the highest-ranked instance. If several instances were ranked highest, the instance with the chain containing the closest mention to the active mention is chosen.

**Global decision.** Recently, we have seen the emergence of approaches to coreference resolution which group mentions into their respective chains in one step. These are usually implemented as unsupervised graph partitioning algorithms, where the document is represented as a graph whose nodes correspond to the mentions in the document. The main idea is to induce a clustering of the nodes in one step, i.e., without the intermediate steps of classification or ranking.

Cai and Strube \cite{153} took this approach by representing a document as an undirected weighted hypergraph, a graph in which an edge, known as a hyperedge, is composed of any number of vertices. In their representation, the vertices of the hypergraph are mentions in the document, whilst the weighted hyperedges connecting them are the features. The hypergraph is partitioned into subhypergraphs using a spectral clustering algorithm. Each resulting subhypergraph corresponds to a coreference chain; the contained nodes are the coreferent mentions. Their method was evaluated on the ACE 2005 corpus.

To represent a document, Martschat \cite{154} used a multigraph, i.e., graphs wherein two vertices can be connected by multiple edges with different weights. Similarly, the vertices of the multigraph represent the mentions in the document, whilst the weighted edges are the features. The sequence of mentions as they appear in text is also incorporated into the multigraph by the directionality of the edges. Partitioning of the multigraph is performed using an unsupervised method, specifically, \emph{k}-nearest-neighbour
clustering with \( k = 1 \). The developers reported results on the CoNLL 2012 shared task data.

3.1.2.4 Comparison

Presented in Table 3.2 is a summary of the approaches to biomedical coreference resolution we have reviewed, grouped by paradigm. Where possible, we provide the performance of each approach for the reader’s reference. However, except for Castaño et al. and Liang and Lin’s approaches, they are not comparable as they were evaluated on different corpora, as indicated in the table.

Regardless of paradigm, the methods can be broadly categorised into heuristics-based and machine learning-based approaches. Amongst the heuristics-based ones, some similarities in their key ideas can be observed. These include the implementation of a scoring system based on salience, inspiration from the Centering Theory and agreement between mentions in terms of syntax, semantics and morphology. We can also observe the significant performance improvement obtained by Liang and Lin’s method over that of Castaño et al., which can be attributed to their extensive use of external semantic resources. Similarly, the machine learning-based approaches (e.g., decision tree, maximum entropy and naïve-Bayes models) utilised rich syntactic, morphological and semantic features.

It is noticeable that most of the approaches fall under the mention pair classification paradigm, with very few exploring mention pair ranking and mention-chain classification. Expectedly, the mention-chain ranking paradigm has not yet been explored for biomedical coreference resolution considering that it was proposed only recently by Rahman and Ng [143]. Whilst the mention pair classification paradigm is most commonly used, the other three paradigms have strengths or advantages over MPC which are worth noting. As mention pair ranking imposes a ranking over several possible
referents, it is capable of capturing the competition amongst several candidates and 
returns the most compatible one. Mention-chain classification also has an advantage over 
mention pair classification in its capability to match the active mention against a par-
tial coreference chain instead of just another mention. This allows for a more expres-
sive instance representation, as features of a set of mentions capture more information 
about the candidate referent than features of a single mention. Finally, mention-chain 
ranking combines the strengths of both mention pair ranking and mention-chain clas-
sification. It accounts for more information about each candidate referent by matching 
the active mention against a partial coreference chain and, at the same time, captures 
the competition amongst several candidates by inducing a ranking over them.
Table 3.2: Comparison of Approaches to Biomedical Coreference Resolution

<table>
<thead>
<tr>
<th>Approach</th>
<th>Proponents or System</th>
<th>Key Ideas and Resources</th>
<th>Eval. Corpus</th>
<th>Pronominal</th>
<th>Sortal</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC</td>
<td>Gasperin &amp; Briscoe</td>
<td>naïve-Bayes probabilistic model, biotype features</td>
<td>FlySlip</td>
<td>N/A</td>
<td>68.3% F₁</td>
</tr>
<tr>
<td></td>
<td>Castaño et al.</td>
<td>salience-based scoring, syntactic, semantic, morphological agreement</td>
<td>MEDSTRACT</td>
<td>75.23% F₁</td>
<td>74.5% F₁</td>
</tr>
<tr>
<td></td>
<td>Liang &amp; Lin</td>
<td>salience-based scoring, semantic agreement with UMLS, PubMed, GENIA, WordNet</td>
<td>MEDSTRACT</td>
<td>92.31% F₁</td>
<td>78.26% F₁</td>
</tr>
<tr>
<td></td>
<td>BioAR</td>
<td>Centering Theory and salience-based scoring</td>
<td>120 biological interactions from MEDLINE</td>
<td>64.32% F₁</td>
<td>61.56% F₁</td>
</tr>
<tr>
<td></td>
<td>DrugNerAr</td>
<td>Centering Theory and rules</td>
<td>DDI Corpus subset of 49 documents</td>
<td>91% F₁</td>
<td>56% F₁</td>
</tr>
<tr>
<td>MPR</td>
<td>Nguyen &amp; Kim</td>
<td>maximum entropy ranking model</td>
<td>MEDCo abstracts</td>
<td>80.85% success rate</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Torii &amp; Vijay-Shanker</td>
<td>maximum entropy ranking model, highlighting features</td>
<td>297 MEDLINE abstracts</td>
<td>N/A</td>
<td>74.2% F₁</td>
</tr>
<tr>
<td>MCC</td>
<td>Yang et al.</td>
<td>C5 decision tree classifier</td>
<td>MEDCo subset of 100 abstracts</td>
<td>81.8% F₁</td>
<td></td>
</tr>
</tbody>
</table>
3.2 Corpus Development

As mentioned in Section 3.1, previous work has demonstrated the advantages of utilising full-text articles in information extraction tasks [131-133], especially in coreference resolution [134]. We have determined from our review of coreference-annotated corpora that currently, FlySlip is the only publicly available corpus containing full articles. Since the documents in FlySlip are not suitable for our study of coreference resolution in pharmaceutical chemistry literature, we decided to develop our own corpus of full-text articles from a chemical subdomain, before proceeding with the development of our own methods for coreference resolution. We shall hereafter refer to this document collection as the HANAPIN corpus. Three graduate students, all having a bachelor’s degree in chemistry, were employed and trained as annotators. They provided the domain expertise necessary in the design and development of this corpus.

3.2.1 Composition of Corpus Documents

As we intend to make our corpus publicly available, we gathered from the PubMed Central Open Access repository documents relevant to our domain of interest, i.e., pharmaceutical chemistry. Narrowing down our journal choices to those which focus on drug and lead compound development, we were left with the Marine Drugs journal. The said journal covers subject areas such as marine natural products, medicine analysis, pharmacology, pharmaceutical biology, drug development and biotechnology, among many others. From all of its articles from 2003 to 2009, we randomly selected twenty (20) which seemed to be a reasonable number considering that: (1) only five months were allocated for the annotation of the corpus, and (2) we were not particularly concerned about the small number of documents as a previous study

---

6http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist
7http://www.mdpi.com/journal/marinedrugs
3.2. CORPUS DEVELOPMENT

Table 3.3: Coreference Types with Examples

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>pronominal</td>
<td>demonstrative</td>
<td><em>this, that, these, those</em></td>
</tr>
<tr>
<td></td>
<td>personal</td>
<td><em>it, they, its, their, theirs</em></td>
</tr>
<tr>
<td></td>
<td>indefinite</td>
<td><em>another, few, other, some, all, any</em></td>
</tr>
<tr>
<td></td>
<td>distributive</td>
<td><em>both, such, each, either, neither</em></td>
</tr>
<tr>
<td></td>
<td>relative</td>
<td><em>which, that, whose</em></td>
</tr>
<tr>
<td>sortal</td>
<td>definite</td>
<td><em>the loihichelins</em></td>
</tr>
<tr>
<td></td>
<td>indefinite</td>
<td><em>an alkaloid, a mycalamide</em></td>
</tr>
<tr>
<td></td>
<td>demonstrative</td>
<td><em>this metabolite, these compounds</em></td>
</tr>
<tr>
<td></td>
<td>distributive</td>
<td><em>both compounds</em></td>
</tr>
<tr>
<td></td>
<td>predicate nominative</td>
<td>“<em>Galactans are polysaccharides...</em>”</td>
</tr>
<tr>
<td></td>
<td>appositive</td>
<td>“<em>Radiosumin, an N-methyl dipeptide...</em>”</td>
</tr>
<tr>
<td>numerical</td>
<td>N.A.</td>
<td>“<em>The structures of 1 and 2...</em>”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“<em>Compounds 1-3 inhibit...</em>”</td>
</tr>
<tr>
<td>abbreviation</td>
<td>N.A.</td>
<td>“<em>...as a membrane type 1 matrix metalloproteinase (MT1-MMP) inhibitor.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“<em>Compound 1 inhibited MT1-MMP with...</em>”</td>
</tr>
</tbody>
</table>

on biomedical corpora usage has shown that corpus size does not determine usability [155]. The experimental sections of the articles were removed as they contain only detailed descriptions of the methods carried out by the authors. According to a study [131], these usually include technical data, instruments and measurements – details which are not relevant to the information we aim to extract, i.e., drug-target interactions. The corpus contains a total of 1,027 sentences (with a total of 27,358 words).

3.2.2 Annotation Scope

As our focus in this study is on direct nominal coreference, the verbal and indirect coreference types were not taken into account. Only nominal and direct coreferring expressions were covered by the annotations in this corpus. After a preliminary phase in which the annotators were asked to observe the types of nominal coreferring expressions in five (5) documents, four upper-level nominal categories were identified:
pronominal, sortal, numerical and abbreviation. Table 3.3 presents the subtypes of sortal and pronominal coreference, as well as examples for all types. For examples of sentences containing such expressions, the reader is referred to the complete listing of the annotation guidelines, provided in Appendix B.2.

3.2.2.1 Pronominal Coreference

This type of coreference is characterised by a pronoun referring to a noun phrase. The pronominal mention is used as a substitute to a noun. We have further identified the following subtypes: demonstrative, personal, indefinite, distributive and relative.

3.2.2.2 Sortal coreference

Also referred to as lexical noun phrase coreference, sortal coreference is characterised by a noun phrase consisting of a head noun and its modifiers. The subtypes of sortal coreference which have been identified include: definite, indefinite, demonstrative, distributive, predicate nominative and appositive.

3.2.2.3 Numerical coreference

In scientific literature from the chemical domain, a number is conventionally used to refer to a chemical entity which was introduced using the same number. Often, a range of numbers is also used to refer to a number of compounds previously mentioned.

3.2.2.4 Abbreviation

In annotating the HANAPIN corpus, abbreviations were also considered as co-referring expressions. The distinction between them and the other nominal coreference types was made in order to make their annotations useful to developers of abbreviation recognition algorithms as well.
3.2. CORPUS DEVELOPMENT

3.2.3 Annotation Scheme and Procedure

Considering the points we have presented in Section 3.1, we adapted a pairwise linking scheme for annotating coreferring expressions in our corpus. We had access to only one annotation tool that supports such linking, namely, the XML Concordancer (XConc) Suite. As the MEDCo corpus was developed using the same tool and based on the same pairwise linking scheme, we adapted its annotation scheme which was realised as a Document Type Definition (DTD) file. The coreference types were modified and extended following our own typology. The final version of the DTD file is provided in Appendix B.3 for the reader’s reference.

As its name implies, the XML Concordancer tool expects as input XML files. Our full articles, downloaded as MEDLINE XML files, were first processed to remove all XML tags in order to retain only the document text. The LingPipe MEDLINE sentence model was then employed to split text into sentences, the basic units required by XConc. The resulting sentences were then programatically encoded into XConc-compliant XML format. All annotations added to a document were automatically encoded by XConc into the corresponding XML file as in-line XML elements.

3.2.3.1 Annotation of named entities

After identifying the most prevalent semantic classes in our corpus, the annotators were asked to annotate and categorise any named entity (NE) encountered in the documents according to the following types:

---

http://www.nactem.ac.uk/genia/tools/xconc
• chemical compound

• organism

• drug effect

• disease

• drug target (further categorised into: protein, enzyme, nucleic acid, tissue, cell, cell component, cell line, pathogen)

For each markable, the annotator creates a term element which is assigned an identifier and one of the semantic types above. The scheme supports the annotation of embedded named entities, as well as NEs in discontinuous text regions. The former entails placing a term element within an encterm element whilst the latter is done by dividing the discontinuous text into the individual NEs and annotating each one in the same manner as an ordinary term element. The individual term elements are then grouped together by a cons element. Shown in Figure 3.1 is an example, as displayed by XConc, of an annotated discontinuous named entity (jaspamide R) appearing as one of two coordinated NEs. The coordinated expression jaspamide Q and R was segmented into the individual entities jaspamide Q and R which were grouped together into one cons element C5. The fragment common to both entities jaspamide is then placed in a frag element.

3.2.3.2 Annotation of co-referring expressions

An annotator proceeds to the annotation of coreferring expressions after annotating all entities within a document. If an expression was found to be coreferring with another named entity, the annotator assigns the identifier of the latter as the value of the idref attribute of the former. If the referring expression, however, is a noun phrase rather than an entity that was previously annotated during NE annotation, it is marked as a
ref element and then linked to its referent. Annotators delimit these expressions by including the necessary modifiers of the coreferring element (e.g., the new in the new jaspamide derivatives). A coreference type which could be any of pronominal, numerical, abbreviation, and sortal (further categorised into definite, indefinite, demonstrative, distributive, predicate nominative and appositive) is also assigned as the value of the type attribute of each link created. We decided not to further divide pronominal coreference into its subtypes as it was observed during the preliminary annotation phase that pronominal expressions constitute a small fraction of the mentions in the corpus. Figure 3.1 shows coreferring expressions (connected by arrows) linked by the mechanism just described.

Listed below are some of the main points of the annotation guidelines, provided in full in Appendix B.2:

- The more specific one between two coreferring expressions is considered as the referent. Consequently, an expression occurring later than the referring expression in the text can be marked up as the referent. For example, R30: the new
natural products is the referring expression and C5: jaspamide Q and R is the referent in Figure 3.1

- A referring expression may be linked to multiple referents.
- In cases where there are multiple mentions which qualify as referent of a referring expression, the closest one should be chosen.
- To allow for the eventual formation of coreference chains, pairs referring to the same entity need to be linked to facilitate merging.
- There are cases when more than one coreference type applies. For example, in Figure 3.1, the new natural products is both an appositive and a definite noun phrase. In such cases, the appositive and predicate nominative types were given higher precedence over the other sortal types.

Provided in Figure 3.2 is the XML code for the example in Figure 3.1. From the annotations for this example, the following coreference chains are obtained.

\[
c_1 : \{R30:[the new natural products], C5:[jaspamide Q and R], R10:[the new jaspamide derivatives], R11:[which], R12:[both]\}
\]

\[
c_2 : \{T66:[jaspamide Q], R34:[2]\}
\]

\[
c_3 : \{T67:[jaspamide R], R35:[3]\}
\]

\[
c_4 : \{T70:[jaspamide], R36:[1]\}
\]

The complete annotation guidelines will be publicly released together with the annotated corpus. It is worth noting, however, that a reader for the HANAPIN corpus is already provided in the stand-alone, interoperable text mining platform U-Compare, which is based on the Unstructured Information Management Architecture (UIMA) standard [156].
3.2. CORPUS DEVELOPMENT

3.2.4 Results and Discussion

The three annotators were asked to complete the coreference annotations within five months. Bi-weekly meetings were held to address questions and issues which could not be addressed or resolved by means of an on-line project forum.

3.2.4.1 Frequencies

The frequency of each annotation type was calculated over the annotations (Figure 3.3). For each type, we obtained the average over the annotations from the three coders.

There is a total of 395 coreference chains (not including singleton chains or those with only one mention) in the entire corpus. The coreference chains are of the following semantic types: chemical compounds (70.89%), drug targets (12.66%, accounting for proteins, cell lines, pathogens, enzymes, cells, cell parts, nucleic acids and tissues), organisms (9.87%), drug effects (3.29%) and diseases (3.29%). Among the drug targets, the most prevalent ones are proteins, cell lines and pathogens.

A total of 760 coreference links have been found in the corpus. The most common among the types is the numerical one (46%), followed by the sortal type (33%, accounting for the definite, indefinite, demonstrative, appositive, predicate nominative
and distributive types). Less common are annotations falling under the pronominal type (11%) and abbreviation (10%). Amongst the sortal coreferring expressions, the most common are the definite and indefinite types, followed by the demonstrative type.

### 3.2.4.2 Corpus Reliability

Following Passoneau’s proposed method for computing reliability for coreference annotation [157], we computed for the reliability of the corpus in terms of Krippendorff’s alpha, a coefficient of agreement that allows for partial disagreement with the use of a distance metric based on the similarity between coreference chains. Passoneau’s first proposed distance metric \(d_P\) assigns 0 for identity, 0.33 for subsumption, 0.67 for intersection and 1 for disjunction. There are, however, alternative distance metrics that consider the sizes of the coreference chains, such as Jaccard’s coefficient of community \(d_J\) and Dice’s coincidence index \(d_D\) which can be computed by Equations 3.1 and 3.2, respectively [158]:

\[
d_J = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{3.1}
\]

\[
d_D = 1 - \frac{2|A \cap B|}{|A| + |B|} \tag{3.2}
\]

A new distance metric called Measuring Agreement on Set-valued Items (MASI) was then later proposed by Passoneau. It is obtained by getting the product of the original distance metric \(d_P\) and Jaccard’s coefficient \(d_J\).

Initially using Passoneau’s first proposed distance metric \(d_P\) in computing for Krippendorff’s alpha, we obtained an average of 75% over all documents in the HANAPIN corpus. Computing for alpha using the MASI distance metric gives 84%. Though there is no value of alpha that has been established to be an absolute indication of high agreement, previous works cited by Krippendorff have shown that values of alpha less
than 67% indicate unreliability \[159\]. We can therefore regard the obtained values of alpha as satisfactory.

3.3 Methodology

To the best of our knowledge, this study of coreference resolution in a chemical domain is the first of its kind. As such, we decided to explore each of the coreference resolution paradigms presented in Section 3.1.2 focusing only on those which can be realised using supervised methods. Taking inspiration from the work of Rahman and Ng \[143\], we implemented each of the paradigms and made use of classification and ranking models. Specifically, we employed the support vector machines (SVMs) learning algorithm, both for the classification and ranking tasks.

3.3.1 Support vector machines (SVMs)

Initially proposed as a binary classifier, SVMs learn a decision function that acts as a hyperplane separating positive \(n\)-dimensional data points from negative ones \[160, 161\]. The algorithm is given \(l\) examples; each example is represented as \((x_i, y_i)\) where \(x_i \in \mathbb{R}^n\) and \(y_i \in \{-1, 1\}\), corresponding to the label of the example. We want to find the unique hyperplane that separates the positive from the negative examples whilst maximising the margin, i.e., the distance between the hyperplane and the closest data point from either classes. We take into consideration that each hyperplane can be expressed as Equation 3.3 where \(\mathbf{w}\) is a vector and \(b\) is a constant.

\[
\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3.3}
\]

We aim to find a decision function \(f(\mathbf{x})\) as specified by Equation 3.4.
This function has to satisfy the following two properties.

1. that its value should be equivalent to the label of the training instance (Equation 3.5), and
   \[ \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) = y_i \]  
   (3.5)

2. that the data point closest to the hyperplane from either side should be “away” by a distance of at least 1 (Equations 3.6 and 3.7)
   \[ \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \text{ when } y_i = +1 \]  
   (3.6)
   \[ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \text{ when } y_i = -1 \]  
   (3.7)

   or simply:
   \[ y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \forall i \]  
   (3.8)

As previously mentioned, it is desirable to maximise the distance \( d \) to the closest data points on either side of the hyperplane, which is computed by means of Equation 3.9.

\[ d((\mathbf{w}, b), \mathbf{x}_i) = \frac{y_i(\mathbf{x}_i \cdot \mathbf{w} + b)}{||\mathbf{w}||} \geq \frac{1}{||\mathbf{w}||} \]  
(3.9)

Maximal distance is obtained by minimising \( ||\mathbf{w}|| \) which is facilitated by the use of Lagrange multipliers. The solution is eventually given by Equation 3.10.
3.3. METHODOLOGY

\[ W(\alpha) = \arg\min \left( \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{i=1}^{l} \alpha_i \right) \]  

(3.10)

with the constraints

\[ 0 \leq \alpha_i \leq C \forall i \]  

(3.11)

and

\[ \sum_{i=1}^{l} y_i \alpha_i = 0 \]  

(3.12)

where \( \alpha \) corresponds to the vector of \( l \) Lagrange multipliers to be computed and \( C \) is a constant. In many cases, most of the \( \alpha_i \) found by quadratic programming has a value of 0. The ones where \( \alpha > 0 \) are called support vectors and are the only ones which define the optimal hyperplane. This hyperplane can also be adjusted by tuning the constant \( C \); giving it a lower value results in a flexible hyperplane, allowing for misclassification of some of the data points whilst minimising the margin error, i.e., attempting to keep the value of \( y_i (x_i \cdot w + b) \) as close to 1 as possible.

The support vector machines algorithm can also be adapted to perform ranking instead of binary classification [162]. In this case, the training data consists of \( n \) queries. Each query \( q \) is comprised of several data points over which a ranking \( r^* \) is defined. The training set can then be represented as \( (q_1, r^*_1), (q_2, r^*_2), \ldots, (q_n, r^*_n) \). Instead of a decision function (as in the case of classification), the algorithm learns a ranking function \( f \) that maximises \( \tau \) in Equation (3.13)

\[ \tau(f) = \frac{1}{n} \sum_{i=1}^{n} \tau(r_{f(q)}, r^*_i) \]  

(3.13)

Each data point is a vector representing the features of that instance. In the next section, we discuss in detail each of the features we employed.
CHAPTER 3. COREFERENCE RESOLUTION

3.3.2 Feature set

In this section, we present a discussion of the different features which were used to encode instances presented to the SVM classification and ranking algorithms. We begin by giving a description of the features we adopted from the work of Rahman and Ng on coreference resolution for the general domain [143], followed by an overview of our own proposed chemistry-specific features. Each instance, whether a mention pair or a mention-chain pair, was represented with features describing a preceding mention \( m_p \), the mention being currently resolved or active mention \( m_a \), and the relation between the two.

3.3.2.1 Default features

Features describing a preceding mention \( m_p \).

- Is a pronoun: Returns true if \( m_p \) is a pronoun. Although part-of-speech (POS) taggers distinguish personal, possessive and \( wh \)-pronouns (e.g., which, whose) from other token types, a list of more complete pronouns which includes other types (e.g., demonstrative, distributive pronouns) was compiled as a resource for determining the value of this feature (provided in Appendix C.1).

- Is a nested noun phrase: Determines whether \( m_p \) is a noun phrase (NP) which is contained in another. This information can be provided by a syntactic parser capable of generating phrase structures. The noun phrase *M. jannaschii* in the partial parse tree shown in Figure 3.4, for instance, is nested within the NP *a nucleotide-binding USP from M. jannaschii, MJ0577*.

- Is a subject: Returns true if \( m_p \) acts as a subject in the given sentence. Predicate-argument structures given by a syntactic parser provide this information. Shown in Figure 3.5 is an example of a parse tree. Conventionally, the subject of a
3.3. METHODOLOGY

Figure 3.4: Partial syntactic parse tree showing *M. jannaschii* as a nested noun phrase.

sentence is indicated as the first argument, labelled as arg1, of the sentence’s head node. In the given example, the sentence’s head node is the verb *inhibited*. Labelled as the first argument of *inhibited*, the mention *The lead compound* is the subject of the sentence.

Features describing an active mention $m_a$.

- **Number**: Determines if $m_a$ is in the singular or plural case.
- **Is a pronoun**: Returns *true* if $m_a$ is a pronoun.
- **Is a nested noun phrase**: Checks whether $m_p$ is a noun phrase (NP) which is contained in another, based on phrase structures provided by a syntactic parser.
- **Semantic type**: Determines whether $m_a$ is any of the following types: chemical compound, organism, gene/protein, enzyme, cell line, disease and biological
Figure 3.5: Syntactic parse tree showing The lead compound as the subject of the sentence head inhibited.

Table 3.4: Nominative form lookup table.

<table>
<thead>
<tr>
<th>Nominative form</th>
<th>Pronouns</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I, my, mine, me</td>
</tr>
<tr>
<td>you</td>
<td>you, your, yours</td>
</tr>
<tr>
<td>he</td>
<td>he, his, him</td>
</tr>
<tr>
<td>she</td>
<td>she, hers, her</td>
</tr>
<tr>
<td>it</td>
<td>it, its</td>
</tr>
<tr>
<td>we</td>
<td>we, our, ours, us</td>
</tr>
<tr>
<td>they</td>
<td>they, their, theirs, them</td>
</tr>
</tbody>
</table>

activity, based on the results of a named entity recogniser.

- Nominative case: Returns the nominative case of \( m_a \) if it is a pronoun, N/A otherwise. Provided in Table 3.4 is the lookup table used to determine the value of this feature.

Features describing the relation between \( m_p \) and \( m_a \).

- Have the same head noun: Determines whether the two mentions have the same head noun. Predicate-argument structures generated by a syntactic parser indicate the head noun of each noun phrase. For example, the partial parse tree in
Figure 3.6: Partial syntactic parse tree showing *extract* as the head noun of the NP *cell extract*.

- Are the same string: Returns *true* if both mentions are exactly the same string if case is ignored.

- One is a substring of the other: Returns true if one of the two mentions is a substring of the other if case is ignored.

- Are both pronouns and the same string: Determines if the two mentions are both pronouns and exactly match each other if case is ignored.

- Are both proper names and are the same string: Returns true if the two mentions are both proper names and exactly match each other if case is ignored. Tokens assigned POS tags of *NNP* (or *NNPS* for plural forms) signify proper names.

- Are both pronouns: Returns *true* if both mentions are pronouns.

- Are both proper nouns: Checks if both mentions are proper names according to the POS tags assigned to them.
• Neither is a pronoun and are the same string: Returns true if both mentions are non-pronouns but exactly match each other (ignoring case).

• Have the same modifier: Returns true if both mentions have the same modifiers, and N/A if either does not have any modifier. We extract a mention’s modifier by keeping every token in a mention except for the head noun. In Figure 3.6, for instance, the modifier of the NP cell extract is the token cell.

• Have the same nominative form: Returns true if both mentions are pronouns and have the same nominative form, N/A if either is not a pronoun.

• Have the same number: Determines if the two mentions are both in the singular or plural case.

• Are in an appositional relationship: Checks if the two mentions are in an appositional relationship according to the predicate-argument structures given by a syntactic parser. Shown in Figure 3.7, there are two mentions which act as appositives of each other, namely, Sarcotriol and a semi-synthetic derivative of sarcophine.

• $m_a$ is an indefinite noun phrase and is not in an appositional relationship with $m_p$: Returns true if the active mention $m_a$ is an indefinite noun phrase but is not an appositive of the preceding mention $m_p$. Whilst predicate-argument structures from a syntactic parser can provide information on appositional relationships, detection of indefinite noun phrases is facilitated by a set of rules checking if an NP begins with an indefinite article (a or an), an indefinite pronoun (e.g., some, all, such, many, other), or a number in its word form (e.g., one, three, eighteen). This feature will have a value of false for the example in Figure 3.7 since the indefinite noun phrase a semi-synthetic derivative of sarcophine is an appositive of the preceding mention Sarcotriol. For the sentence in Figure 3.8
Figure 3.7: Partial syntactic parse tree showing the appositional relationship between the mentions *Sarcotriol* and *a semi-synthetic derivative of sarcophine*.

however, this feature will be set to *true* since the indefinite noun phrase is not in an appositional relationship with the preceding mention.

- Are in a copular construction: Returns *true* if the two mentions are connected by a copular verb which links a sentence’s subject to a subject complement, as in the sentence in Figure 3.8. As with appositional relations, predicate-argument structures provide this information by labelling each copular auxiliary verb and annotating the subject and subject complement as *arg1* and *arg2*, respectively. The mentions *A. armata* and *a Lessesonian immigrant* ... are in a copular construction, being the subject and subject complement of the copular verb *is*.

- Have the same semantic type: Determines whether both mentions have the same semantic type (any of chemical compound, organism, gene/protein, enzyme, cell line, disease and biological activity).

- One is an alias of the other: Checks if one of the mentions is an abbreviation
of the other based on a lookup list of named entity-abbreviation pairs, generated on-the-fly for each document $d$ by a named entity recogniser (NER). The named entities in $d$ are first tagged by an NER. With $i$ as the index of the last token in any named entity $e$, the following conditions are checked:

- the token at $i + 1$ is the opening parenthesis, and
- the token at $i + 3$ is the closing parenthesis

If both hold true, a step verifying that the named entity $e$ is a possible expanded form of the token at $i + 2$ is performed. This is done using a simple abbreviation detection algorithm [112] that checks if each character in the short form occurs in the candidate expanded form. If named entity $e$ was confirmed to be an expanded form of token $i + 2$, the named entity-abbreviation pair is added to the lookup list for $d$.

- Sentence distance: Returns the distance between $m_p$ and $m_a$ in terms of number of sentences; returns 0 if both mentions are in the same sentence.
3.3. METHODOLOGY

3.3.2.2 Proposed features

In this section, we describe each of the features that were introduced to tune the feature set for coreference resolution in scientific documents from the chemistry domain.

Features describing a preceding mention $m_p$.

- Pertains to chemical compound: Determines if $m_p$ contains a hyponym of "chemical compound" according to a list compiled from WordNet entries [163], provided in Appendix C.2. This feature was introduced to account for sortal expressions which rename chemical entities using categorisation (e.g., acid, lipid, oxide). The mention the jasplamide derivative, for instance, will be given a value of true for this feature since derivative is a hyponym of chemical compound.

- Consists only of numerals: Returns true if $m_p$ consists only of numerical symbols. We included this feature after having observed the extensive use of what we call numerical coreference, or the assignment of arbitrary numbers to chemical entities by authors of chemistry papers, in order to avoid repetition especially of long systematic names and chemical structures throughout a document (Example 3.6).

Example 3.6. A novel anthracycline, komodoquinone A (1), and its aglycone, komodoquinone B (2) were isolated from the solid-state fermentation of the marine Streptomyces sp. KS3, which was isolated from marine sediment. The absolute stereostructures of 1 and 2, except for the sugar portion, were elucidated on the basis of chemical and physicochemical evidence.

- Contains a numerical range: Determines if $m_p$ has a substring corresponding to a range of numbers. Included in the feature set for similar reasons as the preceding feature, this is assigned a value of true for expressions such as 1-4, compounds 7-10 and amides 5-6.
• Contains a number in word form: Returns true if \( m_p \) has a substring corresponding to a number in word form. This feature acts as an indicator of an indefinite noun phrase renaming another mention. An example is the expression Ten in the sentence in Example 3.7.

**Example 3.7.** Ten brominated alkaloids were isolated from the dichloromethane extract of the North Sea bryozoan Flustra foliacea.

**Features describing an active mention** \( m_a \). Aside from the same four features formulated to describe the preceding mention \( m_p \) discussed above, the following were also used to describe the active mention \( m_a \). All of them were introduced into the feature set to account for the different types of coreference we have identified in the HANAPIN corpus.

• Is a definite noun phrase: Returns true if the active mention \( m_a \) is a definite noun phrase, i.e., starts with the article the.

• Is a relative pronoun: Determines if \( m_a \) is a relative pronoun. This is facilitated by checking the part-of-speech tags assigned by a tagger to a sentence, through which relative pronouns can be identified.

• Is a demonstrative noun phrase: Returns true if \( m_a \) begins with any of the following demonstrative determiners: this, such, these, those and that.

• Begins with a distributive pronoun: Returns true if \( m_a \) starts with any of the following distributive pronouns: both, each, either, neither, any and other.

**Features describing the relation between** \( m_p \) **and** \( m_a \).

• Token distance: Returns the distance between \( m_p \) and \( m_a \) in terms of number of tokens.
• Are consecutive mentions with matching semantic types: Checks if \( m_p \) and \( m_a \) are consecutive mentions and if they have the same semantic types based on the responses of a named entity recogniser. This feature was formulated to capture appositives which appear in the same noun phrase, as in Example 3.8 where the mentions the Formosan soft coral and Clavulariaviridis are in an appositional relationship.

Example 3.8. Three new cytotoxic prostanoids were isolated from the methylene chloride solubles of the Formosan soft coral Clavulariaviridis.

• Levenshtein distance: Also known as the edit distance, this string similarity feature returns the required number of one-character edits (i.e., insertions, deletions and substitutions) to transform one mention into the other. The mentions Sargassum fulvellum and S. fulvellum, for instance, have eight (8) as the edit distance between them. This feature is useful for capturing similarity between morphological and/or spelling variants of the same name.

• Have minimal Levenshtein distance: Determines if the surface forms of the two mentions are similar enough by checking if the edit distance between them is minimal (i.e., below an established threshold of 5).

• Longest common subsequence length: Returns the length of the longest common subsequence (LCS) between the two mentions. This corresponds to the longest possible sequence of tokens, consecutive or otherwise, which is common to both \( m_p \) and \( m_a \). The mentions a temperate distributed species and a tropical to warm temperate species, for example, have a temperate species as their longest common subsequence. Like the Levenshtein distance, this feature can capture similarity between variants of the same expression.

• Have maximal longest common subsequence length: Determines if the overlap
between the two mentions is significant by checking if the length of the LCS is above an established threshold of 7.

**Concatenated features describing the relation between** $m_p$ **and** $m_a$. As with the work of Rahman and Ng\cite{143}, we also employed concatenated features in describing the relationship between two mentions, in order to capture patterns in their different combinations. These are generated by concatenating the values of each of the following features for $m_a$ and $m_p$:

- Number
- Is a pronoun
- Is a nested noun phrase
- Semantic type
- Nominative case
- Pertains to a chemical compound
- Consists only of numerals
- Contains a numerical range
- Contains a number in word form

### 3.3.2.3 Feature encoding

Whilst many of the features we have described in the above sections are binary (i.e., having only *true* or *false* as possible values, as in the case of the ones checking if $m_a$ is a pronoun, a nested noun phrase or consists only of numerals), some of our features are multi-valued (i.e., can take any one value out of multiple categories). Some
examples of these multi-valued features include the ones determining the semantic type, nominative form, Levenshtein distance and the length of the longest common subsequence. A feature conversion step is performed to convert every feature value to its binary equivalent. For example, the semantic type feature, which can have any of (1) chemical compound, (2) organism, (3) gene/protein, (4) enzyme, (5) cell line, (6) disease and (7) biological activity as its value, will be converted to seven binary features, one for each of the possible types. Only the binary feature corresponding to the determined semantic type will be enabled (i.e., given a value of true). We applied the same procedure to Levenshtein distance and LCS length by setting the maximum number of possible values based on certain thresholds.

Whilst two of the four coreference resolution paradigms (discussed in more detail in the succeeding section) employ features of mention pairs, the other two require features describing an instance consisting of an active mention $m_a$ and a chain of preceding mentions $c$. In describing the relationship between $m_a$ and $c$, we adopted Rahman and Ng’s four chain-level predicates [143] which are listed below. For any feature $f$ used to describe the relation between $m_a$ and every preceding mention $m_p$ in the chain, we count the number of times that the value assigned to $f$ is true. Exactly one of the following four predicates will be enabled (i.e., assigned a value true) based on the ratio of the count $n$ to the total number of mentions in the chain $|c|$.

- **All**: enabled if $n = |c|$

- **MostTrue**: enabled if $\frac{|c|}{2} \leq n < |c|$

- **MostFalse**: enabled if $0 < n < \frac{|c|}{2}$

- **None**: enabled if $n = 0$
CHAPTER 3. COREFERENCE RESOLUTION

3.3.3 Implementation of paradigms

Following the ideas proposed by Rahman and Ng [143], we developed our own implementation of each paradigm. To illustrate our points, we will be using as a running example the annotated text extract in Example 3.9 with the mention These as the active mention (i.e., the current expression of interest). We will assume, for the sake of illustration, that the annotations in this example are gold standard.

Example 3.9. Indeed, we have recently identified several cyanobacterial compounds that are potent [protease] inhibitors, such as [grassystatins A-C] that selectively inhibit [cathepsin E] and [lyngbyastatins 4-7] [which] inhibit [porcine pancreatic elastase]. [The latter group] are part of a class prolifically produced by marine and aquatic cyanobacteria, containing [3-amino-6-hydroxy-2-piperidone] ([Ahp]) as part of a six-unit cyclic core with a pendant side chain. [These] generally inhibit certain serine proteases.

Text spans enclosed in square brackets correspond to mentions. Coreferent mentions are underlined using the same style (i.e., those with single underlines refer to one entity and those with double underlines refer to another). Mentions which are not underlined are not coreferent with any other mention, except These which is the active mention $m_a$.

3.3.3.1 Mention pair classification (MPC)

As described in Section 3.1.2.3, this binary classification-based paradigm requires instances in the form of mention pairs. To train the classifier, learning instances were generated based on gold standard annotations, i.e., by using pairs of coreferent mentions for positive instances and pairs of non-coreferent mentions for negative ones. A positive-labelled instance was formed between any active mention $m_a$ and the closest preceding mention $m_c$ that corefers with it. Instead of forming a negative instance
for every non-coreferent pair in the document, only pairs formed out of mentions between \( m_a \) and every mention between \( m_a \) and \( m_c \) were generated, in order to obtain a balanced number of positive and negative training instances. From Example 3.9, a positive training instance will be formed between the active mention *These* and *The latter group*, whilst the following negative samples will be formed: \(<These, Ahp>\) and \(<These, 3-amino-6-hydroxy-2-piperidone >\). With these training instances, we trained a binary SVM model that predicts whether a mention pair contains coreferring expressions or not; this model was then employed by our coreference resolver. For every mention \( m_a \) in an unseen document, our resolver forms a pair between \( m_a \) and each mention preceding it (starting with the closest one), and presents it to the SVM model. If the model classifies the pair as having coreferring mentions, the resolver links \( m_a \) to the other mention in that pair and proceeds to processing the next mention in the document. Otherwise, it forms another pair between \( m_a \) and the next closest preceding mention and repeats the same process until the model gives a positive prediction or the beginning of the document has been reached. If the model does not give a positive prediction for any of the pairs with \( m_a \), the active mention is considered discourse-new and is not resolved to any referent.

3.3.3.2 Mention pair ranking (MCR)

This ranking-based paradigm is similar to the previously discussed mention pair classification in requiring mention pairs as instances. In the same manner that training instances were generated for MPC, example mention pairs were formed between the active mention \( m_a \) and its closest coreferring preceding mention \( m_c \). Non-examples were also formed with \( m_a \) and each of the mentions between \( m_a \) and \( m_c \). However, to allow the ranker to learn the correct ranking of these instances, priorities were assigned to them (instead of positive or negative labels, as in MPC). Whilst non-coreferent pairs were given the priority value 1, a higher priority (e.g., 2 or any value greater than 1)
was assigned to the pair of coreferent mentions.

Unlike MPC, this paradigm calls for an anaphoricity determination step to account for cases where the active mention \( m_a \) is not referring to any mention (i.e., discourse-new). Unless anaphoricity of \( m_a \) is taken into consideration, a ranking-based coreference resolver will always return the top-ranked instance as a referent. To alleviate this issue, we took inspiration from Rahman and Ng’s joint method for anaphoricity determination and coreference resolution, which entailed the addition of an instance for each active mention \( m_a \) in which \( m_a \) is not paired with any mention. If \( m_a \) is discourse-new (i.e., non-anaphoric), this instance is assigned the higher priority value of 2; pairs between \( m_a \) and every preceding mention are then created and given lower priority. From the same text extract in Example 3.9, the pair \(<\text{These, The latter group}>\) will be given priority 2 whilst \(<\text{These, Ahp}>\) and \(<\text{These, 3-amino-6-hydroxy-2-piperidone}>\) will be given priority 1. Using the generated learning instances, we trained an SVM ranking model which we then employed in our coreference resolver. Given a document, our resolver generates test instances by forming a pair between any active mention \( m_a \) and each of the mentions preceding it, as well as an additional pair with only \( m_a \). If the instance containing only \( m_a \) was given the highest ranking by the SVM model, \( m_a \) is considered discourse-new and is not resolved to any referent. Otherwise, the active mention is resolved to the other mention in the top-ranked pair.

### 3.3.3.3 Mention-chain classification (MCC)

As with mention pair classification, this paradigm is based on binary classification, i.e., a prediction of whether an instance is characterised by coreference or not. However, an instance in this paradigm corresponds to a mention-chain pair, rather than a pair of mentions. To exemplify the generation of instances under this paradigm, we take note of the following partial coreference chains in Example 3.9.
3.3. METHODOLOGY

A positive training instance is generated by pairing the active mention $m_a$ with the partial coreference chain containing the closest coreferent mention. Based on our example, the positive training instance that will be formed for $m_a$ is the pair $<\text{These, } c_4>$ since the closest coreferent mention $m_c$ is *The latter group*. Negative instances will then be generated by pairing $m_a$ with each of the partial chains containing the mentions between $m_a$ and $m_c$. Using the same example, the only negative instance will be $<\text{These, } c_6>$ since the two mentions 3-amino-6-hydroxy-2-piperidone and Ahp belong to the same cluster $c_6$. A binary SVM model is then trained using these instances. As an initial step for utilising the model to make predictions, test instances will be generated by pairing the active mention $m_a$ with each of the partial coreference chains formed so far. The model will then classify each of the test instances as positive (coreferent) or negative (non-coreferent). If none of the instances was predicted as positive, $m_a$ is considered discourse-new and initialises a new coreference chain (i.e., a singleton with only $m_a$ as its member at this point). Otherwise, the chain of the positive instance that contains the mention closest to $m_a$ will be considered as coreferent with $m_a$. The active mention is hence added to the said chain.

3.3.3.4 Mention-chain ranking

Like the mention-chain classification paradigm, mention-chain ranking requires instances to be in the form of mention-chain pairs. During the generation of training
instances, any active mention \( m_a \) is paired with every partial coreference chain formed so far for the current document. To account for anaphoricity, an additional instance that contains only \( m_a \) is also created. However, instead of labelling instances as positive or negative (as with MCC), we indicated which instance is most preferable (i.e., coreferent) by assigning it a higher priority (i.e., the value 2) than the rest which are assigned the value 1. In the case of a discourse-new mention, the most preferred instance pertains to the mention-chain pair which contains only \( m_a \). Using the same text extract in Example 3.9, a total of seven training instances will be created for the active mention *These* which, in this case, is anaphoric. \(<\text{These}, c_4>\) will be assigned the label 2 whilst the following instances will be assigned the label 1: \(<\text{These}, c_1>, <\text{These}, c_2>, <\text{These}, c_5>, <\text{These}, c_6>\) and the special instance \(<\text{These}>\). An SVM ranking model was trained using the generated learning instances.

In testing the model, a set of instances is generated by pairing the active mention \( m_a \) with each of the partial coreference chains. To account for the possibility that \( m_a \) is discourse-new, an additional instance that consists only of \( m_a \) is also created. The model then imposes a ranking on these instances. If the model assigns the highest ranking to the instance which does not have a chain, \( m_a \) is considered discourse-new, is not resolved to any of the partially formed chains and, instead, initialises a new one. Otherwise, \( m_a \) is resolved and added to the chain of the highest-ranked instance. If several instances were given the highest ranking, the instance with the chain containing the mention closest to \( m_a \) is chosen.

### 3.4 Evaluation and Discussion

Described in this section is our experimental setup for coreference resolution. As the HANAPIN corpus is comprised of a relatively small number of documents, we decided to carry out the evaluation using ten-fold cross validation, as opposed to dividing the
data set into training and test sets. As an initial step to ten-fold cross validation, the corpus was split into ten equal subsets (i.e., folds), each one containing two documents. For each fold \( f \), system predictions were generated using a model trained on the other nine subsets, and evaluated against the gold standard annotations for the documents in \( f \).

### 3.4.1 Scoring schemes

For evaluating coreference resolution methods, there is no one established standard yet as to how the values of the performance metrics precision (P), recall (R) and F-score (\( F_1 \)) should be computed. A number of scoring schemes have been proposed, including that of the MUC [164], B\(^3\) [165], BLANC [166] and CEAF [167]. Although the MUC scheme has become the most commonly used scoring scheme, it has a significant shortcoming which will be discussed below. For this reason, we also report our results using not only the MUC scheme but also those of B\(^3\) and BLANC. In discussing the details of each of these schemes, we will be using the sample reference (gold standard annotations) and response chains (system-generated annotations) presented in Table 3.5 which we adapted from Recasens and Hovy’s paper on BLANC [166].

#### Table 3.5: Sample reference and response chains.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)(<em>{G1}) (B)(</em>{G2}) (C)(<em>{G3}) (D)(</em>{G4}) (E, L, N)(<em>{G11}) (F)(</em>{G5}) (G, I)(<em>{G10}) (H)(</em>{G6}) (J)(<em>{G7}) (K)(</em>{G8}) (M)(_{G9})</td>
<td>(A)(<em>{S1}) (B)(</em>{S2}) (C)(<em>{S3}) (D, F)(</em>{S8}) (E, L)(<em>{S9}) (F, I, N)(</em>{S10}) (H)(<em>{S4}) (J)(</em>{S5}) (K)(<em>{S6}) (M)(</em>{S7})</td>
</tr>
</tbody>
</table>

#### 3.4.1.1 MUC

Developed for the coreference resolution task of the 6th Message Understanding Conference, this scoring scheme is based on a comparison of the equivalence classes...
formed by the links in the reference (i.e., the gold standard annotations) and the response (system-generated annotations) \[164\]. With \( S \) as an equivalence class defined by the reference and \( R_1 \ldots R_m \) as the equivalence classes in the response, the scheme makes use of the functions listed below.

- A partitioning of \( S \) as specified by the response \((p(S))\). If we take chain \((E, L, N)_{G11}\) in Table 3.5 as \( S \) and the chain \((E, L)_S9\) as the response, \( p(S) \) will have \((E, L)\) and \((N)\).

- The minimum number of correct links \( c(S) \) necessary to form \( S \). This can be trivially computed as \( c(S) = (|S| - 1) \). This will have a value of 2 for chain \( G11 \).

- The number of links \( m(S) \) which are missing in the response but necessary to completely build \( S \) from the members of \( p(S) \). This can be trivially computed as \( m(S) = (|p(S)| - 1) \) which will have the value 1 for our example \( p(S) \).

Recall can then be computed using Equation 3.14:

\[
R = \frac{c(S) - m(S)}{c(S)} = \frac{|S| - 1 - (|p(S)| - 1)}{|S| - 1} = \frac{|S| - |p(S)|}{|S| - 1} \tag{3.14}
\]

Using our running example, recall will have a value of 0.5 from the following:

\[
R = \frac{|S| - |p(S)|}{|S| - 1} = \frac{3 - 2}{3 - 1} = \frac{1}{2}
\]

To compute recall over a set of responses for an entire document \( D \), we simply obtain the summation using each equivalence class \( S_i \) in the reference, as shown in Equation 3.15:

\[
R_D = \frac{\sum(|S_i| - |p(S_i)|)}{\sum(|S_i| - 1)} \tag{3.15}
\]
From the example in Table 3.5, we will obtain the following value for $R_D$:

$$R_D = \frac{0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 2 + 0 + 0 + 0 + 0 + 0 + 0}{0 + 0 + 0 + 0 + 2 + 0 + 1 + 0 + 0 + 0 + 0 + 1} = \frac{2}{3} = 0.667$$

To compute precision, we switch the roles of the reference and response chains, i.e., we now produce a partitioning $p'(S')$ on an equivalence class $S'$ in the response using chains in the reference. Taking the chain $(E, L)_{S9}$ as $S'$ and $(E, L, N)_{G11}$ as the response, $p'(S')$ will have $(E, L)$.

Precision is then computed with Equation 3.16

$$P = \frac{|S'| - |p'(S')|}{|S'| - 1}$$

With our example, precision will have a value of 1.0 based on the following:

$$P = \frac{2 - 1}{2 - 1} = \frac{1}{1}$$

As with the computation of overall recall, precision over a set of responses for a document $D$ requires a simple summation using each equivalence class $S'_i$ in the response, as shown in Equation 3.17

$$P_D = \frac{\sum(|S'_i| - |p'(S'_i)|)}{\sum(|S'_i| - 1)}$$

Again using the example in Table 3.5 we will obtain $P_D$ as follows.

$$P_D = \frac{0 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 0}{0 + 0 + 0 + 1 + 1 + 2 + 0 + 0 + 0 + 0 + 1} = \frac{2}{4} = 0.50$$
CHAPTER 3. COREFERENCE RESOLUTION

The value of F-score on a document can then be computed in the traditional manner, i.e., as the harmonic mean of precision and recall (Equation 3.18).

\[ F_1 = 2 \cdot \frac{P_D \cdot R_D}{P_D + R_D} \]  

(3.18)

Using this equation, we will obtain an F-score of 0.571 for our running example in Table 3.5.

The MUC scoring scheme, however, has one major shortcoming, i.e., it does not account for singletons or coreference chains containing only one mention. It does not give any credit to a coreference resolver which made the correct decision of not merging a singleton with another chain. Since our corpus contains singletons, it is desirable to explore other scoring schemes for evaluating our methods.

3.4.1.2 B³

Instead of counting links, the B³ algorithm computes for the values of precision and recall for each mention \( m_i \) in a document [165]. Representing the reference and response chains containing \( m_i \) as \( G_{m_i} \) and \( S_{m_i} \) respectively, we use Equations 3.19 and 3.20 to determine precision and recall.

\[ P = \frac{\text{number of correct elements in } S_{m_i}}{|S_{m_i}|} \]  

(3.19)

\[ R = \frac{\text{number of correct elements in } S_{m_i}}{|G_{m_i}|} \]  

(3.20)

The final values of precision and recall over an entire document \( D \) having \( N \) mentions are calculated with Equations 3.21 and 3.22.

\[ P_D = \sum_{i=1}^{N} w_i \cdot P_i \]  

(3.21)
3.4. EVALUATION AND DISCUSSION

\[
RD = \sum_{i=1}^{N} w_i \ast R_i
\]  

(3.22)

where \( w_i \) is the weight of mention \( m_i \). The developers of this scoring scheme suggest the use of the same weight for each \( m_i \), specifically, \( 1/N \). Taking the reference and response chains in our example in Table 3.5 we compute for the values of \( PD \) and \( RD \) as follows.

\[
PD = \frac{1}{14} \ast \left[ \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{2}{2} + \frac{2}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} \right] = \frac{70}{84} = 0.833
\]

\[
RD = \frac{1}{14} \ast \left[ \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} \right] = \frac{76}{84} = 0.905
\]

As with the MUC scheme, F-score is computed simply by obtaining the harmonic mean of precision \( PD \) and recall \( RD \). The F-score for our example will be 0.868.

Unlike the MUC scheme, the \( B^3 \) scoring scheme gives credit to correctly identified singletons since it takes into consideration each mention in a document, regardless of the chain it belongs to.

3.4.1.3 BLANC

Given reference and response chains, the BLANC scoring scheme [166], as an initial step, categorises links according to two dimensions:

1. The actualisation of links (according to an automatic coreference resolver)

   - coreference (c): a link between any two coreferent mentions
   - non-coreference (n): a link between any two mentions which are not coreferent
Table 3.6: BLANC Confusion matrix for example in Table 3.5

<table>
<thead>
<tr>
<th></th>
<th>Response</th>
<th>Coreference (c)</th>
<th>Non-coreference (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Coreference (c)</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Non-coreference (n)</td>
<td>3</td>
<td>84</td>
</tr>
</tbody>
</table>

2. The correctness of links (according to an evaluator)

- right (r): a link whose value (c or n) is the same in both the reference and response chains
- wrong (w): a link with inconsistent values in the reference and response chains

Provided in Table 3.6 is the confusion matrix for the running example in Table 3.5. Based on such confusion matrix, BLANC defines four different frequency counts, one for each of right coreference links $rc$, wrong coreference links $wc$, right non-coreference links $rn$ and wrong non-coreference links $wn$.

Adapting the Rand index [168], BLANC gives equal importance to both coreference and non-coreference links. As such, it calculates precision, recall and F-score for each category of links (coreference and non-coreference) before computing for the final scores by averaging. On the one hand, we use Equations 3.23, 3.23 and 3.23 to compute the correctness of the coreference links. To exemplify, we also show the worked out equation for our running example.

\[
P_c = \frac{rc}{rc + wc} = \frac{2}{2 + 3} = 0.4 \quad (3.23)
\]

\[
R_c = \frac{rc}{rc + wn} = \frac{2}{2 + 2} = 0.5 \quad (3.24)
\]
3.4. EVALUATION AND DISCUSSION

\[
    F_c = 2 \cdot \frac{P_cR_c}{P_c + R_c} = 2 \cdot \frac{0.4 \cdot 0.5}{0.4 + 0.5} = 0.444 \tag{3.25}
\]

On the other hand, the following equations (3.26, 3.26 and 3.26) are for computing the correctness of the non-coreference links. Again, we also provide the worked out equation for our running example for illustration purposes.

\[
P_n = \frac{rn}{rn + wn} = \frac{84}{84 + 2} = 0.977 \tag{3.26}
\]

\[
R_n = \frac{rn}{rn + wc} = \frac{84}{84 + 3} = 0.966 \tag{3.27}
\]

\[
F_n = 2 \cdot \frac{P_nR_n}{P_n + R_n} = 2 \cdot \frac{0.977 \cdot 0.966}{0.977 + 0.966} = 0.971 \tag{3.28}
\]

The final BLANC scores are then obtained by simple averaging:

\[
P = \frac{P_c + P_n}{2} = 0.689 \tag{3.29}
\]

\[
R = \frac{R_c + R_n}{2} = 0.733 \tag{3.30}
\]

\[
F = \frac{F_c + F_n}{2} = 0.708 \tag{3.31}
\]

For the rules used in handling boundary cases, we refer the reader to the BLANC paper [166].
3.4.2 Pre-processing pipeline

As with our method for named entity recognition, a series of pre-processing steps were performed on the documents, consisting of sentence splitting, tokenisation and part-of-speech and chunk tagging. We refer the reader to Section 2.3.1.1 for the specific details of each of these steps. In addition to these, the steps described below were also performed to facilitate the generation of many of the features described in Section 3.3.2.

3.4.2.1 Named entity recognition

For each fold, a conditional random fields (CRF) model for recognising semantic types annotated in the HANAPIN corpus (i.e., chemical compound, organism, gene/protein, enzyme, cell line, disease, biological activity) was trained using the CRFsuite package [169]. This software is an implementation of the CRF algorithm [107], discussed in Section 2.2.1. Included in the downloadable package is a feature generation script for biomedical NER which extracts features such as orthography, word shape and affixes, as well as unigrams and bigrams over token surface forms, part-of-speech tags and chunk tags. CRFsuite reads in gold standard annotations and outputs its predictions in the BIO encoding format which we have described in Section 2.2. We have also incorporated a simple abbreviation recognition method [112] to capture abbreviations missed by the CRF models. Evaluated using ten-fold cross-validation, the NER achieved micro-averaged precision of 91.7%, recall of 73.9% and F_1 score of 81.84%.

3.4.2.2 Syntactic parsing

All of the documents in the HANAPIN corpus were processed by the Enju syntactic parser [170]. Based on a head-driven phrase structure grammar (HPSG), Enju provides phrase and predicate-argument structures for each input sentence. Although Enju can take as input raw sentences, we provided it with tokenised and POS-tagged sentences
3.4. EVALUATION AND DISCUSSION

from our previous pre-processing steps, and disabled its own tokeniser and tagger. The parser can process both general- and biomedical-domain documents (with around 90% accuracy), as it can be configured to employ either of two models: a general one trained on the Brown corpus [171] and a biomedical one trained on GENIA [22]. Parse results are provided in one of three formats, namely, as predicate-argument relations, as XML elements and as stand-off annotations. To facilitate the alignment of the parse results with responses from named entity recognition, we configured Enju to produce outputs in stand-off format.

3.4.3 Training and development

Since we preferred to focus our investigation on the comparison of the four paradigms, it was desirable to eliminate any influence that might be propagated by errors in automatically extracted mentions. For this reason, we decided not to perform any automatic mention extraction step; instead, we assume that the input documents to our coreference resolvers include pre-annotated mentions.

Each of the four coreference resolution paradigms was implemented in the Java programming language and built on top of our SVM implementation of choice, the SVMlight package. We employed the learning and classification modules of SVMlight using the default parameters.

For each of the four paradigms, the following procedure was performed for each of the 10 data folds.

1. Training instance generation with the HANAPIN corpus as our gold standard data

2. Model training (facilitated by the learning module of SVMlight)

3. Test instance generation using gold standard mentions

---

http://svmlight.joachims.org/
4. Model prediction (facilitated by the classification module of SVMlight)

Response coreference chains were generated based on the predictions of the SVM models. On the one hand, the mention-chain classification and ranking paradigms do not require any post-processing step since their outputs are already in the form of coreference chains. For the mention pair classification and ranking paradigms, on the other hand, coreference chains were formed by linking together resulting mention pairs having common members. The responses were then evaluated against the gold standard annotations in the HANAPIN corpus.

### 3.4.4 Comparison

Two rounds of experiments were carried out: the first one employing only the default features and the second using the feature set enriched by our own proposed features. Each test document was scored based on the MUC, $B^3$ and BLANC scoring schemes, described in Section 3.4.1. The results of each of the two rounds of experiments, macro-averaged over all the documents in the corpus, are summarised in Table 3.7. The same results can be visualised in Figures 3.9, 3.10 and 3.11 which show the performance of each paradigm in each experiment round according to the MUC, $B^3$ and BLANC scoring schemes, respectively.

We observe that regardless of the feature set used, mention pair ranking achieves the best precision scores amongst the four paradigms. The highest recall scores, on the contrary, were obtained by the mention-chain classification paradigm. The best balanced F-scores were obtained by mention pair classification (according to the MUC and BLANC scoring schemes) and mention pair ranking (according to $B^3$ scoring).

Whilst the use of default features results in optimal precision for mention pair ranking and mention-chain classification, the employment of our enriched feature set brings about better precision for mention pair classification. Our enriched feature set resulted
3.4. EVALUATION AND DISCUSSION

Table 3.7: Summary of macro-averaged results for each paradigm using default (d) and enriched (e) feature sets.

<table>
<thead>
<tr>
<th></th>
<th>MUC</th>
<th></th>
<th></th>
<th>B3</th>
<th></th>
<th></th>
<th>BLANC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>MPC₁</td>
<td>62.23</td>
<td>68.11</td>
<td>65.04</td>
<td>63.33</td>
<td>71.21</td>
<td>67.04</td>
<td>76.16</td>
<td>70.84</td>
</tr>
<tr>
<td>MPC₂</td>
<td>70.37</td>
<td>71.58</td>
<td>70.97</td>
<td>79.65</td>
<td>74.33</td>
<td>76.90</td>
<td>85.25</td>
<td>76.23</td>
</tr>
<tr>
<td>MPR₁</td>
<td>77.80</td>
<td>62.64</td>
<td>69.40</td>
<td>85.87</td>
<td>69.72</td>
<td>76.96</td>
<td>86.10</td>
<td>72.55</td>
</tr>
<tr>
<td>MPR₂</td>
<td>77.78</td>
<td>63.59</td>
<td>69.97</td>
<td>85.69</td>
<td>69.33</td>
<td>76.64</td>
<td>85.88</td>
<td>72.46</td>
</tr>
<tr>
<td>MCC₁</td>
<td>66.48</td>
<td>73.02</td>
<td>69.60</td>
<td>74.30</td>
<td>74.08</td>
<td>74.19</td>
<td>80.82</td>
<td>75.72</td>
</tr>
<tr>
<td>MCC₂</td>
<td>65.33</td>
<td>75.84</td>
<td>70.20</td>
<td>68.25</td>
<td>77.12</td>
<td>72.42</td>
<td>75.86</td>
<td>77.69</td>
</tr>
<tr>
<td>MCR₁</td>
<td>70.50</td>
<td>49.52</td>
<td>58.18</td>
<td>78.44</td>
<td>65.86</td>
<td>71.60</td>
<td>72.38</td>
<td>65.17</td>
</tr>
<tr>
<td>MCR₂</td>
<td>73.35</td>
<td>62.49</td>
<td>67.48</td>
<td>75.41</td>
<td>72.14</td>
<td>73.74</td>
<td>70.63</td>
<td>73.61</td>
</tr>
</tbody>
</table>

in consistently better recall for all paradigms except mention pair ranking. It also consistently improved the F-scores for mention pair classification and mention-chain ranking.

Quite unexpectedly, the best F-scores were obtained with the use of the paradigms based on mention pair instances (i.e., mention pair classification and ranking). We attribute this to the errors brought about during feature extraction, specifically in the values of the semantic type features supplied by our named entity recogniser. Our NER performs quite poorly on types such as gene/protein, enzyme and disease, for which the corpus contains sparse training samples. As a result, the values of our semantic type feature were not very reliable. The said features had a detrimental impact especially on the paradigms based on mention-chain instances, since their contribution was effectively more significant at the chain level than at the individual mention level.

Since our work is, to the best of our knowledge, the first investigation into the full-text coreference resolution problem in a chemical domain, it is not directly comparable with the biomedical coreference resolution methods we have reviewed in Section 3.1.2.4. Considering, however, that Gasperin and Briscoe obtained an F-score of 68.3% for sortal coreference resolution in full-text biomedical papers, we can say that
the best F-scores we obtained, ranging from 70 to 80.5%, were quite satisfactory.

3.5 Summary

In this chapter, we presented our study on coreference resolution for scientific documents from a chemical domain. A thorough review of the state-of-the-art in coreference-annotated corpora and proposed approaches was provided, followed by a detailed comparison. We then described our procedure for the development of the HANAPIN corpus, our own manually annotated document collection consisting of full-text scientific articles from a pharmaceutical subdomain. In presenting our proposed method, we provided details of the support vector machines-based paradigms we have adopted, as well as of the enriched feature set that we designed to capture chemical coreference. Lastly, by means of ten-fold cross validation on the HANAPIN corpus, we evaluated
3.5. SUMMARY

Figure 3.10: Comparative evaluation results using the B3 scoring scheme.

Figure 3.11: Comparative evaluation results using the BLANC scoring scheme.
our proposed method according to the MUC, B³ and BLANC scoring schemes and obtained satisfactory F-scores ranging from 70 to 80.5%.
Chapter 4

Interaction extraction

In the competitive fields of chemoinformatics and drug discovery, literature awareness is crucial as it translates to a more focused effort for lead compound discovery [172] and more opportunities for developing new drug targets [173]. Given the vast amount of available scientific literature, researchers in the field have recently turned to information extraction to reduce the amount of time spent on manually distilling information from unstructured data [173, 174]. Its application has been explored for tasks such as extraction of genotype-phenotype-drug relationships [175, 176], compound profiling [34], automatic curation of chemoinformatic databases [33, 30-32], extraction of pathways [177, 43], identification of adverse drug reactions [38, 178] and drug repurposing [38-42].

The backbone of most of these text mining applications is the extraction of interactions between chemical (i.e., drugs) and biological entities (e.g., genes or proteins). In drug repurposing, for instance, the extraction of a drug’s associated target entities and biological processes from the literature enables the building of drug and target profiles which facilitate the prediction of new drug uses [42]. In this chapter, we describe our work on the development of techniques for extracting drug-target interactions from the literature.
Poly-APS can act as AChE inhibitors.

Figure 4.1: An interaction where the patient is an entity: agent = Poly-APS, interaction clue = inhibitors and patient = AChE

Bisebromoamide inhibits ERK phosphorylation.

Figure 4.2: An interaction where the patient is a biological process: agent = Bisebromoamide, interaction clue = inhibits and patient = ERK phosphorylation

A drug-target interaction (DTI) is any association between a drug and a gene or gene product (GGP) implying a biological activity or effect. As the name implies, it involves two types of entities, namely, drugs and drug targets. We define a drug as any chemical substance that produces a pharmacological effect; it can pertain to a compound which has already been clinically approved or otherwise. A target, in contrast, is a GGP or GGP-centric biological process that drugs are usually observed in relation to or tested against. Each of the drug and GGP participating in an interaction may take a role of an agent or patient. Whilst an entity acting as an agent serves as the initiator of the biological effect, one which has a patient role is the recipient of the said effect elicited by the agent. The association between a drug and its target is centred on an interaction clue, i.e., a word or sequence of words that serves as evidence of the interaction. Examples of expressions pertaining to agents, patients and interaction clues are shown in Figures 4.1 and 4.2.

From a linguistic point of view, such interactions can be cast either as relations or as events. A relation is a binary association represented as a link between its two participants. The relation may be typed, in which case the link is assigned a label. If participants in a relation assume the same roles, we say that the relation is symmetric, in which case the link is undirected. Otherwise, the relation is asymmetric and is represented using a directed link. The roles in relations are implicitly specified by the direction of the link. A participant shown as the origin, for example, of a link is
Poly-APS can act as AChE inhibitors.

Bisebromoamide inhibits ERK phosphorylation.

Figure 4.3: Drug-target interactions represented as events (left-hand side) and as relations (right-hand side).

A more expressive representation is the event which allows for the inclusion of not only two but any number of participants (event arguments), each of which can take any one of various roles. Unlike relations, events support the specification of the interaction’s textual evidence (event trigger). If an event is typed, the trigger is assigned the appropriate semantic label. For each event argument \( a \), a link originates from the trigger and terminates at \( a \). Roles of arguments are explicitly indicated as labels of links, unlike in the case of relations. Another significant difference between relations and events is the support of the latter for the representation of more sophisticated, nested associations by allowing any number of arguments to take values which are also events. Furthermore, the event representation can capture attributes or modifications such as negation and speculation, also known as meta-knowledge [179].

Shown in Figure 4.3 are two drug-target interactions represented as events on the left-hand side of the figure and as relations on the right. Noticably, some loss of information is brought about by the relation representation. A DTI element that is clearly missing in the right-hand side examples is the interaction clue. Furthermore, the relation representation fails to accurately capture, for instance, details of the association between Bisebromoamide and ERK. By looking at sentence (2b) in the figure, one might acquire the understanding that the first entity directly interacts with the second, when in fact the interaction is with the biological process involving the second entity.
Considering these issues, we chose to cast drug-target interactions as events, and correspondingly, their automatic extraction as an event extraction task. As an event consists of various elements, its extraction can be decomposed into several subtasks. These include the core subtasks of event trigger recognition and event argument recognition, as well as the optional identification of event meta-knowledge. In this study, we focus on the core event extraction task and exclude from our scope meta-knowledge identification. It is assumed that any document text provided as input will have already been pre-annotated with named entities.

Previously known as the scenario template extraction task in MUC-6 [52] and 7 [53], event extraction originally pertained to the recognition of news events such as labour negotiations and rocket or missile launches. In the Automatic Content Extraction (ACE) program in 2004 [54], event types such as destruction/damage, creation/improvement, transfer of control, movement and interaction of agents were defined. Complementing those types are event roles such as agent, object, source, target, time and location. In the 2005 edition of ACE [121], the event categorisation was redefined to include eight types, each of which is further subdivided into several subtypes.

The biomedical text mining community then adapted the notion of event to represent associations at the molecular level, e.g., protein-protein interactions and gene regulation. For a number of years, most of the biomedical event extraction tasks were designed to capture such types. The first shared task of this kind, BioNLP 2009, was organised to encourage the biomedical NLP community to develop event extraction techniques for event types such as gene expression, transcription, phosphorylation, regulation, binding, catabolism and localisation, all of which have genes or proteins as participants [17]. These event types are collectively known as GENIA events owing to the use of the GENIA corpus [22] as the primary source of annotations for these event types. Although the succeeding edition, BioNLP 2011, touched on more
4.1 LITERATURE REVIEW

subject domains (e.g., infectious diseases, bacterial interactions, epigenetics and post-translational modifications), the event types were still mostly focussed on molecular events involving GGPs as participants [18]. This was changed, however, in the most recent BioNLP 2013 where tasks relevant to pathway curation and cancer genetics were also defined, leading to the inclusion of more diverse event types such as metabolic processes and even anatomical-level (e.g., blood vessel development) and pathological-level (e.g., metastasis) events [180].

The extraction of drug-target interactions as events is an interesting problem, considering that the extraction of events involving drugs has not been explored previously. Perhaps the most similar task to ours, in terms of domain, is the extraction of drug-drug interactions [114], although still different as this has been cast as a relation extraction (rather than as an event extraction) problem. Biomedical event extraction is still considered an unsolved problem, with the best performing systems achieving F-scores ranging from 55% to 58%. Casting the extraction of drug-target interactions as an event extraction task hence poses a challenging, non-trivial problem.

In the next section, we provide an overview of existing supporting resources, followed by a review of various proposed methods for biomedical event extraction.

4.1 Literature Review

4.1.1 Annotated Corpora

4.1.1.1 Biomedical event-annotated corpora

GENIA Corpus. The publicly available GENIA corpus[1] is a collection of 1,999 MEDLINE abstracts on the topic of human blood transcription factors, annotated at various linguistic and semantic levels [22]. On top of the part-of-speech and term

annotations, the developers of the corpus added event annotations to 1,000 of the abstracts, resulting in more than 36,000 annotated biomolecular events. By their definition, an association is annotated only when at least one of the participating GGPs is affected. It follows that similarity as well as static relations such as part_of and is_a were not included in the scope of the annotation task. The annotation scheme required the annotators to annotate an event only if expressions in text signifying the event exist. For each event, a stand-off event element was created, consisting of event attributes and elements. Attributes include assertion (whether the event is positive or negated) and uncertainty (whether the statement of the event is certain, probable or doubtful). Elements were used to specify the event trigger, event type and arguments (e.g., theme and cause). The value assigned to event type can be any of the ones defined in the GENIA ontology \[181\] which contains entries corresponding to the Gene Ontology’s biological processes (e.g., cellular and physiological) and molecular functions \[182\].

The scheme, implemented as an XML document type definition (DTD) file, provided support for the annotation of nested events by allowing any of the arguments to take as its value another event element. The annotation task was carried out by three molecular biology graduate students with the use of the XML Concordancer (XConc) Suite, which runs as an Eclipse plug-in and saves annotated documents in XML format.

**Gene Regulation Event Corpus (GREC).** The Gene Regulation Event Corpus \[183\] consists of 240 abstracts from MEDLINE, in which a total of 3,067 events were annotated. The annotation task covered gene regulation and expression events, including related information such as location, time and manner. As with the annotation of the GENIA corpus, annotators were asked to annotate events only if they are expressed in text by triggers. To facilitate the faster identification of event triggers (and thus, of events), a curated list of 353 gene regulation verbs was given to the two annotators.

\[2\]http://www.eclipse.org
The annotation scheme also required that each event is sentence-bound, i.e., each of its arguments is found within the same sentence. An event argument is assigned any one of the following 13 semantic roles: agent, theme, manner, instrument, location, source, destination, temporal, condition, rate, descriptive agent, descriptive theme and purpose. Each argument was also linked to the best matching concept in the Gene Regulation Ontology\(^3\). A Java-based annotation tool called WordFreak was employed in the annotation task after some customisations (e.g., dictionary-based pre-annotation of gene regulation verbs). GREC has been made available for public download\(^4\) in both XML and delimiter-separated values format.

**Bio Information Extraction Resource (BioInfer) Corpus.** The publicly available BioInfer Corpus\(^5\) contains 1,100 sentences from biomedical abstracts, annotated at three different levels: named entities, relationships and syntactic dependencies \([21]\). Many of the relationship types of interest were formally defined as events similar to those in the GENIA corpus. Whilst similar to that of GENIA’s in supporting the annotation of nested associations, the BioInfer annotation scheme differs from it by including static relations in the annotation scope. It is also unique in representing relationship annotations by means of logic formulae. In marking up a relationship, an annotator uses a predicate from the BioInfer relationship ontology and fills in its arguments with entities or other predicates. A total of 2,662 relationships were annotated in the corpus.

\(^{3}\)http://www.ebi.ac.uk/Rebholz-srv/GRO/GRO.html
\(^{4}\)http://www.nactem.ac.uk/GREC
\(^{5}\)http://mars.cs.utu.fi/BioInfer
BioNLP Shared Task Corpora. Whilst the first edition of the BioNLP shared task in 2009 provided only a slightly modified version of the GENIA corpus as its benchmark data, the succeeding editions, having more tracks, produced a number of event-annotated corpora. In BioNLP 2011, tracks for the extraction of events pertaining to infectious diseases [184], epigenetics and post-translational modifications [185] and bacterial interactions [186] were introduced, for each of which event-annotated corpora was developed as a resource. Similarly, annotated corpora were provided by the organisers of the BioNLP 2013 event extraction tracks on newly explored domains (e.g., cancer genetics [187], pathway curation [188], gene regulation [189, 190]). All of these newer corpora follow the general structure of GENIA events, consisting of semantically labelled triggers and arguments and allowing nested associations. Whilst we provide in Table 4.1 some of the most pertinent information about these corpora, we refer the reader to the overview paper of each BioNLP track for more details.

4.1.1.2 Summary

Provided in Table 4.1 is a summary of details of the biomedical event-annotated corpora we have reviewed. Almost all of them are similar in terms of annotation scheme or representation, i.e., they all use event frames having slots for event type, trigger and arguments, except for BioInfer. The document composition of all of the corpora is the same (i.e., MEDLINE abstracts or sentences), except for the BioNLP 2011 infectious diseases corpus which is comprised of open-access full-text articles. As one might expect, all of the BioNLP shared task corpora are available in the same stand-off delimiter-separated values format which is the native format of the tool used in the annotation of most of them, i.e., the brat rapid annotation tool (brat) [191]. All of these biomedical event-annotated corpora have been made publicly available.
Table 4.1: Comparison of Corpora with Biomedical Event Annotations

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Domain</th>
<th>Document Type/Size</th>
<th>Scheme</th>
<th>Encoding/IAA</th>
<th>IAA</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENIA</td>
<td>human blood transcription factors</td>
<td>1,999 MEDLINE abstracts</td>
<td>event frames</td>
<td>stand-off XML</td>
<td>56% (F_1) (strict) 77% (F_1) (relaxed)</td>
<td>public</td>
</tr>
<tr>
<td>GREC</td>
<td>gene regulation</td>
<td>240 MEDLINE abstracts</td>
<td>event frames</td>
<td>stand-off XML &amp; stand-off DSV</td>
<td>66-90% (F_1)</td>
<td>public</td>
</tr>
<tr>
<td>BioInfer</td>
<td>general molecular biology</td>
<td>1,100 sentences from MEDLINE abstracts</td>
<td>predicates</td>
<td>stand-off XML</td>
<td>undisclosed</td>
<td>public</td>
</tr>
<tr>
<td>BioNLP 2011</td>
<td>infectious diseases</td>
<td>30 PubMed Central open-access articles</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>75% (F_1)</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td>epigenetics &amp; post-translational modifications</td>
<td>1,200 MEDLINE abstracts</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>82% (F_1)</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td>bacterial interactions</td>
<td>142 MEDLINE abstracts</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>undisclosed</td>
<td>public</td>
</tr>
<tr>
<td>BioNLP 2013</td>
<td>cancer genetics</td>
<td>600 MEDLINE from abstracts</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>N/A</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td>pathway curation</td>
<td>525 MEDLINE from abstracts</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>61% (F_1)</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td>gene regulation</td>
<td>300 MEDLINE abstracts</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>43-56% (\kappa)</td>
<td>public</td>
</tr>
<tr>
<td></td>
<td>bacterial gene regulation</td>
<td>201 sentences from MEDLINE abstracts</td>
<td>event frames</td>
<td>stand-off DSV</td>
<td>undisclosed</td>
<td>public</td>
</tr>
</tbody>
</table>
4.1.2 Approaches to event extraction

As noted above, our study is focused on the core event extraction subtasks which include: (1) the recognition of event triggers and their corresponding event types, and (2) the recognition of event arguments and their roles. In this section, we review some of the most notable reported approaches to the core event extraction subtasks.

4.1.2.1 Event trigger recognition

Dictionary-based approaches. Early work on biomedical event extraction (i.e., participating systems in the BioNLP 2009 shared task) saw the application of dictionary-based approaches to event trigger recognition. A dictionary of event triggers occurring in the GENIA corpus, as well as their lemmata, was compiled by the JULIELab team [192]. Biology students were then asked to categorize each trigger as being one of important and discriminative, important but not fully discriminative, non-discriminative, and absolutely non-discriminative. A final dictionary was formed by the combination of entries from the first two groups, against which input text was matched using the dictionary-based chunker tool of LingPipe [111]. Similarly, the submission of the team from the University of Concordia was based on a dictionary of triggers from the GENIA corpus, albeit one in which only single-word verbs, nouns and adjectives were retained [193].

After processing text against their dictionaries, both teams perform a trigger disambiguation step in which each matching trigger was linked to exactly one event type. This was done by the computation of a frequency-based score for each trigger-type pair. A candidate does not qualify as a trigger for a certain event type if the value of this score falls below an empirically determined threshold [192]. The trigger ambiguity problem was addressed differently by the Ghent University team, i.e., by forming a number of disjoint dictionaries, each one corresponding to triggers specific to one of
Machine learning-based approaches. Other approaches to event trigger recognition cast the problem as a multi-way classification task which categorises tokens in text according to the event types they signify (if they are triggers). The Turku Event Extraction System (TEES) employed a multi-class support vector machines (SVM)-based classifier in assigning an event class to each token if it is detected as a trigger or a negative class otherwise [11]. Features used in training and classification include local and contextual token features, frequency features and dependency chains. Token features include orthography, word stem and character $n$-grams, whilst frequency features were generated from bag-of-word counts as well as the number of named entities within a window. Provided by the results of a syntactic parser, dependency information (e.g., tokens and dependency types) reaching up to three levels from a token of interest were also generated as features.

Similarly, the EventMine system treated the trigger recognition problem as a multi-way classification task and also took an approach based on SVM classifiers [195]. It was, however, implemented in two phases, i.e., one for simple event triggers and another for nested events. Features employed in the first phase were similar to those of TEES, with the addition of shortest paths between the token of interest and the GGP$s closest to it based on the output of a syntactic parser. For the detection of triggers for nested events during the second phase, confidence scores given by the SVM model from the previous phase as well as shortest paths between the token of interest and the closest simple event triggers (as detected in the previous phase) were incorporated into the feature set. In a later version of EventMine [10], the feature set was enriched with dictionary features, e.g., synonyms, hypernyms and other related forms of matching tokens provided by WordNet [163] and the UMLS SPECIALIST Lexicon [24].
Whilst different in using L2 regularisation as its classification algorithm, the Stanford Biomedical Event Parser (SBEP) employed features similar to those of TEES [196].

4.1.2.2 Event argument recognition

Rule-based approaches. The team from the University of Concordia which participated in the first BioNLP shared task in 2009 implemented event argument recognition using a rule-based approach. After extracting dependency parses of sentences with the Stanford Parser [197], they formed a grammar based on their observations on dependency relation paths between event triggers and their arguments in gold standard annotations. The most frequently occurring dependency paths were then implemented as rules, applied on sentences in order of simplicity [193]. Similar approaches have been taken by research groups from the Arizona State University [198], University of Cambridge [199] and University of Zurich [200].

Machine learning-based approaches. The Turku Event Extraction System (TEES) represents each sentence as a graph with nodes standing for named entities and triggers, and edges as event arguments. Recognising event arguments is hence equivalent to finding the edges of the graph, cast as a classification problem on which a multi-class SVM model was applied. Each potential edge was classified by the model as any of the allowed event roles (e.g., theme, cause or none). Features used in training the SVM classifier included attributes of tokens such as character \( n \)-grams, named entity type labels and frequencies. Additionally, shortest undirected paths of syntactic dependencies according to results of the Stanford Parser [197] were incorporated into the feature set.

JULIELab’s method extracted sentence-bound pairs of event triggers and arguments [192]. They built two classifiers, namely, a feature-based classifier and a kernel-based
one. For the former, lexical, chunk and dependency parse features were extracted and used to train a maximum entropy classifier. The latter one, in contrast, was implemented as an SVM model trained on graph kernels (i.e., dependency graphs with dependency nodes replaced with labels). Both classifiers independently determine whether a link between the trigger and the argument in each pair should be created or not.

AkaneRE, originally developed as a system for extracting protein-protein interactions, was extended to support biomedical event extraction [201]. As an initial step, event frames (or templates where slots correspond to argument roles) were automatically extracted from the GENIA corpus, resulting in a total of 148 frames. These were grouped into nine general templates, each of which was employed to gather positive and negative instances out of the possible combinations of a sentence’s triggers and named entities. Maximum entropy models, one for each general template, were then trained on these instances. Given input text, AkaneRE generates all possible combinations of triggers and entities in each sentence. Each combination is then presented to each trained model, which predicts the probability of the candidate combination matching the template being represented by the model. Candidates whose scores fall below a certain threshold are then discarded.

Like TEES, EventMine treats the event argument recognition problem as a graph edge detection task [195]. Adopting features employed by TEES, EventMine utilises SVM models to determine the semantic label appropriate for each candidate edge. As with its trigger recogniser, EventMine accomplishes this event extraction subtask in two phases, the first one handling simple events and the second one for nested events. For the SVM model in the second phase, predictions of the SVM model from the previous phase were also used as features.

The Stanford Biomedical Event Parser (SBEP) is unique in casting the edge recognition problem as a dependency parsing task [196]. A dependency parsing model for
the developers’ parser of choice, the MSTParser \[202\], was trained on dependency
trees with incorporated event information. Several features were introduced into the
feature set for training (and applying) the parsing model, including syntactic paths,
original surface forms, parent, child and sibling nodes and ontology match features.
The trained model is then applied by the MSTParser in parsing unseen text. The re-
sulting dependency parse trees with incorporated event information are then converted
to an event frame format.

4.1.2.3 Joint trigger and argument recognition

We can think of the event extraction methods that we have reviewed so far as pipeline-
based approaches in which the two core subtasks of trigger and argument recognition
were addressed consecutively and independently of the other. Other notable solutions
to the event extraction problem were founded on joint approaches to the two subtasks.
Riedel and McCallum of the University of Massachusetts (UMass) were the first to ex-
plain joint inferencing of triggers, incoming/outgoing arguments and protein-protein
bindings with the application of Markov logic-based dual decomposition \[203\]. Sim-
ilarly founded on Markov logic, Poon and Vanderwende’s joint inferencing approach
simultaneously and mutually disambiguates event triggers and arguments \[204\].

Built upon the dual decomposition model from UMass is the FAUST event extrac-
tor \[205\]. Employing a stacking framework, FAUST leverages the UMass system as a
stacking model which enriches the feature set with outputs of a stacked model, i.e., the
Stanford Biomedical Event Parser.

4.1.2.4 Summary

We provide in Table 4.2 a summary of the different approaches to event extraction, sub-
divided into two groups. The first one corresponds to pipeline-based methods which
decompose the task into two consecutive subtasks, namely, trigger recognition and argument recognition. The second group is comprised of approaches which are based on joint models which recognise both triggers and arguments simultaneously. Based on the F-scores obtained by the methods, ranging from 50-58%, it can be clearly observed that event extraction is a very challenging problem which has yet to be solved.
<table>
<thead>
<tr>
<th>Approach</th>
<th>Proponents or System</th>
<th>Key Ideas</th>
<th>Eval. Corpus</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipeline</td>
<td>JULIElab</td>
<td>dictionary-based trigger recognition, trimmed dependency graphs</td>
<td>BioNLP '09 test</td>
<td>47.52%</td>
<td>45.82%</td>
<td>46.66%</td>
</tr>
<tr>
<td></td>
<td>UConcordia</td>
<td>dictionary-based trigger recognition, rules from dependency relations</td>
<td>BioNLP '09 test</td>
<td>61.59%</td>
<td>34.98%</td>
<td>44.62%</td>
</tr>
<tr>
<td></td>
<td>UGhent</td>
<td>trigger disambiguation with disjoint dictionaries</td>
<td>BioNLP '09 test</td>
<td>51.55%</td>
<td>33.41%</td>
<td>40.54%</td>
</tr>
<tr>
<td></td>
<td>TEES</td>
<td>multi-class SVM trigger &amp; argument classifier</td>
<td>BioNLP '09 test</td>
<td>58.48%</td>
<td>46.73%</td>
<td>51.95%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '11 ID test</td>
<td>48.62%</td>
<td>37.85%</td>
<td>42.57%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '11 EPI test</td>
<td>53.98%</td>
<td>52.69%</td>
<td>53.33%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '13 CG test</td>
<td>64.17%</td>
<td>48.76%</td>
<td>55.41%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '13 PC test</td>
<td>55.78%</td>
<td>47.15%</td>
<td>51.10%</td>
</tr>
<tr>
<td></td>
<td>EventMine</td>
<td>multi-class SVM trigger &amp; argument classifier</td>
<td>BioNLP '09 test</td>
<td>58.96%</td>
<td>48.62%</td>
<td>53.29%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '11 ID test</td>
<td>54.97%</td>
<td>60.55%</td>
<td>57.63%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '11 EPI test</td>
<td>55.39%</td>
<td>49.06%</td>
<td>52.03%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '13 CG test</td>
<td>55.82%</td>
<td>48.83%</td>
<td>52.09%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BioNLP '13 PC test</td>
<td>53.48%</td>
<td>52.23%</td>
<td>52.84%</td>
</tr>
<tr>
<td></td>
<td>SBEP</td>
<td>multi-class L2 regularisation trigger &amp; argument classifier</td>
<td>BioNLP '11 ID test</td>
<td>55.86%</td>
<td>46.30%</td>
<td>50.63%</td>
</tr>
<tr>
<td></td>
<td>AkaneRE</td>
<td>matching with automatically extracted templates</td>
<td>BioNLP '09 test</td>
<td>53.56%</td>
<td>28.13%</td>
<td>36.88%</td>
</tr>
<tr>
<td>Joint</td>
<td>UMass</td>
<td>Markov logic, dual decomposition</td>
<td>BioNLP '11 ID test</td>
<td>62.02%</td>
<td>46.92%</td>
<td>53.42%</td>
</tr>
<tr>
<td></td>
<td>Poon &amp; Vanderwende</td>
<td>joint inferencing</td>
<td>BioNLP '09 test</td>
<td>41.55%</td>
<td>28.08%</td>
<td>33.52%</td>
</tr>
<tr>
<td></td>
<td>FAUST</td>
<td>stacking and stacked models</td>
<td>BioNLP '11 ID test</td>
<td>65.97%</td>
<td>48.03%</td>
<td>55.59%</td>
</tr>
</tbody>
</table>
4.2 Corpus Development

As there are no available corpora with annotations for drug-target interactions (DTI), we developed our own corpus which we will henceforth refer to as the HANAPIN DTI corpus. In this section, we describe the procedure by which this corpus was developed and annotated by two domain experts.

4.2.1 Composition of Corpus Documents

Considering that previous work has demonstrated the benefits gained from utilising full-text articles in information extraction tasks [131-134], we developed our corpus from full journal articles. As with our HANAPIN coreference-annotated corpus (Section 3.2), documents were collected from the open-access Marine Drugs journal. We retrieved all available documents except for those which are included in the HANAPIN coreference corpus, resulting in a total of 231 full articles. Each document was submitted to a pre-processing pipeline consisting of the following: (1) metadata removal, (2) MEDLINE XML to plain text conversion, (3) sentence splitting using the LingPipe MEDLINE sentence model [111], (4) tokenisation with the chemical tokeniser of OSCAR [8] and (5) part-of-speech and chunk tagging using the GENIA Tagger [56]. The final corpus contains a total of 49,419 sentences, with 1,108,447 tokens.

4.2.2 Annotation Tasks

In capturing drug-target interactions, we focussed our annotation efforts on interactions involving only certain types of named entities, namely, drugs and enzymes. The

---

6http://www.mdpi.com/journal/marinedrugs
7as of April 2011
8http://sourceforge.net/projects/oscar3-chem/files/chemtok
CHAPTER 4. INTERACTION EXTRACTION

annotation task was divided into two subtasks, i.e., named entity annotation followed by interaction annotation.

4.2.2.1 Named entity annotation

For this task, we define a drug as any chemical compound that produces a pharmacological effect; it may already have been clinically approved or otherwise. Therefore, lead compounds such as natural products, secondary metabolites produced by living organisms, are also considered as drugs by our definition. In order to facilitate a more efficient means of annotating named entities, the articles were automatically pre-annotated with drug and enzyme names. This was accomplished in a series of steps. First, one of the annotators was asked to annotate the names of drugs and enzymes in 15,000 of the sentences in the document collection, using the Accelerated Annotation (ACELA) tool [206].

ACELA is similar to an active learning framework in that it automatically generates candidate annotations based on the manual annotations which are available so far. An annotation project in ACELA is initialised with the provision of just a few annotations (i.e., seed). Using the seed annotations, a conditional random fields (CRF) model is trained and employed to tag entities in 100 randomly selected sentences which will be presented to the curator. The curator manually corrects the automatically generated annotations; this feedback is then used in the training of the CRF model for the next iteration. With the feedback from each succeeding iteration, the CRF model achieves better coverage; consequently, the effort required from the annotator becomes less as the number of iterations increases.

Utilising the named entity annotations in those 15,000 sentences, we developed a named entity recogniser for drugs and enzymes by training a joint CRF model using the CRFsuite package [169]. Although the final model was trained on all of the annotated sentences, the performance of this NER method was initially evaluated using
ten-fold cross validation, by which we obtained 92.96% precision, 91.26% recall and 92.10% F-score for drugs and 93.31% precision, 86.79% recall and 89.92% F-score for enzymes, all computed by micro-averaging. With the final CRF model, the drug and enzyme names in sentences in all 231 articles of the corpus were tagged. This way, named entity annotation was reduced to a manual correction task. Instead of creating fresh annotations, the annotators validated the automatically generated annotations by adding, removing and modifying annotations as necessary. This correction phase was carried out using the Manual Annotation Editor component of the Argo text mining platform [116]. The editor, supporting the creation and manipulation of text span annotations, was configured to allow the annotators to assign either a label of drug (\texttt{MDG}) or enzyme (\texttt{ENZ}) to named entities.

#### 4.2.2.2 Drug-enzyme interaction annotation

In this annotation task, we are interested in interactions between drugs and enzymes. For a span of text to qualify as an interaction description, three elements need to be present: the initiator of the interaction (agent), the receiver of said interaction (patient), and a word or sequence of words implying the association (interaction clue). Drug-target interactions are not necessarily direct associations. A drug, for instance, might be indirectly associated with an enzyme via its interaction with a biological process centred on the said enzyme, as in the example in Figure 4.2. In this task, therefore, whilst we are restricting the scope of agents to entities, we are taking into consideration patients in the form of entities or biological processes involving entities.
4.2.3 Annotation Scheme and Procedure

For reasons stated at the beginning of this chapter, we cast drug-target interactions as events in which each drug or enzyme may take the role of agent or patient. Event annotation, however, is a more sophisticated task than named entity (NE) or coreference annotation. Decomposing a span of text expressing a drug-target interaction into the elements of an event requires some linguistic background knowledge. Since our annotators are biochemists, we designed an event annotation scheme that makes the event annotation task simpler and requires minimum linguistic knowledge. The scheme calls only for the assignment of semantic labels, making the task similar to and as intuitive as named entity annotation. It is worth noting, however, that this scheme is viable in our case since the event types that we intend to capture are relatively simple, compared to events of interest in the BioNLP shared tasks, for instance.

After the completion of the named entity correction task by both annotators, a phrase extractor was run, taking as inputs sentences and annotated named entities. Each sentence was processed by Enju, a deep syntactic parser that produces phrase and predicate-argument structures [170]. For each drug-enzyme pair, consecutive tokens of the minimum subtree subsuming both entities are marked up as a candidate phrase describing a DTI and assigned the label `PHRASE`. This was done to allow the annotators to more easily find interactions between the two entities.

The annotators were asked to perform the following during interaction annotation:

1. Marking up spans of text signifying drug-target interaction clues and assigning them the label `INTERACTION`

2. Assigning the label `AGENT` to named entity (NE) annotations which act as initiators in DTIs

3. Assigning the label `TARGET` to NE annotations acting as recipients in DTIs
4. In cases where the target is a biological process, marking up the word(s) signifying the process as \texttt{BIOPROCESS}, and marking up the entire text span signifying the target (i.e., including the relevant NE) as \texttt{TARGET}.

5. Creation and manipulation of \texttt{PHRASE} annotations as necessary, and whilst keeping only the minimal phrase (the shortest span of text containing the agent, target and interaction clue) for each DTI.

One of the points emphasised in the annotation guidelines (provided in Appendix D) is the inclusion of inter-sentential drug-target interactions (i.e., DTIs described over multiple sentences). Hence, although preliminary phrase annotations were provided to them, the annotators were nevertheless asked to read the entire document to ensure that inter-sentential interactions are also captured.

Having these labelled text span annotations allowed us to automatically generate event frames using heuristics. To exemplify, we provide in Figures 4.4 and 4.5 examples of original annotations and the resulting equivalent event representation, respectively. The event representations were produced in the stand-off delimiter-separated values format of the \texttt{brat} rapid annotation tool (\texttt{brat}) \cite{191}. This allowed us, with periodic consultation with our annotators, to check the quality of the automatically generated events and make corrections as necessary using the said tool.

### 4.2.4 Results and Discussion

After validating event frames, we evaluated the agreement between the document sets from the two annotators. In performing this evaluation, the standard metrics of precision, recall and F-score were employed as they do not require the set of items for annotation to be identical between the curators, unlike other inter-annotator agreement measures (e.g., Cohen’s kappa coefficient \cite{207}). Our chosen metrics are more suitable for evaluating the annotations resulting from this effort in which the curators were
Efforts toward finding marine cyanobacterial metabolites with antitumor activity led to the isolation of bisebromoaamide (17) from a Lyngbya sp. harvested in Okinawa Prefecture (19). Bisebromoaamide (17) is featured by four unusual structural units, the 2-substituted thiazole-4-methyl-4-carboxylic acid unit fused to a methyl-proline, 2-(1-oxopropy) pyrrolidine (Opp) residues, N-methyl-bromo-tyrosine and N'-pyvalamide moieties. Prior to the isolation of bisebromoaamide (17), the Opp unit in had not been observed in any natural product. Bisebromoaamide (17) showed cytotoxicity against HeLa S3 cells (IC 50 = 0.04 μg/mL) and a panel of 30 human cancer cell lines (termed JFCR30) (the average GI 50 = 40 nM). At 10 to 0.1 μM, bisebromoaamide (17) selectively inhibited the phosphorylation of ERK (extracellular signal regulated protein kinase) in NRK (normal rat kidney) cells by PDGF (platelet-derived growth factor stimulation) however PKB, PKD, PKA, PLCγ1, (phospholipase Cγ1), or 56 ribosomal protein was not affected by bisebromoaamide (17) at the same concentration range. Bisebromoaamide (17) did not affect tubulin acetylation as other tubulin modulators. It is possible that bisebromoaamide (17) targets the ERK signal pathway which is activated in various cancers. Therefore, bisebromoaamide (17) has potential as a lead for anticancer drugs.

Figure 4.4: A drug-target interaction annotated using our scheme, as shown in the Manual Annotation Editor of Argo (*inhibited* as INTERACTION, *bisebromoaamide* as AGENT, *phosphorylation of ERK* as TARGET and *phosphorylation as BIOPROCESS*).

Figure 4.5: The equivalent event representation of the interaction annotations in Figure 4.4 generated by our heuristic-based converter, as visualised in the brat annotation tool.

Agreement was measured at two levels: (1) trigger identification and (2) argument identification. Shown in Table 4.3 are the results at each level, taking one of the annotation sets as gold standard and the other as response. The last row in the table shows the agreement when all event elements (i.e., triggers, arguments and argument roles) are compared between the two sets using exact matching, which requires that all text span boundaries and types correspond. The overall inter-annotator agreement on the corpus is 63%.

An annotation harmonisation step was introduced to produce the final version of
4.3. METHODOLOGY

Table 4.3: Inter-annotator agreement on drug-target interaction annotations

<table>
<thead>
<tr>
<th>Task</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger identification</td>
<td>70.40</td>
<td>74.38</td>
<td>72.34</td>
</tr>
<tr>
<td>Argument identification</td>
<td>87.15</td>
<td>87.01</td>
<td>87.08</td>
</tr>
<tr>
<td>Overall exact matching</td>
<td>61.35</td>
<td>64.71</td>
<td>62.99</td>
</tr>
</tbody>
</table>

Table 4.4: Frequencies in the HANAPIN DTI Corpus

<table>
<thead>
<tr>
<th>Set</th>
<th>Documents</th>
<th>Interactions</th>
<th>Bioprocesses</th>
<th>Inter-sentential interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>78</td>
<td>1,101</td>
<td>116</td>
<td>101</td>
</tr>
<tr>
<td>Development</td>
<td>18</td>
<td>258</td>
<td>54</td>
<td>16</td>
</tr>
<tr>
<td>Test</td>
<td>20</td>
<td>395</td>
<td>89</td>
<td>42</td>
</tr>
<tr>
<td>Total</td>
<td>116</td>
<td>1,754</td>
<td>259</td>
<td>159</td>
</tr>
</tbody>
</table>

The corpus. The first step towards this is the removal of documents from the corpus in which neither of the annotators found drug-target interactions. Left with 116 out of the 230 documents in the original set, we automatically combined the annotations. Conflicting event annotations were resolved by manual inspection and upon consultation with both annotators who were requested to come up with a consensus for each case.

The corpus documents, containing a total of 1,594 annotated interactions, were randomly split into training, development and test sets. Provided in Table 4.4 are the number of documents and annotated events in each subset.

4.3 Methodology

We developed a method for extracting drug-target interactions in the form of events, which can be formally defined with the templates below.

**Interaction** Cause:Drug|Enzyme, Theme:Enzyme|Drug|Bioprocess

**Bioprocess** Theme:Drug|Enzyme

In the above, the event type is specified as the first space-delimited field, whilst the rest of the line specifies event argument slots in a role:semantic_label format.
Semantic label is specified as a pipe-delimited list of entity types which can possibly take the corresponding role in the event. It is worth noting that although it is defined as having only a participant, a biological process is considered as an event because it inherently signifies change or action.

Our approach is pipeline-based, addressing each of the two subtasks of trigger and argument recognition in two separate steps. Trigger recognition was cast as a sequence labelling problem to which we applied the conditional random fields algorithm, whilst argument recognition was handled by a rule-based approach based on grammatical frame matching.

4.3.1 Trigger Recognition

4.3.1.1 Sequence labelling

The problem of recognising interaction triggers was cast as a sequence labelling problem, i.e., the automatic assignment of labels to a sequence of items. Each sentence corresponds to a sequence of items and was encoded using the begin-inside-outside (BIO) encoding. Provided in Example 4.1 are space-delimited sentence tokens in BIO encoding which specifies the first, succeeding and negative tokens making up event triggers with the labels \( B \), \( I \) and \( O \), respectively. It is worth noting that for this subtask, the triggers are not typed unlike those in the BioNLP shared tasks where each trigger is assigned a specific subtype label corresponding to the event type it signifies.

Example 4.1. Kim et al. reported the inhibitory effect of marine derived chitooligosaccharides (COS) on the activation and expression of matrix metalloproteinase-2 (MMP-2) in primary human dermal fibroblasts (HDFs).
4.3. METHODOLOGY

As mentioned previously in Section 2.2, the conditional random fields (CRF) algorithm has been shown to perform well in sequence labelling tasks, including biomedical named entity recognition [4, 5]. Based on the formulation of the algorithm, as discussed in Section 2.2.1, a CRF model finds the most probable label sequence given an observation sequence. The probabilities computed by the model are defined by features which are used to represent each item in the observation sequence.

4.3.1.2 Feature Set

Aside from the default features extracted by NERsuite (Section 2.2.2.1), our CRF implementation of choice, custom features were added to enrich the feature set. These consist of three feature types that, we believe, best allow a model to distinguish between tokens which comprise interaction triggers from those which do not.

**Dictionary features.** These consist of matches between token surface forms and entries in a dictionary. This feature type was added based on the intuition that a token whose surface form exists in a domain expert-curated trigger dictionary has a high probability of being (part of) a trigger. As events were observed to be centred on verbs, a dictionary of verbs (and their variants and related forms) was compiled from the BioLexicon, a large-scale lexical and terminological resource [208]. The BioLexicon is a repository which integrates various types of biomedical terms from diverse sources, and enriches the information on each term by collecting variants that appear in actual scientific literature. It also stores 658 verbs which have been observed to be frequently used in biomedical literature, together with their variants and derived forms (e.g., nominalisations). Furthermore, it contains grammatical frames curated from biomedical documents for 168 of the verbs. This subject will be touched upon further in Section 4.3.2.1.

Some of the 658 verbs have spelling and orthographic variants, examples of which
are *synthesize*, *synthesise*, *up-regulate*, *upregulate*, and *harbor*, *harbour*. Counting these variants as well, we have an initial list of 761 verbs. In forming our trigger dictionary, we gathered all of the inflections and derived forms available in the BioLexicon for each of these verbs. Provided in Table 4.5 are forms which were retrieved from the BioLexicon for a few specific examples. A total of 5,170 forms were collected in this manner.

Table 4.5: Examples of forms retrieved from the BioLexicon

<table>
<thead>
<tr>
<th>Verb</th>
<th>Variants</th>
<th>Inflected forms</th>
<th>Derived forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>inhibit</td>
<td>inhibit</td>
<td>inhibit, inhibits, inhibited, inhibiting</td>
<td>inhibiting, inhibition inhibitor, inhibitory</td>
</tr>
<tr>
<td>synthesize</td>
<td>synthesise</td>
<td>synthesise, synthesises synthesised, synthesising</td>
<td>synthesis, synthesisable synthetisation, synthesisability</td>
</tr>
<tr>
<td></td>
<td>synthesize</td>
<td>synthesized, synthesizing</td>
<td>synthesis, synthesizable synthetization, synthesizability</td>
</tr>
<tr>
<td>upregulate</td>
<td>upregulate</td>
<td>upregulate, upregulates, upregulated, upregulating</td>
<td>upregulatable, up-regulated, up-regulating, up-regulator, up-regulatory</td>
</tr>
<tr>
<td></td>
<td>up-regulate</td>
<td>up-regulate, up-regulated, up-regulating</td>
<td>up-regulatable, up-regulated, up-regulating, up-regulatory, up-regulator, up-regulatory</td>
</tr>
</tbody>
</table>

This initial set was then augmented by 114 additional entries derived from gold standard triggers in the HANAPIN DTI training set which do not overlap with any of the entries from the BioLexicon. These additional entries are provided in Appendix E.1 for the reader’s reference. The resulting 5,284 entries in the final trigger dictionary were normalised by converting all alphabetic characters to lowercase. The surface forms of tokens in input text were also converted to a normalised form in the same manner. Utilising the dictionary matcher bundled with NERsuite, we captured the longest possible matches between a sequence of normalised surface forms and entries in the compiled dictionary. Shown in Table 4.6 is an example of a token sequence whose BIO-encoded dictionary matches are provided in the third column.
Table 4.6: Example of a token sequence tagged with matches against our trigger dictionary.

<table>
<thead>
<tr>
<th>Token</th>
<th>Gold Std Label</th>
<th>BioLexicon match</th>
<th>NE label</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>inhibitory effect</td>
<td>B</td>
<td>B</td>
<td>O</td>
</tr>
<tr>
<td>of</td>
<td>I</td>
<td>B</td>
<td>O</td>
</tr>
<tr>
<td>marine derived chitooligosaccharides</td>
<td>O</td>
<td>B</td>
<td>O</td>
</tr>
<tr>
<td>(COS)</td>
<td>O</td>
<td>O</td>
<td>B-drug</td>
</tr>
<tr>
<td>)</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>on</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>the activation and expression of matrix metalloproteinase-2 )</td>
<td>O</td>
<td>B-drug</td>
<td>B-enzyme</td>
</tr>
<tr>
<td>MMP-2</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Unigrams and bigrams (as defined in Section 2.2.2.1) were then formed, both from the BIO labels as well as from the combination of the BIO labels with the corresponding surface form. With expression as the active token, i.e., the token we are currently extracting features for, and two as the maximum distance, the following features will be extracted by matching the token sequence in the first column of Table 4.6 against the trigger dictionary:

- Surface form and dictionary match n-grams:
  - Unigrams: \{activation:B\}, \{and:O\}, \{expression:B\}, \{of:O\}, \{matrix:O\}
CHAPTER 4. INTERACTION EXTRACTION

• Dictionary match n-grams:

  – Unigrams: \{B\}, \{O\}, \{B\}, \{O\}

  – Bigrams: \{B, O\}, \{O, B\}, \{B, O\}, \{O, O\}

Semantic type features. Since pre-annotated named entities are provided as input to the event extraction task, semantic type labels, encoded in BIO format, are available and can be leveraged as features for our trigger recogniser. Proposed based on the intuition that some interaction triggers are likely to be located close to names of drugs or enzymes, these features partially capture semantic context surrounding a token. The fourth column of Table 4.6 corresponds to the BIO-encoded semantic type labels for our example sentence. As with dictionary matches, unigrams and bigrams were generated out of the semantic type labels. Again using expression as our active token and two as the maximum distance, the following semantic type features will be generated based on the named entity annotations.

  • Unigrams: \{O\}, \{O\}, \{O\}, \{O\}, \{B-enzyme\}

  • Bigrams: \{O, O\}, \{O, O\}, \{O, O\}, \{O, B-enzyme\}

Co-occurrence features. As intuition tells us, interaction triggers are more likely to occur in sentences where drug and/or enzyme names are found. We introduced co-occurrence features to check for the presence of named entities in the same sentence as the active mention. These are boolean values corresponding to: (1) the presence of both a drug and an enzyme name, and (2) the presence of either a drug or an enzyme name in the same sentence as the token of interest.
4.3. METHODOLOGY

4.3.2 Argument Recognition

As previously mentioned, most events are centred around verbs and/or forms derived from them. Thompson et al. stipulated [208] that in most cases, semantic event participants or arguments are connected to the trigger by means of a syntactic relation. This is also the idea behind the dependency features used by most of the machine learning-based argument recognition methods which we have reviewed in Section 4.1.2.2. On the one hand, in many cases, the token expressing an event trigger has syntactic arguments which coincide with the event’s semantic arguments. Given that the word inhibited in the parse tree in Figure 4.6 is an event trigger, we can find the agent 2,4-dibromo-3,6-dihydroxyphenyl and the patient Pfnek-1 as its syntactic subject and object arguments, respectively, taking note that Pfnek-1 is the head of the subtree labelled as arg2 in the figure. In some cases, on the other hand, the token expressing a trigger is the syntactic argument of another token which is closely linked with one of the event’s arguments. The event trigger inhibition in Figure 4.7, for example, is one of the arguments of the preposition of, whose other argument is the patient. The syntactic subject and object of the preposition by (lightly shaded in grey), meanwhile, correspond to subtrees containing the event trigger and patient on one side, and the agent on the other, respectively.

This implies that there exist some patterns according to which syntactic arguments of triggers can be mapped to corresponding semantic event arguments. We decided to further explore this idea as a solution to our event argument recognition task. Realised as a rule-based method, our approach leverages a library of mappings which encode the patterns observed between syntactic and semantic arguments. In this section, we first give a discussion of syntactic and semantic frames and the mappings between them, followed by an overview of the library of frame mappings that we developed as a resource.
Figure 4.6: Example of an event centred on *inhibited* whose arguments correspond to the verb's syntactic arguments.
4.3. METHODOLOGY

4.3.2.1 Frames

The patterns according to which a certain verb (or any of its derived forms) takes its arguments are captured by means of templates known as frames.

**Syntactic frames.** Depending on their type, some verbs take specific types of syntactic arguments.

**Example 4.2.** Decadienal inhibits tubulin polymerization.

**Example 4.3.** DG42 synthesizes a Nod-like chitin oligosaccharide.

Whilst transitive verbs such as *inhibit* and *synthesize* always take direct objects (Examples 4.2 and 4.3), the intransitive verbs *co-occur* and *behave* do not (Examples 4.4 and 4.5).

**Example 4.4.** It has been previously reported that glycerol ethers co-occur with methyl ethers in the tissues.
**Example 4.5.** They behave as precursors of PAF.

Verbs like *arrive* and *migrate* often take prepositional phrases as arguments (Examples 4.6 and 4.7) whilst others like *observe* and *conclude* can take *that*-complement clauses (Example 4.8).

**Example 4.6.** We arrived at this conclusion based on our recent experiments.

**Example 4.7.** Developing germ cells of different phases migrate from the basal through the intermediate to the adluminal compartment.

**Example 4.8.** We showed that the second compound exhibits this biological activity to a greater extent.

These patterns are encoded using syntactic frames, such as the ones shown in Table 4.7. Entries in the second column of the table contain some of the most frequent combinations of syntactic arguments which have been observed for the given verb. Each frame entry in this table uses the hash symbol # to delimit the syntactic arguments. ARG1 and ARG2 correspond to syntactic subject and object, respectively, whilst arguments beginning with PP are prepositional phrases using the indicated preposition. Clauses which serve as a verb’s argument are represented in a frame with the first word in the clause (e.g., *that*) with CL appended in the end. An infinitive clause, however, is specified in a more detailed manner, i.e., as to-INF. It is worth noting that various computational lexica encode frames in different ways and that this is only one of them.

**Semantic frames.** Looking at several verbs, we observed that each is different in terms of the semantic arguments that it takes. Some verbs like *inhibit* and *activate* inherently mean that an actor is performing an action on a recipient, and hence, require an agent and a patient. Others, in contrast, often have an argument corresponding to a result, as in the case of the verb *converts* in Example 4.9. Such patterns in the semantic
arguments that a certain verb (or its derived form) can take, as well as their typical semantic types, are captured by semantic frames. We refer the reader to Table 4.8 for some specific examples. The second column of the table contains some of the most frequently observed combinations of semantic arguments for the specified verb. Delimited by the # symbol, each argument consists of a role-semantic type pair. The first semantic frame of the verb inhibit, for example, has two arguments: the first one is an agent of semantic type DNA and the second one is a theme which can be of any semantic type (indicated as O).

**Example 4.9.** The release of proteases converts the autolytic enzyme autolysin from a latent inactive form to an active state.
There are several resources, known as computational lexica, which store information pertaining to frames. Except for WordNet [163] (which does not contain syntactic frames), databases such as VerbNet [209], FrameNet [210] and PropBank [211] all contain syntactic and semantic frames acquired from general domain documents. For the biomedical domain, lexica such as BioFrameNet [212], SPECIALIST Lexicon [213], PASBio [214] and BioLexicon have been developed. They all contain both syntactic and semantic frames, except for the SPECIALIST Lexicon which does not store semantic frames. However, the BioLexicon is unique in terms of its storage of mappings between syntactic and semantic frames for a subset of the verbs.

<table>
<thead>
<tr>
<th>Verb</th>
<th>Semantic Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>inhibit</td>
<td>Agent@DNA#Theme@O &lt;br&gt;Agent@Experimental#Theme@DNA &lt;br&gt;Agent@Experimental#Theme@Protein &lt;br&gt;Agent@O#Theme@O &lt;br&gt;Agent@O#Theme@Protein &lt;br&gt;Agent@Protein#Theme@O &lt;br&gt;Agent@Protein#Theme@Processes &lt;br&gt;Agent@Experimental#Theme@O#Location@O &lt;br&gt;Agent@Experimental#Theme@Processes#Location@O &lt;br&gt;Agent@O#Theme@Protein#Location@O &lt;br&gt;Agent@Protein#Theme@Processes#Location@O</td>
</tr>
<tr>
<td>synthesis</td>
<td>Agent@O#Theme@Organisms &lt;br&gt;Agent@Organisms#Theme@Protein &lt;br&gt;Agent@Organisms#Theme@Protein#Condition@Experimental &lt;br&gt;Agent@Organisms#Theme@Protein#Condition@Processes &lt;br&gt;Theme@O &lt;br&gt;Theme@Protein &lt;br&gt;Theme@O#Condition@Experimental &lt;br&gt;Theme@Protein#Condition@Experimental &lt;br&gt;Theme@Protein#Condition@O &lt;br&gt;Theme@Protein#Location@O &lt;br&gt;Theme@O#Manner@O &lt;br&gt;Theme@Protein#Rate@O &lt;br&gt;Theme@O#Temporal@O &lt;br&gt;Theme@Protein#Temporal@O</td>
</tr>
</tbody>
</table>
4.3. METHODOLOGY

4.3.2.2 Syntactic-Semantic Frame Mappings

As stipulated by Thompson et al. [208], in most cases, correspondences can be observed between arguments in a semantic frame and those in a syntactic one. However, the correspondences vary from one verb class to another. As shown in Table 4.9, two distinct verbs having the same syntactic arguments differ in terms of their semantic arguments. Both increase and transport have the syntactic frame ARG1#PP-from#PP-to#. However, whilst the from and to prepositional phrases of the verb increase correspond to expressions signifying quantities, those of transport correspond to entities acting as source and destinations, respectively. For this reason, the BioLexicon was enriched with syntactic and semantic frames, and the correspondences or mappings between their arguments for each distinct verb in a set of 168 biomedical verbs.

Table 4.9: Example of varying correspondences between syntactic and semantic arguments.

<table>
<thead>
<tr>
<th>Verb</th>
<th>Syntactic Frame</th>
<th>Semantic Frame</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>increase</td>
<td>ARG1#PP-from#PP-to#</td>
<td>Theme#Quantity#Quantity</td>
<td>A was increased from B to C.</td>
</tr>
<tr>
<td>transport</td>
<td>ARG1#PP-from#PP-to#</td>
<td>Theme#Source#Destination</td>
<td>A was transported from B to C.</td>
</tr>
</tbody>
</table>

The syntactic frames stored in the BioLexicon were automatically captured from resulting syntactic parse results of the Enju parser [170] on a biomedical document collection containing over six million words. To remove noise from the extracted syntactic frames (i.e., those retrieved based on erroneous parse results), a filtering step was introduced, in which frames whose conditional probabilities are below a certain threshold were removed. Collecting semantic frames, in contrast, required some manual effort from the developers of the BioLexicon, since semantic parsers are not yet as reliable as syntactic ones. A corpus of 677 MEDLINE abstracts on gene regulation was hence manually annotated with event information which entailed the marking up
of semantic arguments of each event trigger. This was augmented with the 240 abstracts in the Gene Regulation Event (GREC) corpus [183]. From these gold standard annotations, semantic frames for each of the chosen verbs were extracted and stored in the BioLexicon.

A total of 1,760 syntactic and 856 semantic frames were acquired using the procedure described above. After examining the acquired syntactic and semantic frames for each of the 168 verbs, developers of the BioLexicon manually created links between them as necessary by following a set of rules [215]. In this manner, a total of 748 syntactic-semantic mappings were curated and stored in the BioLexicon, a few examples of which are provided in Table 4.10. In the interest of interoperability, we transcribed our semantic types and roles into those in the BioLexicon, as shown in Table 4.11.

The BioLexicon, however, contains frames for verbs but does not account for event triggers expressed using other parts-of-speech such as verb nominalisations. As we have earlier noted, interactions can also be expressed with nouns, e.g., *conversion*, *specificity* and *inhibitory effect*. For this reason, we decided to augment the frame library in the BioLexicon with our own extracted syntactic and semantic frames, covering such types of triggers.

As an initial step, syntactic parsing with Enju parser was performed on the documents in the HANAPIN DTI training corpus. We examined parse results to gain insight on how frames accounting for non-verb triggers can be automatically captured. It was noticeable in many cases, for instance, that triggers expressed as verb nominalisations are connected to event arguments through prepositions, such as in the example shown in Figure 4.7. To facilitate the extraction of their frames, and since we did not have access to the frame extractor employed in populating the BioLexicon [216, 215], we implemented our own and applied it on the documents in the training corpus. Taking
### Table 4.10: Examples of correspondences between syntactic and semantic arguments in the BioLexicon

<table>
<thead>
<tr>
<th>Verb</th>
<th>Syntactic Frame</th>
<th>Semantic Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>inhibit</td>
<td>arg1_ARG1#ARG2# Agent-Agent@DNA#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@DNA#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@Experimental#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Experimental#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@Experimental@O#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Experimental@O#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@O#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@O#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@O#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@O#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@Protein#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Protein#Theme@O</td>
<td></td>
</tr>
<tr>
<td>produce</td>
<td>arg1_ARG1#ARG2# Agent-Agent@DNA#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@DNA#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@DNA#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@DNA#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@O#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@O#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@Organism#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Organism#Theme@O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@Organism#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Organism#Theme@Protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2# Agent-Agent@Process#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Process#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td>convert</td>
<td>arg1_ARG1#ARG2# Agent-Agent@Protein#Theme@DNA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2# Theme-Agent@Protein#Theme@DNA</td>
<td></td>
</tr>
</tbody>
</table>

Inspiration from that of the BioLexicon, our extractor similarly takes into consideration passive usages and also employs the log likelihood score [217] to select only strongly correlated arguments (i.e., with scores $\geq 0.03$). However, aside from the collection of frames for verbs, our extraction mechanism can handle the recognition of syntactic and semantic frames for each of the gold standard event triggers in the corpus, even those appearing as nouns (e.g., verb nominalisations), adjectives and adverbs. Furthermore, it collected syntactic and semantic frames for each preposition in a pre-compiled list of 82 prepositions (provided in Appendix E.2). We then manually
Table 4.11: Transcription from our semantic types and roles to those in the BioLexicon.

<table>
<thead>
<tr>
<th>Types</th>
<th>HANAPIN DTI</th>
<th>BioLexicon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drug</td>
<td>Experimental</td>
<td></td>
</tr>
<tr>
<td>Enzyme</td>
<td>Protein</td>
<td></td>
</tr>
<tr>
<td>Bioprocess</td>
<td>Process</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Roles</th>
<th>Agent</th>
<th>Theme</th>
</tr>
</thead>
</table>

Table 4.12: Examples of correspondences between syntactic and semantic arguments retrieved based on the HANAPIN DTI training corpus

<table>
<thead>
<tr>
<th>Lemma</th>
<th>Syntactic Frame</th>
<th>Semantic Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>of</td>
<td>arg1_ARG1#ARG2#</td>
<td>Trigger_Trigger@O#Theme@Protein</td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2#</td>
<td>Theme_Trigger@O#Theme@Protein</td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2#</td>
<td>Trigger_Trigger@O#Theme@Process</td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2#</td>
<td>Theme_Trigger@O#Theme@Process</td>
</tr>
<tr>
<td>by</td>
<td>arg1_ARG1#ARG2#</td>
<td>Trigger_Trigger@O#Agent@Experimental</td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2#</td>
<td>Agent_Trigger@O#Agent@Experimental</td>
</tr>
<tr>
<td></td>
<td>arg1_ARG1#ARG2#</td>
<td>Trigger_Trigger@Process#Agent@Experimental</td>
</tr>
<tr>
<td></td>
<td>arg2_ARG1#ARG2#</td>
<td>Agent_Trigger@Process#Agent@Experimental</td>
</tr>
</tbody>
</table>

inspected the resulting syntactic and semantic frames for each entry of interest to find correspondences between their arguments. Out of the 124 additional mapping entries that we curated, 104 were from frames extracted for prepositions, some of which are shown in Table 4.12. The final library of syntactic-semantic frame mappings consists of a total of 872 entries.

### 4.3.2.3 Frame-matching Procedure

With a library of syntactic-semantic frame mappings at our disposal, we developed a rule-based procedure for extracting event arguments of triggers. As a pre-processing step, syntactic parsing is performed on input text using the Enju parser [170]. Our method initially attempts to find semantic arguments of any given trigger following the procedure outlined in Algorithm [1]. To exemplify the simple case, we refer the reader to Figure 4.6. Taking inhibit (the lemma of inhibited) as trigger \( t \), the procedure will find
2,4-dibromo-3,6-dihydroxyphenyl—inhibit—Pfnek-1 as the subject-object-triple τ (since Pfnek-1 is the head of the ARG2 subtree). The list T will have only τ in this case since neither 2,4-dibromo-3,6-dihydroxyphenyl nor Pfnek-1 is in any coordinations. Iterating through the mappings M for inhibit (shown in Table 4.10), we find that the argument types ARG1 and ARG2 of the syntactic frames of the first two entries match the argument types of our τ, since 2,4-dibromo-3,6-dihydroxyphenyl and Pfnek-1 are labelled as ARG1 and ARG2, respectively (Figure 4.6). Lastly, the corresponding semantic frame entries tell us that 2,4-dibromo-3,6-dihydroxyphenyl is the agent whilst Pfnek-1 is the theme.

We illustrate the case where the event trigger is not the predicate in the subject-predicate-object triple with the example in Figure 4.7, taking inhibition as the trigger t. Since none will be returned in attempting to retrieve a triple with t as the predicate, the procedure checks for triples with inhibition as a syntactic subject or object. Two such triples will be found: inhibition–of–Pfnek-1 and inhibition–by–2,4-dibromo-3,6-dihydroxyphenyl. The procedure proceeds to retrieving mappings for each of of and by, provided in Table 4.12. Matching the argument types in the frames for of, it will assign the theme role to ARG2 of inhibition, Pfnek-1. Similarly, after checking the argument types in the frames for by, the procedure will identify 2,4-dibromo-3,6-dihydroxyphenyl as an agent. It is worth noting that a semantic role is assigned to an argument only if it is an expression semantically labelled as drug, enzyme or trigger. Consequently, by default, a pronominal (e.g., it), sortal (e.g., the compound) and numerical expression (e.g., 1) is not assigned any semantic role even if it has been extracted as a trigger’s syntactic argument.

After gathering the semantic arguments of each trigger in a given sentence, we heuristically fill in slots in our Interaction event template, shown in Section 4.3. For each triple t which was assigned semantic argument roles in the procedure just described, our method first determines if it comprises a complete or partial interaction.
Algorithm 1 Assign semantic arguments to event trigger $e$

$\tau \leftarrow$ subject-predicate-object triple with $e$ as predicate

if $\tau = \emptyset$ then
    $\tau \leftarrow$ subject-predicate-object triple with $e$ as subject or object
end if

$T \leftarrow \tau$ and triples for arguments coordinated with $\tau$

for all $t \in T$ do
    $M \leftarrow$ mappings for predicate of $t$
    for all $m \in M$ do
        $s_1 \leftarrow$ syntactic frame of $m$
        $s_2 \leftarrow$ semantic frame of $m$
        if syntactic argument roles($t$) = argument roles($s_1$) then
            semantic argument roles ($t$) $\leftarrow$ argument roles($s_2$)
        end if
    end for
end for

We say that a triple is complete if it contains a trigger, an agent and a theme which is any of a drug, an enzyme or another trigger that already has its own theme assigned. In contrast, $t$ is considered partial if it is missing one of the three elements or if the theme is a trigger which does not have its own theme. With a complete $t$, as in our simple case exemplified above, it is straightforward to populate the slots in our Interaction template. A consolidation step, however, needs to be performed on a sentence’s partial instances upon validation that they complement each other, such as in the case of inhibition–of–Pfnek-1 and inhibition–by–2,4-dibromo-3,6-dihydroxyphenyl in our recent example. Another sentence in which consolidation of extracted triples is necessary is shown in Example 4.10. From this example, the triples $ST$–inducing–expressions and expressions–of–caspases are retrieved. It is quite straightforward to determine that the theme of the first triple corresponds to the biological process captured by the second triple, especially because of the overlapping argument expressions.

Example 4.10. ST has been shown to have chemopreventive effects on skin tumor development in mice by inducing expressions of caspases.
Even on an isolated partial instance, this consolidation step is carried out by checking the rest of the sentence for pre-annotated concepts which can occupy the missing slot. In Example 4.11, the partial triple *it–inhibited–chymotrypsin* where only *chymotrypsin* was assigned a semantic role, was merged with the drug name *salinosporamide A* to form the new triple *salinosporamide A–inhibited–chymotrypsin*. The complete triples are then used to populate slots in our templates.

**Example 4.11.** When salinosporamide A was tested against purified rabbit muscle 20S proteasome, it inhibited proteasomal chymotrypsin.

### 4.3.3 Argument Recognition with Coreferring Expressions

As previously mentioned in Section 4.2.4, 159 of the 1,754 (9%) interactions annotated in the HANAPIN DTI corpus are inter-sentential, in which at least one event argument is found in a sentence other than the one with the event trigger. However, our approach to event argument recognition is sentence-bound, i.e., it attempts to find arguments only within the same sentence as the trigger. Aiming to alleviate this problem, we applied our methods for coreference resolution presented in Chapter 3 to automatically annotate expressions co-referring with names of drugs and enzymes with the appropriate semantic label in order to increase the performance of event argument recognition.

#### 4.3.3.1 Mention Extraction

Since the input documents to the event extraction task have only named entities and triggers annotated, it is necessary to automatically recognise all references to entities of interest, e.g., pronominal and numerical references, as a preliminary step towards coreference resolution. A mention extraction method is expected to recognise, for instance, the expression *a promising lead compound* in the sentence in Example 4.12.
Example 4.12. DDMG-1, with its rare α-carboline structure, is a promising lead compound for the development of anti-TNF-α drugs.

We implemented our own mention extractor based on a variant of the conditional random fields (CRF) algorithm known as semi-Markov conditional random fields (semi-CRFs) [218]. They are different from classic CRFs in that they produce a segmentation of the input sequence of items by directly assigning labels to subsequences rather than to individual items. This can be attributed to its capability to learn features of item subsequences, instead of features of individual tokens as in the case of classic CRFs. This advantage of semi-CRFs is especially useful for mention extraction, since mentions of interest may consist of full noun phrases, i.e., long token sequences, as in Example 4.12. In such cases, a classic CRF model is likely to incorrectly label certain tokens in the mention. Although a CRF model also takes into account the context surrounding each token, it employs a predefined window which restricts it from capturing the features of the whole segment of interest. We hence cast the problem of mention extraction as a semi-CRF segmentation task in which a given sequence of tokens is segmented into mentions and non-mentions. This is illustrated with the sentence in Example 4.12, in which singly underlined segments correspond to non-mentions and doubly underlined ones are the extracted mentions.

The HANAPIN Coreference Corpus was utilised as our data set in developing the mention extractor. As an initial step, the corpus was split into ten equal subsets (i.e., folds), each one containing two documents. For each fold \( f \), mention predictions were generated using a semi-CRF model trained on the other nine subsets, and evaluated against the gold standard mentions for the documents in \( f \). In training the models, we employed Sunita Sarawagi’s semi-CRF implementation[^9] which, by default, utilises the following features[^10], given a segment of interest \( s \):

[^9]: http://crf.sourceforge.net
[^10]: We refer the reader to Section 2.2.2.1 for explanations of the terminology, e.g., unigrams.
Table 4.13: Micro-averaged mention extraction results using 10-fold cross validation on the HANAPIN Coreference Corpus

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-CRFs</td>
<td>90.786</td>
<td>87.727</td>
<td>89.230</td>
</tr>
<tr>
<td>CRFs</td>
<td>89.975</td>
<td>79.839</td>
<td>84.604</td>
</tr>
</tbody>
</table>

- Local segment features
  - unigrams of tokens in $s$
  - unigrams of collapsed word shapes in $s$

- Contextual segment features
  - unigrams of tokens within a maximum distance of three before and after $s$
  - unigrams of collapsed word shapes in $s$ within a maximum distance of three before and after $s$

After performing ten-fold cross validation on the corpus, we obtained the micro-averaged results in Table 4.13. Also included in the table are the results of ten-fold cross validation with classic CRFs using the same feature set as above. As expected, the semi-CRF model achieves better recall than the classic CRF one.

A final semi-CRF model for mention extraction was trained using all of the documents in the corpus. Mentions within documents in the HANAPIN DTI Corpus were automatically annotated by the application of the final model. The final set of mentions was then obtained by the getting union of automatically extracted mentions and gold-standard named entity annotations. In the event of partially overlapping cases, we retain the latter and discard the former.
4.3.3.2 Application of Coreference Resolution

In Section 3.4.4, we presented results of our evaluation of four different coreference resolution paradigms, namely, mention pair classification, mention pair ranking, mention-chain classification and mention-chain ranking. According to two out of three scoring schemes, the best F-score is obtained by mention pair classification employing our enriched set of features, described in Section 3.3.2. As none of the documents in the HANAPIN Coreference Corpus overlap with those in the DTI corpus, we decided to train a new mention pair classification model, employing our enriched feature set, on all of the documents in the former. This final model was incorporated into our mention pair classifier which was applied to the now mention-annotated documents in the HANAPIN DTI test corpus to automatically generate coreference chains. We developed a simple semantic type propagation mechanism that performs the routine in Algorithm 2:

Algorithm 2 Assign semantic type to mentions of interest in document $D$

\[
\text{\textbf{for all} sentence } s \in D \text{ do}
\]

\[
\text{if } s \text{ contains an event trigger then}
\]

\[
\text{\textbf{for all} mention } m \in s \text{ do}
\]

\[
c \leftarrow \text{coreference chain containing } m
\]

\[
t \leftarrow \text{most predominant semantic type in } c
\]

\[
\text{semantic type } (m) \leftarrow t
\]

\[
\text{end for}
\]

\[
\text{end if}
\]

\[
\text{end for}
\]

With this procedure, all mentions within a sentence containing an event trigger are assigned semantic types. Consequently, expressions which were previously semantically empty, e.g., pronouns and numerical references, now have semantic labels. This enables our frame-matching procedure, described in Section 4.3.2.3, to find such mentions as semantic arguments of event triggers. To exemplify, we revisit Example 4.11, from which the triple $\text{it–inhibited–chymotrypsin}$ was originally only partially complete
4.4. EVALUATION AND DISCUSSION

since it was not assigned a semantic role. If the expression it was extracted as a men-
tion and resolved to the drug salinosporamide A, our frame-matching procedure will
label it as an agent, making it–inhibited–chymotrypsin a triple which can be readily
converted into an Interaction event frame.

A post-processing step was performed after event frame creation, in which a slot
value which does not correspond to a named entity is replaced with the nearest named
entity that has the same semantic type. This step was introduced to allow for the eval-
uation of our automatically generated event frames against those in the gold standard
annotations.

4.4 Evaluation and Discussion

The HANAPIN DTI Corpus test set, consisting of 20 full-text articles, was used in
evaluating our approach to event extraction. As with our method for coreference reso-
lution, a series of pre-processing steps were performed on the documents, consisting of
sentence splitting, tokenisation, part-of-speech and chunk tagging, and syntactic pars-
ing. We refer the reader to Sections 2.3.1.1 and 3.4.2 for the specific details of each of
these steps.

In evaluating our approaches to the event extraction task, we used the standard
performance metrics of precision, recall and F-score. We refer the reader to Section 2.3
for the definition and details of computation of each of these metrics.

4.4.1 Trigger Recognition

A conditional random fields (CRF) model was trained on the the features discussed in
Section 4.3.1.2 which were extracted from samples in the HANAPIN DTI training set.
The trained model was evaluated on the test set of the HANAPIN DTI corpus. Shown
in Table 4.14 are the macro- and micro-averaged results of this evaluation.
Table 4.14: Evaluation on the HANAPIN DTI test set of trigger recognition model trained on training set

<table>
<thead>
<tr>
<th></th>
<th>Macro</th>
<th>Micro</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Default features</td>
<td>66.991</td>
<td>48.163</td>
</tr>
<tr>
<td>Enriched features</td>
<td>75.850</td>
<td>54.372</td>
</tr>
</tbody>
</table>

It is noticeable that the utilisation of our enriched feature set significantly improves the performance of the CRF-based approach to trigger recognition.

4.4.2 Argument Recognition

Applying the frame-matching procedure described in Section 4.3.2.3 to the named entity- and now trigger-annotated documents in our test set, we obtained the micro-averaged results in Table 4.15. The results for each of the interaction and bioprocess types are shown, as well as the average over both types. Also presented are results from the incorporation of coreference resolution into the argument recognition method.

We compare the performance of our proposed method against that of EventMine, which we reviewed in Section 4.1.2. In general, the performance of our rule-based approaches is competitive with that of the machine learning-based EventMine. Whilst EventMine achieved better precision across all categories, our frame-matching methods obtained better recall and balanced F-scores. The variant of the frame-matching method which takes into account co-referring expressions has especially achieved the best overall performance in terms of F-score.

We do recognise, however, that whilst such a rule-based method can be feasibly used to extract drug-target interactions, applying it to more complicated event types might not be as feasible. As illustrated in Section 4.3.2, a comprehensive set of hand-crafted syntactic-semantic frame mappings and constraints is crucial in the implementation of such approach. Another weakness of rule-based methods is the susceptibility
4.5. **SUMMARY**

Table 4.15: Summary of micro-averaged event extraction results on the HANAPIN DTI test corpus

<table>
<thead>
<tr>
<th>Method</th>
<th>Interaction</th>
<th>Bioprocess</th>
<th>Macro Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>EventMine</td>
<td>68.85</td>
<td>45.32</td>
<td>54.66</td>
</tr>
<tr>
<td>Frame-matching</td>
<td>62.80</td>
<td>47.79</td>
<td>54.28</td>
</tr>
<tr>
<td>Frame-matching+CR</td>
<td>64.29</td>
<td>53.77</td>
<td>58.56</td>
</tr>
</tbody>
</table>

to extract false positive instances. Nevertheless, in this work, we have demonstrated the feasibility of such a method on relatively simpler types of events such as drug-target interactions.

4.5 **Summary**

In this chapter, we presented our work on the extraction of drug-target interactions, which were cast as events. An overview of available corpora and previously reported approaches to event extraction was provided. We also share ideas, experiences and results from developing a corpus of full-text articles from the pharmaceutical chemistry domain, in which interactions between drugs and enzymes have been annotated as events. Taking a pipeline-based approach to event extraction, we decomposed the problem into two subtasks, namely, trigger and argument recognition. In recognising expressions pertaining to event triggers, an approach based on a conditional random fields model was taken. For the subsequent step of recognising event arguments, we proposed a rule-based approach that leverages a library of syntactic-semantic frame mappings to generate subject-predicate-object triples, which are in turn used to populate interaction and biological process event frames. Whilst a similar frame-matching method was applied to 80,000 raw documents in the UK PubMed Central (UK PMC) repository[^208], ours is the first study to have conducted a proper evaluation.

In an attempt to optimise performance, results from coreference resolution were incorporated into our method for argument recognition. Evaluating our approach and comparing the results against those of the machine learning-based EventMine, we can conclude that our frame-matching approach is feasible for the task of extracting drug-target interactions, and that the inclusion of results from coreference resolution improves overall performance.
Chapter 5

Conclusion

In this chapter, we revisit the research objectives we have established in Section 1.2 and briefly state how we fulfilled each one of them. We also share our plans and recommendations for extending our work in the future.

5.1 Evaluation of Research Objectives

To gain knowledge on the state-of-the-art in chemical and drug named entity recognition, we established objective O₁ to address research question Q₁ (What is the state-of-the-art in chemical and drug named entity recognition (NER))?:

\[ O₁ \text{ To conduct a comprehensive review of existing lexical resources, annotated corpora and approaches for chemical and drug named entity recognition.} \]

As an initial step towards achieving this objective, we analysed available lexical resources and presented our findings in Section 2.1.1. This enabled us to select PubChem Compound, CTD, DrugBank, ChEBI and Jochem as our domain knowledge sources. Examining each of the chemical and drug corpora known to us in Section 2.1.2 we
chose the following corpora as our data sets: CHEMDNER, SCAI and Metabolites for chemical NER, and DDI, PK and EU-ADR for drug NER.

By reviewing previously reported approaches to chemical named entity recognition (Section 2.1.3), we determined that the conditional random fields (CRF) algorithm has become one of the most commonly used approaches for NER. We also gained insight on the types of information which best capture domain knowledge, e.g., affixes, chemical segments and dictionary matches. As a whole, Chapter 2 of this thesis directly answers research question Q$_1$.

These results supported us in fulfilling our next objective:

$$O_2 \text{ To develop chemical and drug named entity recognisers which are enabled by domain knowledge.}$$

Section 2.2 describes our CRF-based approach to chemical and drug named entity recognition. In addition to the feature types typically employed in biomedical NERs, we introduced token attributes which capture domain knowledge. These include the presence of chemical prefixes and suffixes, the number of chemical basic segments, and matches against our five chosen chemical dictionaries (i.e., PubChem Compound, CTD, DrugBank, ChEBI and Jochem).

Implementing these ideas as well as some post-processing rules for capturing abbreviations and chemical segment composition, we developed our own chemical and drug named entity recognisers, as described in Section 2.3. Our chemical NER was thoroughly evaluated on several corpora, e.g., CHEMDNER test set, SCAI pilot corpus and the Metabolites corpus. Furthermore, a comparison of its performance against that of state-of-the-art NER tools such as OSCAR 4 and ChemSpot was carried out. Results show that our chemical NER consistently outperformed the other NERs on the
5.1. EVALUATION OF RESEARCH OBJECTIVES

benchmark corpora. Similarly, our drug NER outperformed the best performing system in the DDI Extraction DrugNER shared task, having been evaluated on the DDI test data. Based on the results we presented, we can conclude that we have succeeded in developing chemical and drug NERs, especially because of the domain knowledge we have incorporated by means of custom features and post-processing rules. With these methods, we have addressed our second research question $Q_2$ (What improvements can we introduce to existing approaches in order to achieve better performance in chemical and drug NER?). Our work on named entity recognition was also validated by the top-performing results we achieved upon applying our methods on the chemical document indexing and CTD curation tasks in the Fourth BioCreative Challenge Evaluation Workshop in which we participated [219, 220].

We established the following objective as an initial step towards our study on coreference resolution, and in attempting to address the third research question $Q_3$ that we have raised (What are the existing approaches to general and biomedical coreference resolution):

$O_3$ To conduct a comprehensive review of existing corpora and approaches for general and biomedical coreference resolution.

In Section 3.1.1 different anaphora- and coreference-annotated corpora, from both the general and biomedical domains, were examined. Out of the various coreference types which were identified, the direct nominal type (i.e., coreference between noun phrases related by identity) was chosen as the focus of our study. It was also established that most corpora were annotated using annotation schemes based on pairwise linking, i.e., by treating coreferring expressions as anaphor-antecedent pairs. However, from our review of various corpora, we determined that there are no available coreference-annotated corpora consisting of scientific documents from our domain of
interest. Since previous studies have shown that it is desirable to utilise full-text articles in investigating coreference resolution, we decided to develop the HANAPIN Coreference Corpus, a collection of 20 full articles on marine natural products chemistry which were annotated by three domain experts with coreference information (Section 3.2). The outcome of this corpus development effort was reported in the BioNLP Workshop co-located with the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies [221].

A review of existing approaches to both general and biomedical coreference resolution was also completed and presented in Section 3.1.2. Whilst traditional approaches focussed on rule-based methods for pronominal resolution, supervised approaches were subsequently proposed for general coreference resolution which were then, later on, explored for the biomedical domain. They can be categorised into four paradigms, namely, mention pair classification, mention pair ranking, mention-chain classification and mention-chain ranking.

These findings directly answered research question Q₃ and at the same time guided us in accomplishing our next research objective:

\[ O₄ \quad \text{To develop implementations of different approaches to coreference resolution and to adapt them for the pharmaceutical chemistry domain.} \]

Each of the four paradigms mentioned above was implemented, as explained in Section 3.3. In mention pair classification, a pair of mentions is presented at a time to a binary classifier, which predicts whether the mentions are coreferent or otherwise. In contrast, a ranker induces an order over mention pairs to capture the competition amongst several candidates in one step. Mention-chain classification is similar to mention pair classification, except that each instance presented to the classifier consists of
5.1. EVALUATION OF RESEARCH OBJECTIVES

a mention and a chain of mentions. Mention-chain ranking, whilst using the same instance representation as mention-chain classification, is different in that it attempts to impose a ranking over several instances.

In training and evaluating the support vector machine-based classifiers and rankers, we proposed several new features (Section 3.3.2) to capture characteristics of expressions often used in documents from the pharmaceutical chemistry domain, e.g., token matches with hyponyms of chemical compound and numerical references. To evaluate the performance of the paradigm implementations, two rounds of experiments were performed with each implementation, one round employing our newly proposed features and the other only traditionally used ones. Performance was measured by means of three different scoring schemes, namely, MUC, B³ and BLANC, since there is no consensus within the NLP community yet as to which of the coreference scoring schemes is most acceptable. To answer research question Q⁴ (Which amongst different approaches to coreference resolution is most suitable for the pharmaceutical chemistry domain?), we note that according to two out of the three scoring schemes, the best balanced F-score was obtained by the mention pair classification implementation that utilised our proposed features. These results were reported in the Fourth International Symposium on Languages in Biology and Medicine [222].

The following objective was also established in the beginning of this research, to explore event extraction as a response to our fifth research question Q⁵ (How can the task of extracting drug-target interactions be accomplished?):

O⁵ To develop an event extraction methodology for capturing drug-target interactions.

We accomplished this objective by dividing the event extraction problem into two subtasks, namely, trigger recognition and argument recognition (Section 4.3). Cast as a
sequential labelling task, trigger recognition was addressed using a conditional random fields model trained on dictionary, co-occurrence and semantic type features. For argument recognition, a rule-based method was developed, in which semantic arguments of an event trigger are found in given text by means of matching its syntactic arguments against a library of syntactic-semantic frame mappings. The library of mappings was built from the combination of curated frames in the BioLexicon and frames extracted from the HANAPIN Drug-Target Interaction (DTI) Corpus training set. Evaluating our event extraction on the HANAPIN DTI test documents, we demonstrated that our event extraction methodology obtains competitive results compared with a machine learning-based approach. To the best of our knowledge, ours is the first investigation that explores and evaluates such methodology as a solution for event extraction. We showed that our proposed method can effectively extract drug-target interactions, thus directly addressing research question Q5. Aiming to improve event extraction results, we also set the following research objective:

\[ O_6 \] To incorporate coreference resolution into our event extraction methodology and observe its impact on performance.

In order to fulfill this objective, we incorporated results from coreference resolution into our argument recognition approach. Initially, a mention extraction method based on semi-Markov conditional random fields was developed and applied on the test documents in order to automatically annotate sortal, pronominal and numerical expressions of interest. Employing the mention pair classifier we developed as part of our study on coreference resolution, coreference chains were generated for each test document, which allowed us to assign semantic labels to previously semantically empty expressions. With such expressions, our event argument recognition method was able to identify semantic arguments of events described over multiple sentences. This was
validated by our event extraction evaluation results. Addressing research question Q₆ (
Does the incorporation of coreference resolution into interaction extraction improve
the performance of the latter?), we demonstrated that the performance of interaction
extraction is improved, evidenced by an increase in F-score of at least two percentage
points, with the incorporation of results from coreference resolution (Section 4.4).

After having fulfilled our research objectives, we obtained the findings summarised
above which prove the research hypotheses that we formulated in the beginning:

H₁ Existing methods in BioNLP can be adapted to information extraction from phar-
maceutical chemistry literature with the incorporation of domain knowledge

H₂ Interactions between drugs and targets can be cast as events, and that they can
be extracted using an event-based approach

5.2 Future Work

The work presented in this thesis can be advanced by exploring its feasibility in drug
discovery use cases or applications. These may be in the form of feasibility studies
evaluating the applicability of the proposed drug-target interaction extraction methods
to semi-automatic chemoinformatic database curation and drug screening. Another
potential application is drug repurposing, in which novel drug uses are proposed based
on indirect associations extracted from the literature. The methods presented in this
study can also be applied to the extraction of pharmacokinetic parameters which are
expressed in text within the context of interactions, e.g., between a drug and an en-
zyme, or between two drugs. For each distinct use case, different event types possibly
need to be defined with the help of domain experts. Consequently, the rule-based event
argument extraction method will have to be revisited to accommodate such additions
and modifications.
Bibliography


[50] C. Mihăilă and R.T. Batista-Navarro. What’s in a Name? Entity Type Variation across Two Biomedical Subdomains. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for*


random fields: Probabilistic models for segmenting and labeling sequence data.
In *Proceedings of the Eighteenth International Conference on Machine Learn-
ing*, ICML ’01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kauf-
mann Publishers Inc.


[109] K.-J. Lee, Y.-S.k Hwang, S. Kim, and H.-C. Rim. Biomedical named entity
recognition using two-phase model based on SVMs. *Journal of Biomedical

[110] J. Hastings, P. de Matos, A. Dekker, M. Ennis, B. Harsha, N. Kale, V. Muthukr-
ishnan, G. Owen, S. Turner, M. Williams, and C. Steinbeck. The ChEBI refer-
ence database and ontology for biologically relevant chemistry: enhancements


[112] A.S. Schwartz and M.A. Hearst. A Simple Algorithm for Identifying Abbrevi-
ation Definitions in Biomedical Text. In *Pacific Symposium on Biocomputing*,

[113] S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (com-

Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction
2013). In *Second Joint Conference on Lexical and Computational Semantics*


of the Fourth International Symposium on Languages in Biology and Medicine, LBM ’11, 2011.
# Appendix A

## Chemical resources

### A.1 List of chemical affixes

<table>
<thead>
<tr>
<th>Prefixes</th>
<th>size 2</th>
<th>size 3</th>
<th>size 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>di-</td>
<td>oxo-,</td>
<td>(R)–,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>oxy-,</td>
<td>amin-,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tri-</td>
<td>azid-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>brom-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>carb-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>chlo-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>cyan-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>deca-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>diaz-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>diha-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dihy-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fluo-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>form-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hal-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hept-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hexa-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hydr-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>imin-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>iodo-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>iody-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mono-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nitr-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nona-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>octa-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>pent-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>perc-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>phen-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sulf-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tetr-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Suffixes</th>
<th>size 2</th>
<th>size 3</th>
<th>size 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-al,</td>
<td>-ane,</td>
<td>-acid,</td>
</tr>
<tr>
<td></td>
<td>-ol,</td>
<td>-ate,</td>
<td>-enyl,</td>
</tr>
<tr>
<td></td>
<td>-yl</td>
<td>-ene,</td>
<td>-hiol,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-ene,</td>
<td>-hyde</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-io-,</td>
<td>-mide,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-l,</td>
<td>-mine,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-ic,</td>
<td>-oate,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-one,</td>
<td>-rile</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-yde,</td>
<td>-ther,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-yne</td>
<td>-ylic,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-ynyl,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-zene</td>
</tr>
</tbody>
</table>

### A.2 List of chemical element symbols

Ac, Ag, Al, Am, Ar, As, At, Au, B, Ba, Be, Bh, Bi, Bk, Br, C, Ca, Cd, Ce, Cf, Cl, Cm, 
Cn, Co, Cr, Cs, Cu, Db, Ds, Dy, Er, Es, Eu, F, Fe, Fm, Fr, Ga, Gd, Ge, H, He, Hf, Hg, 
Ho, Hs, I, In, Ir, K, Kr, La, Li, Lr, Lu, Md, Mg, Mn, Mo, Mt, N, Na, Nb, Nd, Ne, Ni, 
No, Np, O, Os, P, Pa, Pb, Pd, Pm, Po, Pr, Pt, Pu, Ra, Rb, Re, Rf, Rg, Rh, Rn, Ru, S,
Sb, Sc, Se, Sg, Si, Sm, Sn, Sr, Ta, Tb, Tc, Te, Th, Ti, Tl, Tm, U, V, W, Xe, Y, Yb, Zn, Zr
Appendix B

HANAPIN Coreference Corpus

annotation resources

B.1 Annotation guidelines

B.1.1 Named entity annotation

B.1.2 Introduction to the task and annotation types

Our main goal in doing this named entity annotation task is to mark-up all relevant chemical and biological entities. Table B.1 enumerates the types of entities that need to be annotated.

B.1.3 Annotation Elements

There are three different ways by which named entities appear in text.

- Continuous. The named entity appears inside a noun phrase, as a token or consecutive tokens.
Haliclonyne (1), a new polyacetylene carboxylic acid, has been isolated from the marine sponge Haliclona sp. collected in the Gulf of Eilat.

- Discontinuous. Named entities appear in discontinuous text regions, usually when they are coordinated and share a fragment. In the following example, the text *palythoalones A and B* has the two named entities *palythoalone A* and *palythoalone B*.

Two new ecdysteroids, *palythoalones A and B*, have been isolated from the marine zoanthid *Palythoa australiae*.

- Embedded. The named entity is embedded within another named entity. In the following example, the entities *human* (an organism) and *gastric tumor* (a disease) are embedded within the name of a cell line (*human gastric tumor cells*).

Among them, *durissimol B* (2) and duryne showed potent cytotoxicity against *human* *gastric tumor* cells.
Table B.1: Named entity types

<table>
<thead>
<tr>
<th>Type</th>
<th>Annotation label</th>
<th>Description/Coverage</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHEMICAL ENTITIES (chem)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systematic name of compound</td>
<td>chem</td>
<td>IUPAC and IUPAC-like names of chemical compounds</td>
<td>3-amino-2-methylhexanoic acid, sphaerolabdadiene-3,14-diol</td>
</tr>
<tr>
<td>Trivial/group name of compound</td>
<td>chem</td>
<td>common/generic names, family/group names</td>
<td>palythoalone A, paclitaxel, alkaloid, steroid</td>
</tr>
<tr>
<td>Related form</td>
<td>chem</td>
<td>related chemical compound forms</td>
<td>jaspamide derivatives, dibromo analogue</td>
</tr>
<tr>
<td>Chemical abbreviation</td>
<td>chem</td>
<td>chemical abbreviations</td>
<td>MeOH, EtoAc</td>
</tr>
<tr>
<td>Molecular formula</td>
<td>chem</td>
<td>molecular formulas</td>
<td>HCl, CH2Cl2</td>
</tr>
<tr>
<td>Chemical-pertaining term</td>
<td>pertains_to_chem</td>
<td>terms whose base forms pertain to chemical entities</td>
<td>brominated, olefinic, acetylation</td>
</tr>
<tr>
<td>ORGANISMS (organism)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scientific name of organism</td>
<td>organism</td>
<td>Latin names of organisms</td>
<td>Tetilla japonica, Callyspongiidae</td>
</tr>
<tr>
<td>Common name of organism</td>
<td>organism</td>
<td>common names of organisms</td>
<td>sponge, red algae, mice</td>
</tr>
<tr>
<td>Pathogen</td>
<td>organism</td>
<td>names of bacteria, viruses; usually for naming a group</td>
<td>Gram-positive bacteria, LCM-lassa complex viruses</td>
</tr>
<tr>
<td>Organism-pertaining term</td>
<td>pertains_to_org</td>
<td>terms whose base forms pertain to organisms</td>
<td>fungal, bacterial, murine</td>
</tr>
</tbody>
</table>
### Named entity types (continuation)

<table>
<thead>
<tr>
<th>Type</th>
<th>Annotation label</th>
<th>Description/Coverage</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DRUG TARGETS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anatomical part</td>
<td>anatomical_part</td>
<td>organs and tissues</td>
<td>heart, skin, epithelial tissue</td>
</tr>
<tr>
<td>Cell</td>
<td>cell</td>
<td>names of cells and cell types</td>
<td>lymphocyte, hepatocyte</td>
</tr>
<tr>
<td>Cell line</td>
<td>cell_line</td>
<td>names of experimentally modified cells</td>
<td>MCF-7, HL-60</td>
</tr>
<tr>
<td>Cell part</td>
<td>cell_part</td>
<td>cell parts</td>
<td>cytoplasm, microtubule</td>
</tr>
<tr>
<td>Gene</td>
<td>gene_protein</td>
<td>names of genes</td>
<td>DEC1, ACP1</td>
</tr>
<tr>
<td>Protein</td>
<td>gene_protein</td>
<td>names of proteins</td>
<td>cytokines, TNF-alpha</td>
</tr>
<tr>
<td>Enzyme</td>
<td>enzyme</td>
<td>names of enzymes</td>
<td>luciferase, kinase</td>
</tr>
<tr>
<td><strong>BIOLOGICAL ACTIVITIES</strong> (biological_activity)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td>biological_activity</td>
<td>pertains to an entity’s biological or pharmacological role</td>
<td>antibiotics, elastase inhibitor chemopreventive agent</td>
</tr>
<tr>
<td>Biological action</td>
<td>bioprocess</td>
<td>pertains to a biological process, effect or action</td>
<td>cytotoxicity, angiogenesis</td>
</tr>
<tr>
<td>Bioactivity-pertaining term</td>
<td>pertains_to_bioactivity</td>
<td>a modifier pertaining to a biological activity</td>
<td>cytotoxic, antibacterial</td>
</tr>
<tr>
<td><strong>DISEASES</strong> (disease)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disease</td>
<td>disease</td>
<td>names of diseases and disorders</td>
<td>breast carcinoma, Alzheimer's disease, Diarrheic Shellfish Poisoning</td>
</tr>
<tr>
<td>Disease-pertaining term</td>
<td>pertains_to_disease</td>
<td>modifiers pertaining to diseases</td>
<td>nosocomial, mycobacterial</td>
</tr>
</tbody>
</table>
Considering these variations in how named entities appear in text, the following annotation elements have been defined.

**B.1.3.1 term**

This element is for marking up continuous terms.

```
<sentence id="S1">
    Two new<term id="T1">jaspamide derivatives</term> along with ...
</sentence>
```

**B.1.3.2 cons**

This element is for annotating discontinuous terms. The entire text region is placed inside a cons element; each of the individual named entities is marked up using a term element (or an encterm element).

```
<sentence id="S1">
    ... <cons id="C1">
        <term id="T1">palythoalones A</term> and
        <term id="T2">B</term>
    </cons>
    , have been isolated ...
</sentence>
```

**B.1.3.3 frag**

After the annotation of discontinuous terms using the cons and term elements as previously mentioned, the fragment common to the coordinated named entities is marked up using a frag element.
... <cons id="C1">
  <term id="T1">
    <frag id="F1">palythoalones</frag> A
  </term> and
  <term id="T2">B</term>
</cons>
, have been isolated ...
</sentence>

B.1.3.4 encterm

This element is for the annotation of named entities with embedded terms. Each of
the embedded named entities is first annotated using the term element. The enclosing
named entity (the one that contains the embedded term(s)) is then marked up using the
encterm element.

... against
  <encterm id="E1">
    <term id="T1">human</term>
    <term id="T2">gastric tumor</term>
    cells
  </encterm>

</sentence>
B.1.4  Guidelines and Examples

B.1.4.1  Markables

1. In determining which terms need mark-up, the annotator should always refer to Table B.1 for the types of entities which are of interest.

2. For our current task, anything which does not fall under any of the given types does not need annotation. The following example needs no annotation as none of the named entities fall under any of our types of interest.

   The structure elucidations of 1 and 2 were accomplished by means of 1D and 2D NMR, MS, and IR spectroscopy.

3. One possible source of confusion is the mark-up of biological activities. The rule of thumb in the case of biological activities is that only terms which have inherent biological meaning should be annotated. Shown below are two examples to demonstrate this.

   The compound showed anti-tumor activities against the cell line.

   The compound showed inhibition of the cell line.

   The two sentences above express the same meaning but in different ways. When presented with these sentences, the annotator does not need to mark-up anything in the second sentence whilst anti-tumor activities in the first sentence should be annotated and assigned bioprocess as its type. The expression anti-tumor activities inherently means activity against a tumor, and thus has a biological meaning. In contrast, inhibition in the second sentence by itself does not have
any biological meaning. The phrase *inhibition of the cell line* does, but we do not consider such a phrase a markable during named entity annotation.

**B.1.4.2 Assigning a type**

1. Each named entity that is marked up (as described in the previous section, be it a `term`, `cons` or `encterm` element) has to be assigned an entity type, as in the following example where the `sem` attribute of the expression *jaspamide derivatives* was assigned the value of `chem`.

   <sentence id="S1">
   Two new
   <term id="T1" sem="chem">
   jaspamide derivatives
   </term>
   along with ...
   </sentence>

2. All of the annotation labels listed in Table [B.1] are available as options to the annotator. Please refer to the instructions for using XConc for specific details on assigning a type to a `cons` element.

**B.1.4.3 Assigning a lexeme**

1. Each named entity that is marked up (as described in the previous section, be it a `term`, `cons` or `encterm` element) has to be assigned a lexeme for the `lex` attribute. The lexeme of a named entity is its form without any inflections, i.e., modifications brought about by tense and case. In the following example, where the `lex` attribute of the expression *jaspamide derivatives* was assigned the value *jaspamide derivative*, encoded as `jaspamide_derivative`. 
Two new jaspamide derivatives along with ...

2. Please refer to the instructions for using XConc for specific details on assigning a lexeme value to a cons element.

**B.1.4.4 Abbreviations**

1. Only abbreviations which correspond to entity types of interest should be annotated. In the following examples, the abbreviations should not be annotated as they correspond to names of techniques which are not within the scope of this annotation task.

   Structure elucidation required extensive application of 2-D NMR techniques such as COSY, HMQC, HMBC, and ROESY.

2. Such abbreviations, except for those whose full forms cannot be found in the text (usually chemical abbreviations), are assigned the type of its full form.

   The dinoflagellate Prorocentrum lima produces toxins involved in the red tide phenomenon known as diarrhetic shellfish poisoning (DSP). This paper reports the isolation and spectroscopic structural elucidation of new compounds related to DSP toxins, isolated from a laboratory culture of strain
In the given example, DSP is the abbreviation for diarrhetic shellfish poisoning, a disease. Both instances of DSP should then be assigned the disease type.

3. In cases where the full form of a chemical abbreviation is not found anywhere in the text, the chem type should be chosen.

B.1.4.5 Merging

Given the following sentence:

Bioassay-guided fractionation led to the isolation of a known brominated acetylenic fatty acid 1 as the active component.

One would initially annotate the sentence in the following manner:

```xml
<sentence id="S1">
  a known
  <term id="T1" sem="pertains_to_chem">brominated</term>
  acetylenic
  <term id="T3" sem="chem">fatty acid
</sentence>
```
This is correct as *brominated* and *acetylenic* both pertain to chemical terms. They are used as modifiers of *fatty acid*, a group name. However, the modifiers and the head noun all fall under the entity type chem. In such cases, the modifiers can be merged with the head noun.

```xml
<sentence id="S1">
... a known
<term id="T1" sem="chem">
  brominated acetylenic fatty acid
</term>
1 as the ...
</sentence>
```

### B.1.4.6 Complex named entities

One might encounter complex named entities, i.e., those which have both embedded and coordinated entities. In such cases, the encterm and cons elements can be used together.

Both compounds 1 and 2 inhibited the growth of MALME-3M melanoma and MCF-7 breast cancer cell lines in the range 30-40 microg/mL.

*Melanoma* and *breast cancer* are both names of diseases while *MALME-3M* and *MCF-7* are names of cell lines. *Breast* is an anatomical part which will be embedded within *breast cancer*. Coordinated by the conjunction *and* are *MALME-3M melanoma cell line* and *MCF-7 breast cancer cell line*. Below are the recommended annotations for the given example.
... the growth of
<cons id="C1"
    sem="AND MALME-3M_melanoma_cell_line MCF-7_breast_cancer_cell_line">
    <encterm id="E1" sem="cell_line">
        <term id="T1" sem="cell_line">
            MALME-3M
        </term>
        <term id="T2" sem="disease">
            melanoma
        </term>
    </encterm>
    <encterm id="E2" sem="cell_line">
        <term id="T3" sem="cell_line">
            MCF-7
        </term>
        <term id="T4" sem="disease">
            breast cancer
        </term>
    </encterm>
    <frag id="F1">
        cell lines
    </frag>
</cons>
in the range ...
B.2 Coreference annotation

We are also interested in the annotation of linguistic expressions which refer to the same entity, which we call coreferences. Co-referring mentions will be linked pairwise, i.e., by connecting a referring expression to the entity it refers to, which is called referent.

Seven cytotoxic 3-alkylpyridine alkaloids, hachijodines A-G, have been isolated from two marine sponges of the genera Xestospongia and Amphimedon. These alkaloids are moderately cytotoxic against P388 murine leukemia cells with IC(50) values of 1.0-2.3 microg/mL.

In the example above, these alkaloids is the referring expression and hachijodines A-G is its referent.

B.2.1 Introduction to the task and annotation types

Our main goal in doing this coreference annotation task is to mark-up mentions and to link together those which are coreferent. We categorise coreference types in terms of their lexical features.

B.2.1.1 Sortal types

Sortal coreferences (sometimes also known as lexical noun phrase coreferences) are characterised by a noun phrase consisting of a head noun and its modifiers. It is further divided into the following subtypes: definite, indefinite, demonstrative, distributive, predicate nominative and apposition.

Definite noun phrase (definite). This type applies to a definite noun phrase (i.e., a noun phrase starting with the article the) that refers to another noun phrase.
A suite of amphiphilic siderophores, loihichelins A-F, were isolated from cultures of the marine bacterium Halomonas sp. The loihichelins contain a hydrophilic headgroup consisting of an octapeptide comprised of d-threo-beta-hydroxyaspartic acid, d-serine, l-glutamine, l-serine, l-N(delta)-acetyl-N(delta)-hydroxyornithine, dehy-droamino-2-butyric acid, d-serine, and cyclic N(delta)-hydroxy-d-ornithine, appended by one of a series of fatty acids ranging from decanoic acid to tetradecanoic acid.

Note that a definite noun phrase can also be presented as a predicate nominative:

Sarasinosides J (1) and K (2) are the 24,25- hydrogenated congeners of the previously described sarasinosides A1 and H2, respectively.

**Indefinite noun phrase (indefinite).** This type applies to an indefinite noun phrase (i.e., a noun phrase used to introduce something new to the reader, e.g., *a compound, three compounds, all compounds*) that refers to another noun phrase.

A new mycalamide, mycalamide D (3), has been isolated from the New Zealand marine sponge Mycale sp. This new metabolite, in which the C13-O-methyl group of mycalamide A (1) is replaced by a hydrogen atom, was found to be cytotoxic to a range of mammalian cell lines, with a potency approximately 20-fold less than that of 1.

Note that an indefinite noun phrase can also be presented as a predicate nominative:
Thus, antiproteinide is a natural product metabolite of S. tropica.

**Demonstrative noun phrase** (demonstrative). This type applies to a demonstrative noun phrase (i.e., a noun phrase starting with a demonstrative pronoun such as this, these, that or those) that refers to another noun phrase.

A new mycalamide, mycalamide D (3), has been isolated from the New Zealand marine sponge Mycale sp. This new metabolite, in which the C13-O-methyl group of mycalamide A (1) is replaced by a hydrogen atom, was found to be cytotoxic to a range of mammalian cell lines, with a potency approximately 20-fold less than that of 1.

**Distributive noun phrase** (distributive). This type applies to a distributive noun phrase (i.e., a noun phrase starting with both, such, each, either or neither used as an adjective) that refers to another noun phrase.

Secobatzellines A and B inhibited the phosphatase activity of calcineurin, and secobatzelline A inhibited the peptidase activity of CPP32. Both compounds showed in vitro cytotoxicity against P-388 and A-549 cell lines.

Note that this type does NOT cover the case when both, each, either or neither is used as a pronoun, as in the following example:

The new diterpenes possess benzoate and senecioate substitutents, both of which are rare among marine natural products.
In which case, the pronominal type applies with both as the referring expression and which as its referent. Which should be, later on, marked as a referring expression of diterpenes.

**Predicate nominative** (pred_nom). This type applies to a noun phrase appearing after a copular verb (i.e., a verb that does not indicate any action, e.g., inflections of be and appear), which renames the subject of the sentence.

\[
\text{Sargassum carpophyllum} \text{ from the South China Sea is the source of two new bioactive sterols.}
\]

The noun phrase the source which appears after the linking verb is is a predicate nominative describing the subject Sargassum carpophyllum.

**Appositive** (appos). This type applies to a noun phrase that renames another noun phrase beside it.

\[
\text{Radiosumin B (1), an N-methyl dipeptide containing two unusual amino acid residues, was isolated from the cyanobacterium Microcystis aeruginosa.}
\]

An appositional phrase can occur in the same noun phrase as the referent. In the following example, the expression the Formosan soft coral is in an appositional relation with the species name Clavulariaviridis.

\[
\text{Three new cytotoxic prostanoids were isolated from the methylene chloride solubles of the Formosan soft coral Clavulariaviridis.}
\]
B.2. COREFERENCE ANNOTATION

B.2.1.2 Pronominal type

This type of coreference is characterised by a pronoun referring to a noun phrase. We enumerate below the subtypes of pronominal coreference (with corresponding examples) for the annotator’s reference. However, every pronominal mention should be assigned the type proun regardless of its subtype.

Demonstrative pronouns. These include the pronouns this, that, these and those.

The structures of plakorstatins 1 and 2 including relative configuration were elucidated on the basis of mass and 2D NMR spectroscopic interpretations. These are the first plakortides with an epoxy group in the side chain.

Personal pronouns. These include the pronouns it and they. Also falling under this category are possessive pronouns such as its, their and theirs. Although its and their are often used as modifiers of a head noun, we consider them as pronominal since the coreferring expression corresponds to the pronoun itself, not the whole noun phrase.

Dehydroradiosumin, a novel potent trypsin inhibitory dipeptide, was isolated from the freshwater cyanobacterium Anabaena cylindrica (NIES-19). Its structure was elucidated as 1 on the basis of 2D NMR data and chemical degradation.

Indefinite pronouns. Included in this type are the following pronouns: another, one, other, few, many, others, several, all, any, more, most and some.

Four different types of marine natural compounds isolated from tunicates were found to inhibit human aldose reductase. All are characterized by a heterocyclic system, and at least two phenolic groups are present in the structure.
Distributive pronouns. These include the pronouns *both, such, each, either* and *neither*.

The structures of these compounds were determined on the basis of their spectroscopic data analysis (1H, 13C, 1H-1H COSY, HMQC, and HMBC NMR, as well as low- and high-resolution mass experiments). Each was tested for its DPPH (1,1-diphenyl-2-picrylhydrazyl) radical-scavenging property.

Relative pronouns. These include the pronouns *which, that* and *whose*.

Its geometrical isomer, *apakaochtodene B* (2), which could not be separated from 1 and thus characterized as a 95:5 mixture of 2:1 had (1)H and (13)C NMR spectral characteristics similar to previously known ochtodene (3) and the related tetrahalogenated monoterpene 4.

**B.2.1.3 Numerical Type**

This type applies to a number which refers to a compound previously introduced using the same number.

A novel anthracycline, *komodoquinone A* (1), and its aglycone, *komodoquinone B* (2) were isolated from the solid-state fermentation of the marine Streptomyces sp. KS3, which was isolated from marine sediment. The absolute stereostructures of 1 and 2, except for the sugar portion, were elucidated on the basis of chemical and physicochemical evidence.
B.2. COREFERENCE ANNOTATION

B.2.1.4 Abbreviation

An abbreviation should be marked as a referring expression with the full name as its referent.

Acetylacetone is a precursor to **acetylacetonate (ACAC)**. **ACAC** is a common bidentate ligand.

Note that in the example above, both instances of ACAC should be annotated as referring expressions.

B.2.2 Guidelines and Examples

Coreference annotation should be done after named entity annotation of a document. Each mention which has not been annotated as a named entity needs to be marked up as a `ref` element.

B.2.2.1 Attributes

Values for the following attributes then need to be supplied:

- **min** (minimum string) - required only by referents

- **idref1** - required by each referring expression as this realises the link to its referent, specified using its unique identifier. There is a provision for adding values for `idref2` to `idref11`, in cases where there are multiple referents.

- **type** - the type of relation; needed only by referring expressions.

Please refer to the instructions for using XConc for detailed steps on how this is done.
B.2.2.2 Choosing the referent

Between two noun phrases which corefer, the more specific one should always be considered as the referent. (In the case of abbreviations, the full meaning is considered as the referent.)

A cytotoxic cembranoid, claviolide (7), was isolated from the methylene chloride solubles of the Formosan soft coral Clavularia violacea.

In the example above, the referent is *claviolide*.

In cases where the annotator has more than one expression to choose from to mark up as referent, the closest expression should be chosen. This could be an expression that has been marked up as a referring expression previously.

Aeroplysinin-1 (1) and the structurally related dienone 2 were cytotoxic to Ehrlich ascites tumor (EAT) cells and HeLa tumor cells in the microculture tetrazolium (MTT) and clonogenic assays. Both compounds are bromotyrosine derivatives, isolated from the marine sponge Aplysina aerophoba.

For the sake of illustration, let us assume that *Both compounds* has been previously annotated as the referring expression with *aeroplysinin-1* and *the structurally related dienone 2* as its referents. In the current annotation, *bromotyrosine derivatives* is the referring expression with *Both compounds* its referent.

B.2.2.3 The minimum string

The minimum string is the head of a phrase or the main noun (i.e., with the left and right modifiers excluded).
Aeroplysinin-1 (1) and the structurally related dienone 2 were cytotoxic to Ehrlich ascites tumor (EAT) cells and HeLa tumor cells in the microculture tetrazolium (MTT) and clonogenic assays. Both compounds are bromotyrosine derivatives, isolated from the marine sponge Aplysina aerophoba.

In the example above, aeroplysinin-1 and the structurally related dienone 2 are the referents of Both compounds with indefinite noun phrase as the type of coreference. The minimum string \((\text{min})\) needs to be specified only for the referent. The minimum string of aeroplysinin-1 is aeroplysinin-1; that of the structurally related dienone 2 is dienone 2.

In cases when the minimum string is exactly the same as the referent or only differs by the presence of articles such as the, an and a, the annotator may opt to leave the \(\text{min}\) property blank.

Although there might be cases when they would have the same values, the minimum string is different from the lexeme, which is a property of a named entity. The lexeme, specified during named entity annotation, is the mention without the inflections (i.e., modifications due to number, gender, tense, etc.), whereas the minimum string is the main noun of the referent.

Four new marine monoterpenes have been isolated from two species of marine red algae, Plocamium cartilagineum and Pantoneura plocamioides. The structures and relative stereochemistry of these compounds were determined on the basis of spectroscopic evidence and suggest a relationship between P. cartilagineum and P. plocamioides.

In the example above, the lexeme of the term monoterpenes is monoterpane while the minimum string is monoterpenes.
In case of a “headless” noun phrase, the minimum string will be the last token of the noun phrase before any prepositional phrases, relative clauses and other modifiers appearing to the right.

Ten brominated alkaloids were isolated from the dichloromethane extract of the North Sea bryozoan Flustra foliacea. Of the 10, four (1, 2, 3, and 5) represent new natural products.

In the given example, there are two referring expressions: the first is 10 whose referent is brominated alkaloids (or only alkaloids, depending on the annotator’s discretion) with type definite; the second is four whose referent is 10 (since it is closer), also with type definite. For the “headless” referent 10, the min string can be assigned the value 10.

B.2.2.4 Priority types

In some cases, more than one type of coreference applies. This is an especially usual case with appositives. Amongst the sortal subtypes, the appositive and predicate nominative types take precedence over the others.

Secobatzelline A (1), a new batzelline natural analogue, and secobatzelline B (2), a likely artifact formed during the isolation procedure, have been isolated from a deep-water marine sponge of the genus Batzella.

In the example above, the referring expression a new batzelline natural analogue is an indefinite noun phrase which is also an appositive with Secobatzelline A as the referent. In such cases, the appositive type is the lower priority (i.e., indefinite should be chosen as the type).
There are also cases when the numerical coreference type coincides with other types of coreference.

Bioassay-guided fractionation of a marine extract from Trididemnum cyclops afforded the new lipopeptide 39-oxobistramide K (1) and the known bistramides A (2) and D (3). The isolate 1 was tested for antiproliferative activity against the A2780 cell line and exhibited an IC(50) value of 0.34 microM.

In the given example, the phrase The isolate 1 should be taken as one expression only; it should not be broken down into The isolate and 1 as the phrase The isolate is ambiguous without the numerical reference 1. However, there are two applicable types to the phrase The isolate 1: definite and numerical. In such cases, the numerical type is of the higher priority, i.e., num should be chosen as the type.

B.3 Document Type Definition (DTD) file

<!ENTITY % term.class "(
chem|
pertains_to_chem|
organism|
pertains_to_org|
anatomical_part|
cell|cell_line|
cell_part|
gene_protein|
nucleic_acid|
enzyme|
biological_activity|
bioprocess|
pertains_to_bioactivity|
disease|
pertains_to_disease
"

<!ENTITY % coref.class "(
definite|
indefinite|
demonstrative|
distributive|
pron|
appos|
pred_nom|
relat|
num|abbrev|
name|
other
)">

<!-- ============================ -->
<!-- HIERARCHICAL ELEMENTS -->
<!-- ============================ -->

<!ELEMENT Annotation (PubmedArticleSet)>
<!ATTLIST Annotation
  annotates CDATA #IMPLIED>
creator CDATA #REQUIRED
created CDATA #IMPLIED

<!ELEMENT PubmedArticleSet (PubmedArticle)>  
<!ELEMENT PubmedArticle (MedlineCitation)>  
<!ELEMENT MedlineCitation (PMID, Article)>  
<!ELEMENT Article (ArticleTitle, Abstract)>  
<!ELEMENT Abstract (AbstractText)>  

<!-- ============================ -->  
<!-- BLOCK ELEMENTS -->  
<!-- ============================ -->  
<!ELEMENT ArticleTitle (sentence | event)+>  
<!ELEMENT AbstractText (sentence | event)+>  
<!ELEMENT PMID (#PCDATA)>  

<!-- ============================ -->  
<!-- LINGUISTIC ELEMENTS -->  
<!-- ============================ -->  
<!ELEMENT sentence (#PCDATA|cons|term|encterm|frag|ref)*>  
<!ATTLIST sentence id ID #REQUIRED>  

<!ELEMENT cons (#PCDATA|cons|term|encterm|frag|ref)*>  
<!ATTLIST cons
  sem CDATA #REQUIRED
  lex CDATA #REQUIRED>
id ID #REQUIRED
min CDATA #IMPLIED
type %coref.class; #IMPLIED
idref1 IDREF #IMPLIED
idref2 IDREF #IMPLIED
idref3 IDREF #IMPLIED
idref4 IDREF #IMPLIED
idref5 IDREF #IMPLIED
idref6 IDREF #IMPLIED
idref7 IDREF #IMPLIED
idref8 IDREF #IMPLIED
idref9 IDREF #IMPLIED
idref10 IDREF #IMPLIED
idref11 IDREF #IMPLIED
>

<!ELEMENT term (#PCDATA|term|frag|ref)>  
<!ATTLIST term
sem %term.class; #IMPLIED
lex CDATA #IMPLIED
id ID #REQUIRED
min CDATA #IMPLIED
type %coref.class; #IMPLIED
idref1 IDREF #IMPLIED
idref2 IDREF #IMPLIED
idref3 IDREF #IMPLIED
idref4 IDREF #IMPLIED
idref5 IDREF #IMPLIED
idref6 IDREF #IMPLIED
idref7 IDREF #IMPLIED
idref8 IDREF #IMPLIED
idref9 IDREF #IMPLIED
idref10 IDREF #IMPLIED
idref11 IDREF #IMPLIED
>

<!ELEMENT encterm (#PCDATA|cons|term|encterm|frag|ref)*>  
<!ATTLIST encterm
  sem  %term.class; #IMPLIED
  lex  CDATA #IMPLIED
  id   ID #REQUIRED
  min  CDATA #IMPLIED
  type  %coref.class; #IMPLIED
  idref1 IDREF #IMPLIED
  idref2 IDREF #IMPLIED
  idref3 IDREF #IMPLIED
  idref4 IDREF #IMPLIED
  idref5 IDREF #IMPLIED
  idref6 IDREF #IMPLIED
  idref7 IDREF #IMPLIED
  idref8 IDREF #IMPLIED
  idref9 IDREF #IMPLIED
  idref10 IDREF #IMPLIED
  idref11 IDREF #IMPLIED
<!ELEMENT ref (#PCDATA|term|frag|cons|encterm|ref)>*>
<!ATTLIST ref
id   ID    #REQUIRED
min  CDATA  #IMPLIED
type %coref.class; #IMPLIED
idref1 IDREF #IMPLIED
idref2 IDREF #IMPLIED
idref3 IDREF #IMPLIED
idref4 IDREF #IMPLIED
idref5 IDREF #IMPLIED
idref6 IDREF #IMPLIED
idref7 IDREF #IMPLIED
idref8 IDREF #IMPLIED
idref9 IDREF #IMPLIED
idref10 IDREF #IMPLIED
idref11 IDREF #IMPLIED
>
B.4 Instructions for configuring and using XConc Suite

B.4.1 Installation of XConc Suite

1. Download the Eclipse installer. Extract files from the installer into the preferred directory. Run the Eclipse executable. Select the preferred workspace.

2. To install the XConc Suite, select Help → Software Updates → Available Software → Add Site...

3. Enter the URL [http://www.nactem.ac.uk/tsujii/xconc](http://www.nactem.ac.uk/tsujii/xconc).

4. Tick the check box beside the name and click Install... Proceed with the rest of the installation steps (e.g., agreeing to the license).

5. Restart Eclipse for the changes to take effect. Change the perspective by choosing Window → Open Perspective → Other → Vex-XConc.
B.4.2 Accessing the corpus

1. Download the corpus and extract the files into the preferred directory.

2. In Eclipse, create a new Vex Plug-in Project by selecting File → New → Project... → Vex → Vex Plug-in Project and clicking on Next. Enter a project name (e.g., HANAPIN_corpus).

3. Import files by right clicking on the project name and then selecting Import... → General → File System. Browse to the extracted corpus directory from step (1). Tick the check box beside the project name.

4. Register the DTD file. Under the types directory, right click on HANAPIN_1.0.dtd and then select Properties → Vex Document Type.

5. Enter the following values:

   (a) Name: HanapinCoreference

   (b) Public ID: -//HANAPIN//DTD HANAPIN ANNOTATION 1.0//EN

   (c) System ID: ../types/HANAPIN_1.0.dtd

6. Select Annotation as the Root Element then click Apply.

Figure B.1: Setting up the DTD File
7. Register the CSS file. Under the types directory, right clicking on HANAPIN_1.0.css and then select Properties → Vex Style.

8. Enter HanapinCoreference as the value of Name, and check HanapinCoreference under Document Types. Click Apply.

![Figure B.2: Setting up the CSS File](image)

9. To set XConc as the default XML editor, select Window → Preferences → General → Editors → File Associations. Under File Types, click on .xml. Under Associated Editors, click on Vex XML Editor and click on the Default button to set it as the default editor.

![Figure B.3: Setting Up the Default XML editor](image)
10. You can now double-click on any of the XML files under the corpus directory to view the abstracts. Note that each abstract contains the following information: PubMed ID (PMID), title and one or more sentences. The title is represented as the first sentence, and is also included in the annotation.

![Structure of an Abstract](image)

**Figure B.4: Structure of an Abstract**

### B.4.3 Named entity annotation

#### B.4.3.1 Steps:

1. Highlight the text to be annotated. Right click on the highlighted text and select *Insert Element*.

2. In the pop-up window which will appear, select the desired type of element amongst:

   (a) **term** - a named entity in a continuous text region

   (b) **encterm** - for marking up an enclosing named entity, i.e., those which contain embedded entities

   (c) **cons** - for coordinated named entities
B.4. INSTRUCTIONS FOR CONFIGURING AND USING XCONC SUITE

289

Figure B.5: Selecting the Type of Element for Insertion

(d) `frag` - for identifying the common fragment in coordinated named entities

3. After a `term` element has been marked, right click on it and select `Show Property View`. In the pop-up window which appears, the values for the following attributes need to be filled in:

(a) `id` (automatically suggested)

(b) `lex` (lexeme) - the form of the word without any inflections (e.g., modifications due to tense and number); for our use, spaces in between words should be replaced with underscores, and these should begin with lowercase letters (with the exception of proper names)

(c) `sem` (semantic class) - the category a named entity falls under; one of the classes described in the guidelines

B.4.3.2 Examples:

- annotating a named entity (in a continuous text region)

1. Select the noun phrase and mark it as a `term` element by right clicking on
it and selecting Insert Element → term.

2. Set the attributes by right clicking on the selected term and selecting Show Property View. Enter the values for lex and sem.

• annotating embedded named entities

1. Select the enclosing named entity and mark it as an encterm element by right clicking on it and selecting Insert Element → encterm.

2. As with a named entity in a continuous text region, set the attributes by right clicking on the highlighted named entity and selecting Show Property View. Enter the values for lex and sem.

3. Annotate any named entities embedded within the enclosing one using a term element.

• annotating terms in a discontinuous text region

1. Divide the discontinuous text region into individual entities (e.g., for the phrase Palythoalones A and B, there will be two entities - Palythoalones A and B. This can be done by following the same steps as with ordinary named entities.
2. Select the text region containing the discontinuous named entities. Mark it as a `cons` element by right clicking on it, selecting `Insert Element` → `cons`.

3. Set the attributes by right clicking on the selected term and selecting `Show Property View`. Enter the values for `id` (in the form $C_n$ by convention), `lex` (of the form `CONNECTOR lex1 lex2`) and `sem` (of the form `CONNECTOR sem1 sem2`) where:

   - `CONNECTOR` is the coordinating connector (e.g., AND, OR)
   - `lex1` and `lex2` are the lexemes of named entities 1 and 2, respectively
   - `sem1` and `sem2` are the semantic classes of named entities 1 and 2, respectively
4. Finally, place the fragment common to the coordinated entities in a `frag` element.

**B.4.4 Coreference Annotation**

1. To annotate the referent:

   (a) In the case that the referent has been previously annotated during named entity annotation, right click on the named entity and select *Show Property View*.

   (b) Enter a value for the `min` attribute.

2. To annotate the referring expression:

   (a) If the expression has already been annotated during named entity annotation, proceed to entering values for the following attributes by selecting *Show Property View*:

      - `idref1` - the id of the referent this expression is referring to
      - `idref2` to `idref11` - to be filled in as necessary, i.e., if there are multiple referents
      - `type` - the type of relation

   (b) If the expression has not been annotated, highlight the expression and mark it as a `ref` element. Enter values for the following attributes:

      - `id` (automatically suggested)
      - `idref1` - the id of the referent this expression is referring to
      - `idref2` to `idref11` - to be filled in as necessary
      - `type` - the type of relation
Figure B.9: Annotating the referring expressions

Figure B.10: Annotating the referring expressions - fragment
Appendix C

Coreference resolution resources

C.1 List of pronouns

I, all, another, any, anybody, anyone, anything, both, each, either, everybody, everyone, everything, few, he, her, hers, herself, him, himself, his, it, its, itself, little, many, me, mine, more, most, much, myself, neither, no, nobody, none, nothing, one, other, others, our, ours, ourselves, several, she, some, somebody, someone, something, that, their, theirs, them, themselves, these, they, this, those, us, we, what, whatever, which, whichever, who, whoever, whom, whomever, whose, you, your, yours, yourself, yourselves
C.2 List of chemical compound hyponyms from WordNet

U308, acceptor, acid, adduct, alkali, allene, allomorph, aluminate, ammine, analogue, anhydride, anionic compound, antiknock, arsenide, azide, base, benzofuran, benzoquinone, binary compound, bitter principle, buffer, calcium-cyanamide, carbon disulfide, carbonyl, caustic, cementite, chloride, chloropicrin, chromogen, cofactor, complex, compound, coordination compound, corrosive, coumarone, cumarone, cumulene, cyanamide, defoliant, depilatory, derivative, dimer, enamel, enantiomer, enantiomorph, ester, ether, exotherm, fixer, fixing agent, flavone, formulation, goitrogen, heterocycle, heterocyclic, heterocyclic compound, hydrate, hydrazone, hydrogen cyanide, hydroxide, hydroxide hydrated oxide, imide, imine, incense, inorganic compound, iodocompound, iron carbide, isomer, isoprenoid, lipid, manganese tetroxide, menthol, monomer, natural product, nitrate, nitride, nitrochloroform, nitrogen mustard, organic compound, oxide, oxime, ozonide, polymer, pregnanediol, preparation, preservative, quinone, repellant, repellent, repellent repellant, salt, silicide, siloxane, solvate, sternutator, sternutatory, stripper, sulfide, sulphide, synthetic, synthetic substance, taurine, telluride, tenderiser, tenderizer, tetrachloride, thiol, triazine, vanillin, yellowcake
Appendix D

HANAPIN DTI Corpus annotation guidelines

D.1 Pre-annotated Entities.

We are concerned with two types of entities: marine drugs (MDG) and enzymes (ENZ).

These entities have already been automatically marked-up using a tagger.

The HIV replication cycle offers multiple targets for chemotherapeutic intervention, including the viral exterior envelope glycoprotein gp120, viral co-receptors CXCR4 and CCR5, transmembrane glycoprotein gp41, integrase, reverse transcriptase and protease. Most currently approved anti-HIV drugs belong to nucleoside/nucleotide reverse transcriptase inhibitors, non-nucleoside reverse transcriptase inhibitors or protease inhibitors. Highly active antiretroviral therapy, which combines several drugs, has dramatically improved patient lives. However, adverse side effects, long-term toxicity and drug resistance limit their therapeutic effect. In addition to the above-mentioned drugs, compounds that target viral entry and virus-cell fusion have great potential for the treatment of HIV infections. The sponge-derived cyclodepsipeptides papuanide and mirabamides have been shown to inhibit HIV entry into cells [23,24]. HIV entry into host cells is a multi-step process that remains to be fully elucidated. The proteins involved in the entry process have become attractive targets for drug design. Natural probes such as depsipeptides, which target viral entry, could provide valuable information for computer-aided drug design. In this context, recent achievements in the treatment of HIV infection and the application of computational methods for current drug design was recently reviewed [25].

Figure D.1: Pre-annotated marine drugs (in turquoise) and enzymes (in salmon).

As a result, some mistakes might be encountered. Some entities might have been missed while some might have been annotated when they should not be. Also, marine drugs might have been annotated as enzymes and vice-versa. Such cases need to be
D.2 Recognising Interactions.

For this task, we define an interaction as a relation between a marine drug and an enzyme. In such relations, an entity acts as an agent while another characterises the target of the interaction. The interaction is usually implied by words which we will call interaction clues.

We say that an interaction exists if three elements are present: an agent, a target and an interaction clue. If any of these three is missing in a span of text, we do not consider that span as describing an interaction. The span of text in the example below is not considered an interaction since it does not have an agent.

![Figure D.2: A span of text that does not describe an interaction because of a missing element, the agent.](image)

D.2.1 Pre-marked Interaction Phrases.

The phrases which contain both marine drugs and enzymes (PHRASE) have already been highlighted in yellow. This is to make it more convenient for the annotators to find interactions between the two types of entities.

However, some of these marked phrases might need correction. Below are a few cases of incorrectly marked phrases:

1. The marked phrase has both marine drugs and enzymes but does not describe an interaction. Only phrases which have interaction clues should be left annotated. Please remove the annotation of marked phrases which do not have interaction clues. In the sentence below, for example, the PHRASE label should be removed.
Figure D.3: Pre-marked phrases in yellow.

Figure D.4: The **Phrase** label for phrases which do not have interaction clues should be removed.

2. The highlighted text contains unnecessary words or phrases. This usually is the case when the entire sentence has been marked when in fact the interaction is described in only a phrase of the sentence. The mark-up on the entire sentence should then be removed so that the **minimal phrase** can be marked-up.

Figure D.5: An entire sentence annotated by our automatic phrase extractor. The annotated text needs to be trimmed down.

The **minimal phrase** is the shortest span of text that contains the agent, the
D.2. RECOGNISING INTERACTIONS.

target, and the interaction clue. However, if there are contextual words that describe conditions which are relevant to the interaction (e.g., concentration, temperature, medium), they should be included.

In the paragraph in the above figure, the whole sentence was selected by our automated phrase extractor.

The shortest span of text containing the agent (poly-APS), the target (acetylcholinesterase (AChE)), and the interaction clue (inhibitors) is “poly-APS act as antifouling compounds [7], are strong inhibitors of acetylcholinesterase (AChE)”.

However, one might consider the concentration at which the interaction occurs as relevant contextual information. Hence, the minimal phrase should be annotated as shown below.

![Figure D.6: An example of a minimal phrase that includes context.](image)

References should not be included in the phrases. If they were annotated as part of the phrase, please redo the annotation without including the references. In the example below, the references [25,26] were originally included in the phrase but were then removed.
APPENDIX D. HANAPIN DTI CORPUS ANNOTATION GUIDELINES

Figure D.7: References should be removed from the annotated phrase.

3. Missed phrases. The annotators might find phrases describing interactions between marine drugs and enzymes which have not been highlighted. These phrases need to be marked-up. This implies that it is necessary to read the entire document to ensure that there will be no missed phrases.

D.2.2 Annotation of Agents.

When the interaction is caused or triggered by an entity, we say that the entity plays the role of an agent. In this task, agents are usually marine drugs, although there might also be cases when an enzyme acts as an agent.

If an entity acts as an agent, it should be annotated as `AGENT`. All agents, regardless of entity type, will be shown in brown.

Please mark-up as `AGENTs` only the entities which are involved in an interaction. Isolated instances are not `AGENTs`. Note that in the following example, only the instance of `poly-APS` that is involved in an inhibition relation was assigned the `AGENT` label.
D.2. RECOGNISING INTERACTIONS.

Figure D.9: Only instances that are causing an interaction should be annotated as AGENTS.

D.2.3 Annotation of Targets.

When an entity is at the receiving end of an interaction, we say that the entity characterises the target of the interaction. Unlike agents which will always be just entities, targets can be entities or bioprocesses involving the entities. Targets should be assigned the label TARGET.

Below is an example where the target is an entity, specifically an enzyme. It is shown in blue.

In conclusion, the observed in vivo toxic effects of poly-APS (and probably of other polymerised 3-alkylpyridinium salts like halotoxins [10,11]) reflect their polycationic nature and behaviour in aqueous solution. While at lower concentrations poly-APS can act as ACE inhibitors, at higher doses other mechanisms producing haematological and vascular effects dominate and mask these effects. Besides the membrane-permeabilizing activity on erythrocytes and other tissue cells, which is probably also responsible for arrhythmia and thrombocyte aggregation, the non-specific binding of poly-APS to different serum proteins and membrane-bound proteins on platelets induces the formation of large plugs that could block the small blood vessels and stop the lung and heart blood flow, blue-bordered area indicating a target.

In contrast, in the following example, the target is a bioprocess that involves an enzyme, specifically, the phosphorylation of said enzyme. Note that, additionally, in such cases, the word(s) signifying the bioprocess (e.g., phosphorylation) should be
announced as BIOPROCESS, shown in grey in the figure.

As in the annotation of AGENTS, only entities involved in an interaction should be annotated as TARGETS. Isolated instances are not TARGETS. In the following example, only the instance of AChE that is at the receiving end of an interaction is annotated.

D.2.4 Annotation of Interaction Clues.

The interaction is usually implied by clues which are words describing the interaction. They are usually verbs (e.g., suppress), nominalised verbs (e.g., suppression)
and nouns (e.g., inhibitor). Such clues need to be annotated as **INTERACTION**. The interaction clue in the following example is shown in green.

**Figure D.13:** Interaction clue in green.

### D.2.5 Negative, Neutral and Uncertain Interactions.

Also within the scope of the task are interactions which are negative (e.g., *drug D does not inhibit enzyme E*), neutral (e.g., *effect of drug D on enzyme E*) and uncertain (e.g., *It was not established whether drug D activates enzyme E*). In such cases, please make sure that the phrase includes the words which make these interactions negative/neutral/uncertain. The interaction clue is annotated in the same manner as in positive interactions. Below is an example of a neutral interaction.

**Figure D.14:** A neutral interaction.

In the next example, we show that the annotation of the uncertain interaction in the second sentence was done in the same manner as the positive interaction in the first sentence. Please ensure that the word that makes the interaction uncertain (i.e., *whether*)
has been included in the highlighted phrase.

![Figure D.15: An uncertain interaction.](image1)

Same is the case for the negative interaction in the third sentence below.

![Figure D.16: A negative interaction.](image2)

### D.2.6 Inter-sentential Interactions.

Interactions which are described over several sentences also need to be annotated. As the automated phrase marker does not work for multiple sentences, sentences with inter-sentential interactions have not been marked-up yet. Below is an example where the last sentence uses an anaphor (i.e., *it*) in describing the interaction.

![Figure D.17: An example of inter-sentential interaction.](image3)

In such cases, please mark-up the shortest possible text span that would include the
D.3 OTHER ITEMS

305

agent, target and interaction. The agent, target and interaction clue will then be annotated in the usual manner.

Figure D.18: Annotation of inter-sentential interactions.

D.3 Other Items

D.3.1 Spelling or Grammatical Errors

Ignore any spelling mistakes or grammatical errors within the documents, such as in the example below where cellulase was misspelt as cellulose. However, as these are actual published papers, we have to accept them as they are. Editing the document text has been disabled as correcting the authors’ mistakes is beyond the scope of our task.

Figure D.19: Mistakes committed by the authors should be ignored.

D.3.2 Points for Discussion

If during your annotation you come across something that you would like to remark on, please use the COMMENT label to mark-up the pertinent part of the document in grey.
You will then be asked to provide your comment in a small pop-up window.

![Figure D.20: The comment box.](image)

If your remarks need discussion, send an email to all of us so that we can arrive at an agreement regarding your concern. The email addresses of the people involved are as follows.

- **Amir** - syedamiriqbal@googlemail.com
- **Bala** - balakkvj@gmail.com
- **Michelle** - michelleterezm@gmail.com
- **Rafal** - rafal.rak@manchester.ac.uk
- **Riza** - rbbatista@gmail.com
Appendix E

Event extraction resources

E.1 Additional trigger dictionary entries

account, accounts, accounted, accounting, activity, affinity, bioconvert, bioconverts, bioconverted, bioconverting, bioconversion, cascade, cascades, cascaded, cascading, catalysis, contribute, contributes, contributing, contributed, contribution, correlate, correlates, correlated, correlating, correlation, deacetylate, deacetylates, deacetylated, deacetylation, effective, efficiency, elicit, elicits, elicited, eliciting, epimerize, epimerizes, epimerized, epimerizing, epimerization, epimerise, epimerises, epimerised, epimerising, epimerisation, epoxidate, epoxidates, epoxidated, epoxidating, epoxidation, hydrolysis, hydrolytic, inactive, inactivity, lead, leads, led, leading, liberate, liberates, liberating, liberated, liberation, neuromodulate, neuromodulates, neuromodulated, neuromodulating, neuromodulator, neuromodulatory, neuromodulation, oxygenate, oxygenates, oxygenated, oxygenating, oxygenation, pathway, pH, potency, potent, preference, property, receptor, resorb, resorbs, resorbed, resorbing, resorption, resorbent, resorptive, responsible, selective, selectivity, sensitively, sensitivity, specific, specificity, substrate, sulfate, sulfates, sulfated, sulfation, supplement, supplements,
supplemented, supplementing, supplementation, supplementary, supplementarity, trigger, triggers, triggered, triggering

E.2 Pre-compiled list of prepositions

aboard, about, above, absent, across, after, against, along, alongside, amid, amidst, among, anti, around, as, at, atop, before, behind, below, beneath, beside, besides, between, beyond, but, by, concerning, considering, despite, down, during, except, excepting, excluding, following, for, from, in, in front of, inside, instead of, into, like, mid, minus, near, next, of, off, on, on top of, onto, opposite, out of, outside, over, past, per, plus, regarding, round, save, since, than, through, till, times, to, toward, towards, under, underneath, unlike, until, up, upon, versus, via, with, within, without