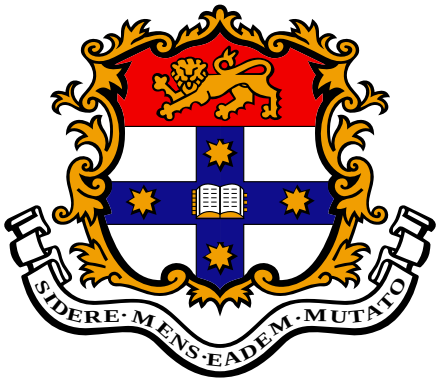
Reducing Semantic Drift in Biomedical Lexicon Bootstrapping



Tara McIntosh

A thesis submitted in fulfillment

of the requirements for the degree of

Doctor of Philosophy

(Computer Science)

School of Information Technologies

The University of Sydney

August 2009

Abstract

Lexical-semantic resources are crucial in many Natural Language Processing (NLP) tasks. These resources are difficult to create and maintain, and their bias, inconsistency and lack of coverage limits their usefulness. This issue is apparent in specialist domains with limited resources, such as biomedicine, where NLP is now being applied. Developing automated methods for extracting such resources is critical to overcoming this knowledge bottleneck.

Biomedical information extraction will greatly improve when systems can exploit full-text articles and process a wider range of linguistic phenomena. I demonstrate the importance of these challenges by analysing the *Molecular Interaction Map* (MIM) corpus I have created. This corpus maps known interaction facts to passages within full-text articles, which are annotated with the phenomena that must be resolved for fact extraction. Unfortunately, the lexical-semantic resources needed for this are limited to specific sub-domains.

Minimally supervised bootstrapping algorithms are commonly used to extract semantic lexicons from text. These methods iteratively expand lexicons from a small number of seed terms. Existing bootstrappers are prone to *semantic drift*, where the original meaning associated with the semantic classes in the lexicon shifts when ambiguous or erroneous terms and/or patterns are introduced during bootstrapping (Curran et al., 2007). This prevents the accurate extraction of large yet precise resources.

I present three novel approaches to reducing semantic drift in bootstrapping, and demonstrate the effectiveness of these within the biomedical domain. Firstly, I have developed a new multi-category bootstrapping algorithm, *Weighted Mutual Exclusion Bootstrapping* (WMEB), which extends MEB (Curran et al., 2007). WMEB incorporates a cumulative pattern pool, and a term and pattern weighting scheme based on association strength. WMEB significantly outperforms existing multi-category bootstrappers, MEB and BASILISK (Thelen and Riloff, 2002).

The standard approach for evaluating bootstrappers uses only one set of hand-picked seeds, which does not allow for robust performance comparisons. I carried out a thorough analysis using random sets of seeds, which demonstrated previously unreported sensitivity to the seeds of the bootstrappers. This leads to my hypothesis that semantic drift can be reduced using an ensemble of bootstrappers seeded with random seeds. I first apply a supervised bagging approach and show that it effectively corrects drift for bootstrappers that are more sensitive and prone to semantic drift, such as BASILISK. However, supervised bagging unfortunately negates one of the main advantages of bootstrapping by requiring thousands of gold seeds.

This motivated the development of a novel unsupervised bagging approach that requires only one set of seeds: by randomly sampling the seeds needed for bagging from the initial lexicons extracted using the original hand-picked seeds. I demonstrate that unsupervised bagging can significantly outperform standard bootstrapping, and even supervised bagging.

My third approach addresses the cause of semantic drift directly within the bootstrapping process. I hypothesise that semantic drift occurs when a candidate term is more similar to the recently added terms than to the seed and/or high precision terms extracted in the earlier iterations. I propose a novel drift metric that measures a candidate term's degree of drift using distributional similarity. This is incorporated directly into WMEB, where it successfully detects and prevents semantic drift that has begun as recently as the previous iteration. This approach significantly outperforms standard WMEB, supervised and unsupervised bagging of WMEB, and other existing filters based on distributional similarity.

This thesis provides significant benefits for the biomedical domain, where there is an immediate need for more comprehensive semantic resources. I have proposed numerous approaches for reducing drift in semantic-lexical bootstrapping. These techniques are domain independent as they do not utilise any bio-specific properties. Therefore, these methods can be exploited to extract large yet precise semantic lexicons for any domain.

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

— Tara McIntosh

Acknowledgments

I know I am incredibly lucky to say I have thoroughly enjoyed my PhD experience. This is largely due to the many people who have had an influence on my life and research during this time. I have been privileged to work closely with my supervisor James Curran. James has been an outstanding supervisor and friend, providing me with endless encouragement, motivation and guidance.

I was fortunate enough to share the PhD journey with some great friends, and being in a research group together with many of them has been most enjoyable. They have all provided sound advice and useful conversations and distractions over the years. I will remember fondly our trips to ALTAS, and to Prague and Singapore for ACL. It is a shame they couldn't share the fancy hotels with me. So I would like to thank David Vadas, for many lunches; Matthew Honnibal, for extending my linguistic knowledge; Mark Assad, for continuing to be an *unreal* friend; Ghazi Al-Naymat, for keeping me up to date in data mining; and Toby Hawker, for making these years some of my most exciting and memorable.

I would also like to thank David Hawking for being supportive and inspirational and always showing a keen interest. I regret not organising more time to chat with David over good coffee.

Many of my fun travels during my PhD would not have been possible without the support of the CSIRO and David Hawking. My four month visit to Edinburgh University was intellectually inspiring. In Edinburgh, I had the opportunity to work closely with other BioNLP researchers and be involved in interesting reading groups and discussions. In particular, Bonnie Webber, Claire Grover, Mike Matthews and Ben Hachey provided me with invaluable comments and suggestions on my research. I would also like to thank Chris Geib, Vera Demberg, Hazel Price and Simon Wilkinson, for making this visit also a holiday.

I would like to thank one of my best friends, Cassie Thornley, who not only kept me positive during the tough moments, but also agreed to manually evaluate the extracted lexicons. Without her, this thesis would not have the much needed interannotator agreement results.

I am grateful to the examiners of this thesis, Marti Hearst, Ellen Riloff and Bonnie Webber, for their encouragement of my work and useful comments that have improved this thesis.

Most importantly, I would like to thank my parents and family for their patience, support and encouragement during this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Processing (NLP) aims to provide computational tools for storing, manipulating and understanding natural language. Lexical semantic knowledge is critical to processing text for many NLP tasks, including but not limited to Information Extraction (IE), Question Answering, Machine Translation and Automated Summarisation. The meaning of a sentence is derived from the meaning of the words themselves (*lexical semantics*) and the way the words interact with each other (*compositional semantics*). One of the simplest and most commonly used representations of semantic information is a *semantic lexicon*. Semantic lexicons group terms or phrases together that are members of the same semantic category.

Lexical-semantic resources are expensive to create and maintain as they need to reflect the current use of language. As a result, resources are often constrained by the sampled texts and rarely provide adequate semantic coverage of a domain. This is particularly a problem in specialist domains, like biomedicine, where NLP techniques are now being applied. These domains frequently use very different semantic categories and terminology to that of traditional newswire, and in turn, require domain specific semantic resources. Therefore, there is an increased need for automatically extracting semantic lexicons from raw text to help overcome this knowledge bottleneck.

This thesis begins with an investigation of various linguistic phenomena that may impair biomedical IE from abstracts and full texts, and an evaluation of the importance of resolving these for accurate IE. My analysis identifies numerous NLP tasks in biomedical literature which can be improved with semantic resources, such as anaphora resolution, and thus demonstrates the importance of lexical-semantic resources for biomedical NLP.

Unfortunately, the biomedical semantic resources available are often restricted to specific sub-domains, such as specific organisms, and thus do not provide sufficient coverage of this broad domain. In this thesis, I address this issue by investigating minimally supervised bootstrapping approaches to automatically extracting biomedical semantic lexicons from raw text. Existing bootstrapping algorithms are prone to extracting terms which rapidly drift away from the original meaning of the lexicon's seed terms (*semantic drift*). This prevents the accurate extraction of large enough resources to be useful. I present an analysis of existing approaches for automatically extracting semantic lexicons. This lead to new intuitions about the task and motivated the development of my new algorithm *Weighted Mutual Exclusion Bootstrapping* (WMEB) and meta-bootstrapping approaches, which outperform the state-of-the-art in bootstrapping algorithms.

1.1 Lexical-Semantic Resources

Lexical semantic resources are often exploited by Information Retrieval (IR) and NLP systems to improve their performance. The simplest of these are semantic lexicons, which map lexical items (words or phrases) to one or more semantic categories. In this thesis, the term *semantic lexicon*, refers to a list of terms that are all associated with a semantic category. For example the terms *influenza*, *arthritis* and *HIV*, are all members of the *Disease* lexicon.

Semantic lexicons are often incorporated within a semantic hierarchy so that generalisations and information about the relationships between semantic categories can be utilised. For example, the terms *influenza* and *HIV* could be assigned to a more fine-grained category *Virus Diseases*, which is a type of *Disease*. WORDNET (Fellbaum, 1998), developed at Princeton

University, is the most commonly used lexical-semantic resource in the NLP community with a hierarchical structure. It is a large database of English nouns, verbs, adjectives and adverbs organised into sets of synonyms (*synsets*), which are connected by various lexical-semantic relations to form a hierarchy. The noun and verb synsets are related by hyponym links (is-a). Antonym and part-of relationships are also present. WORDNET is a broad-coverage semantic resource in that it contains over 155,000 words.[1] However it is often insufficient to cover the specialised vocabulary in non-traditional domains. There are parallel projects for languages other than English, including EuroWORDNET (Ellman, 1998) which connects synsets that are equivalent in several European languages.

The Medical Subject Headings (MeSH, NLM, 2008), is a medical controlled vocabulary thesaurus that arranges canonical medical concepts into a heavily cross-referenced hierarchy. MeSH is currently an eleven-level hierarchy containing 25,186 concept descriptors, such as *Diseases* and *Reticulocytosis*. MeSH is used by the National Library of Medicine (NLM) to index articles with semantic labels to improve IR over the articles in MEDLINE. The NLM has also developed three Unified Medical Language System (UMLS) knowledge sources to facilitate the understanding of medical and health language (NLM, 2006) — the Metathesaurus of medical concepts, such as SNOMED and MeSH, their various names, and the relationships between them; the Semantic Network, which defines the categories and relationships used to classify the terms in the Metathesaurus; and the Specialist Lexicon consisting of both general English and biomedical terms and their syntactic, morphological, and orthographic information.

Semantic resources have been shown to improve the quality of the results retrieved by IR systems. Semantic information is frequently exploited to automatically reformulate a user's query by expanding it with additional synonymous expressions and semantically similar entities related to the original query terms. This helps increase the recall of the system, and removes the necessity of a user to iterate through all possible terms describing the concept of interest. For example, Smeaton et al. (1995) and Gonzalo et al. (1998) exploited WORDNET, and Yang et al. (2005) utilised the UMLS Metathesaurus, as tools for query expansion.

[1] The total number of unique terms in WORDNET 3.0 is 155,287.

Semantic lexicons also facilitate Named Entity Recognition (NER), where improvements in performance are achieved with the incorporation of lexicons (Stevenson and Gaizauskas, 2000). For example, the state-of-the-art supervised NER systems in the MUC-7 (Mikheev et al., 1998) and the CONLL-2003 (Florian et al., 2003) tasks exploited lists of entities, such as person names, cities, countries and organisations.

Semantic information is also critical for Question Answering (QA) systems, and is utilised by all modules of state-of-the-art systems (Paşca and Harabagiu, 2001) including the top-performing systems in the TREC QA task (Hickl et al., 2007; Moldovan et al., 2007). Hickl et al.'s (2007) QA system used more than 500 lexicons and gazetteers to label entity types not covered by the CiceroLite NER system (LCC, 2009), which identifies 300 different types. The Power Answer 4 system also used semantic lexicons (Moldovan et al., 2007). Other NLP tasks benefit from access to semantic information during training, including information extraction (Grover et al., 2007; Daraselia et al., 2004), machine translation (Garcí-Varea et al., 2001; Toutanova et al., 2008), anaphora resolution (Mitkov, 1997; Ng and Cardie, 2002), and summarisation (Barzilay and Elhadad, 1997).

This review highlights a wide range of NLP and IR tasks for which systems have been enhanced by the availability of lexical-semantic resources. Therefore, systems are likely to improve further with the development of additional semantic resources, especially for domains with few and/or limited resources.

1.2 Extracting Semantic Lexicons

The seven traditional Named Entity (NE) categories introduced in the 1995 Message Understanding Conference (MUC, Grishman and Sundheim, 1996): persons, organizations, locations, temporal expressions (times and dates), percentages, and monetary expressions, are too limited for many NLP and IR tasks. As a result, these categories have been re-defined and expanded on. In particular, Sekine and Nobata (2004) have identified 200 semantic categories in newswire texts. Further, as both NLP and IR techniques are now being applied in specialist domains,

additional categories have been specified, such as antibodies (Hersh et al., 2007), proteins and DNA (Ohta et al., 2002), chemicals (Corbett et al., 2007), and galaxies (Murphy et al., 2006). Unfortunately, many of these NE are poorly represented, if at all, in annotated corpora or lexical-semantic resources, which hinders the performance of systems for processing these.

The preparation and maintenance of semantic lexicons is predominantly done by hand, and requires teams with a significant amount of linguistic and domain expertise. This is an extremely time consuming and tedious task, which makes these resources very expensive to create. Further, it suffers from the bias of the experts' background and the sampled texts, that can result in lexicons that are not entirely representative of the category. For example, in WORDNET (Fellbaum, 1998), there are proportionally fewer breeds of cat than dog included (Curran, 2004), while Sekine and Nobata's (2004) reptile lexicon that was manually created by scanning newswire text contains only 66 of the 8734 known reptiles (Uetz et al., 2009). It is also difficult to adapt these resources to the constant changes in language use, especially new terminology and sense distinctions forming and merging.

These issues become even more apparent in specialist domains, which frequently require additional domain independent resources. For example, in the biomedical domain, the TREC Genomics QA task (Hersh et al., 2007) required systems to identify antibodies and mutations within text, even though they are very poorly represented in annotated corpora. Furthermore, the advent of high-throughput bio-technologies has lead to a rapid increase in new terms in the biomedical literature, and it is difficult for manually constructed resources to be kept up to date. Therefore, there is an increased need for automatically extracting semantic lexicons from raw text to help overcome the knowledge bottleneck in many NLP tasks. NLP systems are already exploiting automatically generated resources for QA (Meij and Katrenko, 2007) and IE (Paşca et al., 2006; Lee and Lee, 2007).

1.2.1 Bootstrapping

Many algorithms have been proposed to automatically extract semantic lexicons for NLP tasks with limited linguistic resources (e.g. Hearst, 1992; Riloff and Jones, 1999; Thelen and Riloff, 2002; Yangarber, 2003a; Curran et al., 2007). The most successful methods follow the *minimally supervised* machine learning approach known as bootstrapping that was initially proposed by Riloff and Shepherd (1997). Bootstrapping algorithms can automatically acquire knowledge from unannotated text, and are considered to be minimally supervised as they are typically initialised with a small number of seed instances of the information to extract. For semantic lexicons, seed terms from the category of interest identify contextual patterns in the text where the seeds are located. These patterns are then used to identify new lexicon terms. The new terms expand the lexicon, which in turn is used as the seed set for the next iteration.

Bootstrapping approaches are attractive because they can be domain and language independent (Agichtein and Gravano, 2000; Yu and Agichtein, 2003), require minimal linguistic pre-processing and can be applied to raw text (Curran et al., 2007), and are efficient enough for tera-scale extraction (Paşca et al., 2006). Unfortunately, the existing algorithms suffer from semantic drift, which prevents the extraction of large yet precise semantic lexicons.

During the bootstrapping process, new terms are iteratively added to the semantic lexicon based on a metric scoring both their membership in the category and their suitability for extracting additional terms. In each iteration, a set of patterns are also selected for the purpose of identifying additional lexicon terms. When a bootstrapper extracts terms with multiple senses and/or patterns which weakly constrain the semantic class, these terms and patterns contribute to the selection of additional patterns and terms. This often leads to the identification of incorrect terms and patterns, which in turn causes the meaning associated with the lexicon's semantic class to shift away from its original meaning as defined by the initial seed set. This phenomena is known as *semantic drift* (Curran et al., 2007), and is the central problem I address.

1.3 Biomedical NLP

In this thesis, I develop new minimally supervised bootstrapping algorithms with a specific focus on reducing semantic drift in the acquired lexicons, while only processing raw text. The effectiveness of these algorithms is demonstrated within the biomedical domain, which has an immediate need for these tools and resources.

Almost all known and postulated information relating to biological processes is recorded in the form of semi-structured text, the literature, and to a limited extent, in data repositories. The recent advances in biomedical experimental techniques has lead to many new discoveries, and in turn a massive quantity of data and literature. The amount of biomedical literature available is growing at an unprecedented rate. In 2008, over 720,000 citations were added to the primary biomedical literature collection, PUBMED. This produces a major bottleneck for the interpretation, management and access of scientific information, making it difficult for scientists to keep up with even their own specialised fields. Further, existing IR techniques, using keyword-based queries, often retrieve an infeasibly large number of documents for subsequent manual inspection.

These issues initially prompted considerable interest in the manual curation of databases to aid researchers in finding specific biomedical knowledge stated within the literature. These include the Kyoto Encyclopedia of Genes and Genomes (KEGG, Kanehisa and Goto, 2000), the Database of Interacting Proteins (DIP, Xenarios et al., 2000), the Biomolecular Interaction Network Database (BIND, Bader et al., 2001), and FlyBase (Ashburner and Drysdale, 1994; Tweedie et al., 2009). Unfortunately, the manually curated databases still only cover a small fraction of the published information, and are incredibly difficult to maintain with new discoveries from the literature.

This significant bottleneck has motivated the development of NLP tools for improving the accessibility of knowledge within articles (Ohta et al., 2006; Meij and Katrenko, 2007). In particular, there is a strong focus on the automatic extraction of interactions between bio-entities, such as genes and proteins (Bunescu et al., 2006; Pyysalo et al., 2007; Hunter et al., 2008;

Björne et al., 2009). Unfortunately, statistical NLP models trained on newswire corpora are very inaccurate when applied to biomedical text, as biomedical texts contain frequent domain-specific terminology and notation. New discoveries also result in new concepts and terms, such that biomedical language is continually expanding.

To aid the development of bio-specific NLP tools, biomedical corpora, such as GENIA (Kim et al., 2003, 2008) and BioInfer (Pyysalo et al., 2007), and numerous biomedical IR and NLP competitions, including BioCreative (Hirschman et al., 2005; Wilbur et al., 2007), TREC Genomics (Hersh et al., 2007) and the BioNLP'09 Shared Task on Event Extraction (Kim et al., 2009), have been created. Using these resources, many systems have been developed, ranging from NER (Ando, 2007) and parsers (Pyysalo, 2008) to relation extraction (Pyysalo et al., 2007; Björne et al., 2009). Several tools have also improved the efficiency of curating biomedical databases. For example, FlyBase curators can more effectively navigate relevant articles using the PaperBrowser tool (Karamanis et al., 2008), and the BIND database curators have successfully used the PreBIND/Textomy IE system for recognising abstracts stating protein interactions (Donaldson et al., 2003).

Unfortunately, the available biomedical corpora and semantic resources do not accurately represent the domain as they are limited in size and biased towards specific topics within biomedicine. For example, the GENIA corpus consists of 1999 abstracts indexed with the MeSH terms *Human*, *Blood Cells*, and *Transcription Factors* (Kim et al., 2003), while the NE *Antibody* is marked only five times in the BioInfer corpus. Therefore, it remains difficult to develop accurate statistical models for identifying and extracting the many biomedical entities of interest and the relationships between them. It is reasonable to expect that further improvements will be achieved by the automated extraction of a vast array of biomedical semantic lexicons from raw text.

1.4 Contributions

Chapter 2 describes the *Molecular Interaction Map* (MIM) corpus, which I developed for the primary purpose of determining the importance of a) processing biomedical full-text articles rather than just abstracts, and b) linguistic phenomena that may hinder information extraction from these. This corpus is the first to map interaction facts to annotated passages within full-text articles. These passages are also annotated for synonym and extra fact dependencies, as well as anaphoric and negated expressions. The process of identifying relevant passages and annotating these phenomena is also described. The MIM corpus complements the other biomedical corpora that are annotated with NE and their relationships at the sentence level. This chapter is based on the work presented in McIntosh and Curran (2007a, 2009b).

Chapter 3 presents a detailed analysis of the MIM corpus and demonstrates the importance of processing full-text articles by identifying the article sections where interactions are most commonly stated. The amount of interaction statements requiring synonym and extra facts, as well as external resources, is also determined. I utilise the corpus as a gold-standard test set for an oracle full-text sentence retrieval system to highlight the importance of resolving negated and coreference expressions, and external resources. The work in this chapter has been published as McIntosh and Curran (2007a,b, 2009b).

The MIM corpus is an informative resource and a significant contribution to the bio-NLP community. The detailed annotations in the corpus demonstrate the importance of various NLP challenges for full-text biomedical IE systems, such as resolving anaphoric and negated expressions and identifying dependencies. These tasks are critical for identifying knowledge within abstracts and full-text, and will benefit from bio-specific semantic resources. This insight motivated the main aim of this thesis — to automatically extract large yet precise biomedical semantic lexicons from raw-text.

Chapter 4 reviews the existing minimally supervised bootstrapping algorithms for extracting semantic lexicons. I describe the influential work of Hearst and Riloff, and numerous single-category and multi-category bootstrapping algorithms. Each of the presented algorithms

has been developed to improve recall and reduce semantic drift. I specifically focus on the multi-category algorithms, BASILISK (Thelen and Riloff, 2002) and MEB (Curran et al., 2007), which significantly outperform the single-category bootstrappers. These approaches reduce semantic drift by extracting multiple semantic lexicons simultaneously and incorporating information about other semantic categories. However, these multi-category bootstrappers are still prone to semantic drift. This survey identifies many strengths and weaknesses of these algorithms, and leads to new insights and motivation for the development of a more precise algorithm, *Weighted Mutual Exclusion Bootstrapping* (WMEB), that I describe in Chapter 6.

Chapter 5 describes the manual evaluation methodology used in this thesis to measure the performance of the bootstrapping algorithms. It introduces the ten biomedical semantic categories of interest, and the raw-text biomedical documents their lexicons are extracted from. The quality of the lexicons and their patterns are evaluated manually as the available gold-standard biomedical corpora are too limited for the purpose of automated evaluation. I also describe the guidelines used by the two evaluators to ensure an accurate and consistent evaluation and discuss the quality of the analysis in terms of inter-evaluator agreement.

Chapter 6 presents a new minimally supervised bootstrapping algorithm, WMEB, for extracting larger more precise semantic lexicons. WMEB is a multi-category bootstrapper that extends the mutual-exclusion framework of MEB, by incorporating new candidate term and pattern weighting functions and a novel cumulative pattern pool. I demonstrate that WMEB is less susceptible to semantic drift than existing approaches, and significantly outperforms both MEB and BASILISK for extracting biomedical semantic lexicons. This work has been published as McIntosh and Curran (2008).

In Chapter 7, I argue that the standard methodology used to evaluate bootstrapping algorithms is unreliable. Bootstrappers are typically compared by only evaluating the lexicons they extract when initialised with one set of seeds. This approach provides little insight into an algorithm's variability or general performance. To address this, I use a new evaluation methodology which compares bootstrappers with random seed sets. This approach identifies previously unreported performance variations of the bootstrappers and their sensitivity to the initial seeds.

This insight lead to the main hypothesis of this chapter, that semantic drift within the extracted lexicons can be corrected using an ensemble of bootstrappers. I present a novel unsupervised bagging approach, which improves the precision of the lexicons significantly. This work has been published as McIntosh and Curran (2009a).

My main hypothesis in Chapter 8 is that semantic drift occurs when a candidate term to be extracted is more similar to a group of recently added terms than to the seed and/or high precision terms that were extracted in the early iterations. In this chapter, I explore the use of distributional similarity measurements for extracting semantic lexicons and for filtering terms that are dissimilar to the seeds during bootstrapping. Many of these approaches improve the lexicons extracted but do not address the underlying cause of semantic drift. For identifying and preventing semantic drift within WMEB, I propose a new metric which measures a candidate term's degree of similarity to the initial terms and the recently extracted terms. My results demonstrate that semantic drift can be significantly reduced further by using this metric, and confirm that my hypothesis is valid. This chapter is based on and extends work published in McIntosh and Curran (2009a).

This thesis presents three techniques for automatically extracting semantic lexicons from raw text, which will greatly benefit many NLP tasks. Each method addresses the central problem of drift in semantic lexicon bootstrapping, which prevents the extraction of large yet precise semantic lexicons. These techniques are shown to significantly outperform the existing approaches for extracting biomedical lexicons. However, as these methods do not incorporate any domain-specific information, and rely only on token-based contextual patterns to identify new terms, they are domain-independent. Therefore, they can be utilised to extract semantic lexicons for any domain with suitable seeds and raw text, and in turn contribute to improving many NLP applications.

Publications

The following is a list of publications included in this thesis.

1. Tara McIntosh and James R. Curran. Challenges for automatically extracting molecular interactions from full-text articles. *BMC Bioinformatics*, vol. 10, no. 311, September, 2009.

2. Tara McIntosh and James R. Curran. Reducing Semantic Drift with Bagging and Distributional Similarity. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 396–404, Suntec, Singapore, August 2009.

3. Tara McIntosh and James R. Curran. Weighted Mutual Exclusion Bootstrapping for Domain Independent Lexicon and Template Acquisition. In *Proceedings of the Australasian Language Technology Workshop*, pages 97–105, Hobart, Australia, December 2008.

4. Tara McIntosh and James R. Curran. Sentence retrieval for extracting biomedical knowledge. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 342–349, Melbourne, Australia, September 2007.

5. Tara McIntosh and James R. Curran. Challenges for extracting biomedical knowledge from full text. In *Proceedings of the Workshop on BioNLP (BioNLP)*, pages 171–178, Prague, Czech Republic, July 2007.

The following publication is related but not included in this thesis.

6. Tara Murphy, Tara McIntosh, and James R. Curran. Named entity recognition for astronomy literature. In *Proceedings of the Australasian Language Technology Workshop*, pages 59–66, Sydney, Australia, November/December 2006.

Chapter 2

Molecular Interaction Map Corpus

The increasing availability of full-text biomedical articles will allow more biomedical knowledge to be extracted automatically with greater reliability. However, most Information Retrieval (IR) and Extraction (IE) tools currently process only abstracts. The lack of corpora has limited the development of tools that are capable of exploiting the knowledge in full-text articles. As a result, there has been little investigation into the advantages of full-text document structure, and the challenges developers will face in processing full-text articles.

This chapter describes the *Molecular Interaction Map* (MIM) corpus. The annotation focus during the creation of other biomedical corpora, such as GENIA (Ohta et al., 2002; Kim et al., 2008) and BioInfer (Pyysalo et al., 2007), has been towards the development of specific IE tools. My annotation efforts, which started in 2006, on the other hand, were exploratory and motivated by the few publicly available biomedical corpora at the time. I was interested in identifying linguistic properties that may hinder IE from abstracts and full texts, which were not annotated within existing corpora, and quantifying the relative importance of resolving these for accurate IE.

The MIM corpus, uniquely maps interaction facts to annotated passages within full-text articles. These molecular interaction facts were originally summarised by Kohn (1999) to describe the relationships outlined in his MIM. The corpus tracks the process of identifying

these interaction facts within the articles to form the MIM summaries and captures any factual dependencies that must be resolved to extract the facts completely. For example, a fact in the results section may require a synonym defined in the introduction. The passages are also annotated with negated and coreference expressions that must be resolved.

This chapter details the annotation guidelines for identifying relevant passages which support the MIM interaction facts and their possible dependencies. Through detailed examples, I discuss the critical role of synonym and extra fact dependencies, followed by the guidelines for annotating negated and coreference expressions. The MIM corpus is an invaluable case study that can guide developers of biomedical IR and IE systems, and can be used as a gold-standard evaluation set for full-text IR tasks. The corpus has also provided us with the motivation for the main contributions of this thesis. The following chapter provides a detailed analysis of the MIM corpus.

2.1 Motivation

The development and evaluation of effective NLP tools for the biomedical domain, requires new annotated corpora, as statistical models of language extracted from traditional newswire corpora are very inaccurate when applied to biomedical text. The most comprehensive annotated biological corpora available consist of sets of MEDLINE abstracts (or single sentences) marked with linguistic information such as part-of-speech, anaphoric expressions, and syntactic structure, as well as biological annotation, marking entities such as proteins, genes and cells, and relationships between these entities (Kim et al., 2003; Tanabe et al., 2005; Pyysalo et al., 2007; Kim et al., 2008). As a result, most biomedical IR and IE systems, such as PubMed and Medie (Ohta et al., 2006), have been applied to abstracts only.

Unfortunately, the information in abstracts is dense but limited. For example, Friedman et al. (2001) showed that only 7 out of 19 mentions of unique molecular interactions within a full-text article occur in the abstract. Full-text articles have the advantage of providing more information and repeating facts in different contexts across various sections, increasing the

likelihood of an imperfect system identifying them. This redundancy can also be used for validating and ranking identified facts (Clarke et al., 2001).

Full text contains explicit document structure, e.g. sections and captions, which can be exploited to improve IE. Regev et al. (2002) developed the first biomedical IR system that specifically focused on limited text sections in full-text articles, such as figure captions. Their performance in the KDD Cup Challenge (Yeh et al., 2002), showed the importance of considering document structure. Following Regev et al. (2002), others have investigated the importance of extracting information from specific sections. Yu et al. (2002) retrieved synonyms of proteins and genes from abstracts and full text, and identified more synonyms with higher precision in full text, with the introduction section defining the majority of synonyms. Both Shah et al. (2003) and Schuemie et al. (2004) showed that the results and methods are the least and most informative, respectively, for identifying gene mentions. In contrast, Sinclair and Webber (2004) found the methods useful in assigning Gene Ontology codes to articles.

These section-specific results highlight the information loss resulting from restricting IR and IE to abstracts and other individual sections, as different sections often provide different information (Schuemie et al., 2004). However, there has been little analysis of when the entire document is required for accurate knowledge extraction. For instance, retrieving a fact from the results section may require a synonym to be resolved that is only mentioned in the introduction. Despite this, and the limited full-text annotated corpora available, IR competitions such as the TREC Genomics track (Hersh et al., 2007), require systems to retrieve and rank passages from biomedical full text that are relevant to question style queries.

One of the main issues of processing full-text which has not been addressed in other biomedical corpora, is understanding how text within one section, or even a single sentence, relies on other text within the same article to form a coherent argument. This investigation, aims to identify how important this phenomena is for automatically extracting molecular interactions. This coherency is modeled in the MIM corpus as not only passages that directly state the interaction fact are annotated. The MIM corpus also includes passages from which the fact can be inferred with the addition of knowledge detailed elsewhere in the document. The passages containing

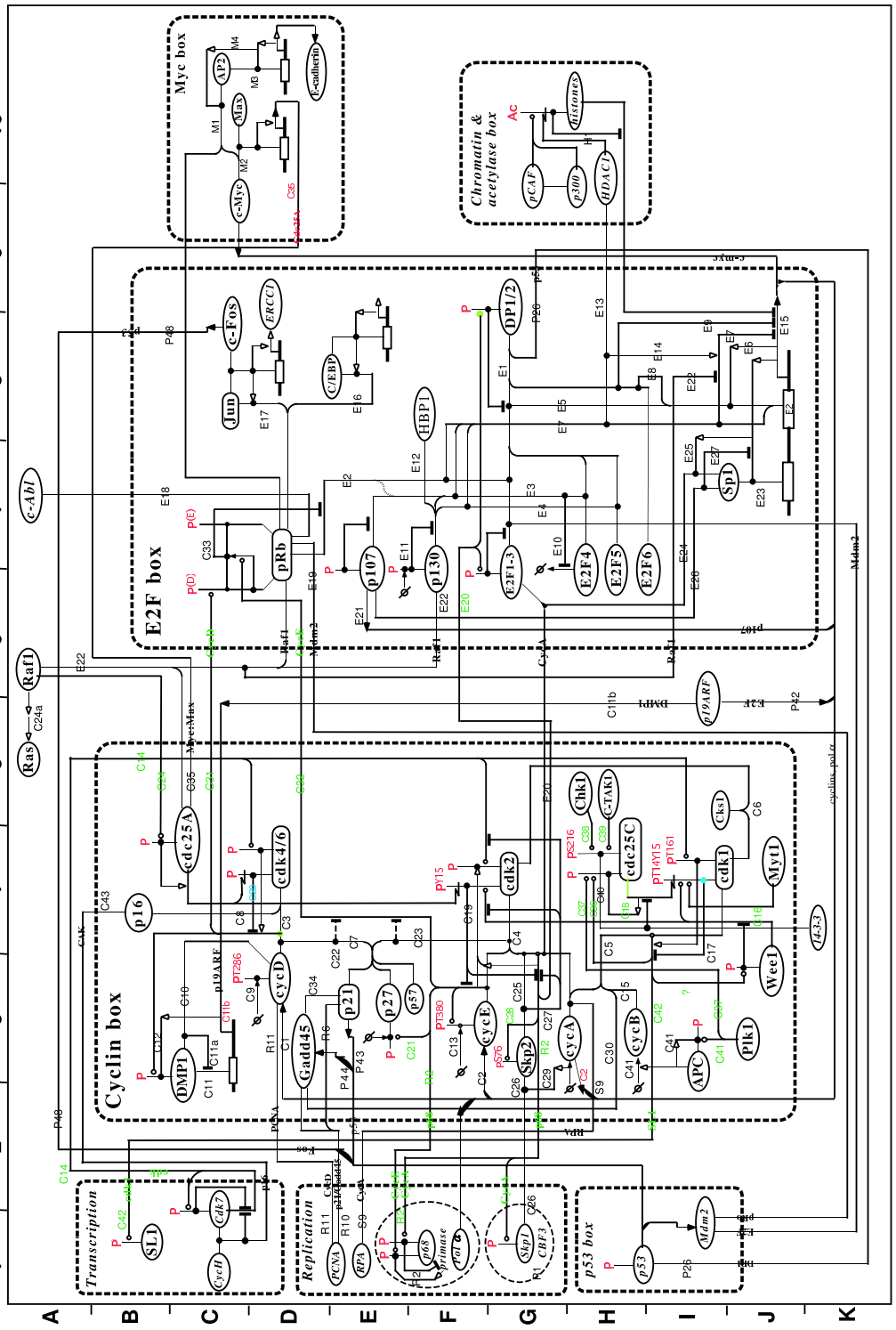


Figure 2.1: Map A of the Molecular Interaction Map compiled by Kohn (1999). See Kohn (1999) for a complete diagram.

this additional knowledge are also annotated, and are referred to as *dependencies*.[1] In the MIM corpus, synonym and extra fact dependencies are annotated. As a result, the corpus uniquely tracks the process of forming summaries of molecular interaction facts from full-text articles.

The MIM corpus also provides insight into the relative significance of other NLP tasks, such as the resolution of negated and coreference expressions. The negated expressions annotated in the corpus are not included in other corpora, and few biomedical corpora include coreference annotations (Castaño et al., 2002; Vlachos et al., 2006; Pyysalo et al., 2007). These corpora only consist of abstracts or individual sentences, and thus do not reflect the level of importance of this task.

The MIM corpus also addresses a number of issues concerning the value of individual full-text sections and other document structures, such as figure headings, and fact redundancy. During the annotation process, the precise location that each fact and its dependencies were identified in has been documented. This brings insight into the applicability of different sections for IE and IR, and whether fact redundancy can be exploited.

2.2 Molecular Interaction Maps

Molecular Interaction Maps (MIM) graphically depict the molecular interactions which occur between molecules of the same or different biochemical families, such as proteins, genes, amino acids, and multimolecular complexes. Kohn (1999) manually constructed a MIM based on scientific literature describing interactions in the mammalian cell nucleus, focusing on cell-cycle regulating molecules and the DNA repair process. Figure 2.1 shows the cell-cycle component of the MIM. This MIM includes 115 individual molecules (excluding complexes) and 203 interactions between them.

Each node in the MIM represents a molecule and the links between nodes correspond to interactions between molecules. Each interaction link is assigned a unique key and is associated with a MIM description composed by Kohn (1999) that summarises evidence for the interac-

[1] These dependencies are not to be confused with syntactic dependencies.

c-Myc and pRb enhance transcription from the E-cadherin promoter in an AP2-dependent manner in epithelial cells (mechanism unknown) (Batschè et al., 1998). *Activation by pRb and c-Myc is not additive, suggesting that they act upon the same site, thereby perhaps blocking the binding of an unidentified inhibitor.* No c-Myc recognition element is required for activation of the E-cadherin promoter by c-Myc. Max blocks transcriptional activation from the E-cadherin promoter by c-Myc, presumably because it blocks the binding between c-Myc and AP2.

Figure 2.2: MIM description for interaction M4 (Kohn, 1999)

tion from the literature, including citations. For example, Figure 2.2 contains the description passage for MIM interaction M4 (on the right of the Myc Box at grid reference C10 in Figure 2.1). It is important to note that the articles cited by Kohn for each MIM description are not exhaustive and thus many of the MIM interactions will be mentioned in other articles. However, the articles selected by Kohn present the primary research documenting the main interaction discoveries/findings. A tool capable of automatically extracting or augmenting a MIM would be extremely useful to biomedical researchers.

2.3 Corpus Annotation

The creation of the MIM corpus involved reverse engineering the formation of Kohn's MIM descriptions by exhaustively tracing and documenting the process of identifying passages from the cited full-text articles that substantiate the MIM interactions. The MIM corpus consists of individual sentences or passages of text (referred to as *instances*) from the cited full-text articles that the MIM descriptions can be inferred from. Each instance in the corpus is separately assigned the location within the full-text article it was retrieved from, and annotated for factual dependencies and coreference and negated expressions. In this thesis, the term MIM will be used to refer to the Molecular Interaction Map and its associated summaries composed by Kohn (1999), while the term MIM corpus will be used to refer to the instances in the corpus.

2.3.1 Annotation Process

The first stage in the development of the MIM corpus involved obtaining the full-text articles cited in the MIM descriptions. There are 262 articles cited by Kohn (1999), and the MIM corpus currently consists of 2005 annotated passages from 78 full-text articles, supporting 76 MIM descriptions. An annotator with a biomedical background exhaustively identified these passages by manually reading each article several times. The corpus is restricted to the cited articles only. This allows us to quantify the need for external resources, e.g. synonym lists and ontologies. The annotation process involved the following:

1. For each MIM description, retrieve the full-text of the cited articles.

2. For each sentence in a MIM description, create a *main fact* to represent the knowledge conveyed.

3. For each main fact, identify and annotate each sentence or passage (*instance*) within the cited articles that the *main fact* can be inferred from. These include direct statements of the fact and passages the fact can be inferred from. An instance of text is said to *support* the fact. I take a minimalist approach, and annotate the shortest sequence of text required to infer the fact, down to an individual sentence.

4. Main facts are often complex sentences, combining numerous facts from the cited articles. Passages from which part of a fact can be derived are also annotated. These instances are assigned to *subfacts* which are created to represent these partial facts. Each subfact must contain at least two of the bio-entities in the original main fact. Subfacts may also be broken down to represent less informative instances. The creation of subfacts (and subfacts within subfacts) is governed entirely by whether an instance is found that expresses part of the fact in question.

5. Many instances cannot be directly linked to their corresponding fact, as they *depend* on additional information from other passages within the full text or external domain knowledge. To represent these *dependencies*, new fact types are created — *synonym*

facts and *extra facts*. All instances of these, within the same article, are annotated and a dependency link is added between the original instance and the new dependency fact. If an instance cannot be identified for a dependency fact it is labelled as *undefined*.

6. Each instance is annotated with its location within the article, and linguistic phenomena, including negated and coreference expressions, which must be resolved for the MIM fact to be inferred.

2.3.2 Example Corpus Annotation

Consider the M4 MIM description in Figure 2.2. A main fact corresponding to the second sentence of this MIM description was created and is shown below:

> Activation by pRb and c-Myc is not additive, suggesting they act upon the same site, thereby perhaps blocking the binding of an unidentified inhibitor.

Due to the complexity of the interaction relationships stated within this main fact, no single sentence or passage of text supporting this entire fact was identified within the cited article. However, instances of the M4 Subfact 1:

> Activation of E-cadherin by pRb and c-Myc is not additive

and M4 Subfact 2:

> Activation of E-cadherin by pRb and c-Myc

were identified, and thus these subfacts were created to represent this partial knowledge. Instances of these subfacts were located in the results and discussion sections of the cited article, and are shown in Examples 1 and 2 in Figure 2.3. Both of these instances depend on the resolution of synonyms to link the instances to the MIM description. Example 1, depends on two synonym facts. Synonym 1, states that the bio-entity *pRb* used by Kohn is equivalent to the bio-entity *RB* from the cited article, and no instance of this fact was identified in the article. Synonym 2 is required to map the term *c-Myc* to the synonymous term *Myc*, and an instance was identified in the introduction section of the cited article, as shown in Figure 2.3.

1. **M4 Subfact 1**
   Activation of E-cadherin by pRb and c-Myc is not additive
   Instance 1

   - *However, the precise molecular mechanisms by which RB, Myc, and AP-2 cooperate to effect transcriptional activation of E-cadherin requires further study. Interestingly, the positive effects of RB and c-Myc were not additive (Fig. 1).* (Discussion, PMID: 9632747)

   Synonym fact 1
   pRb equivalent to RB

   - Undefined

   Synonym fact 2
   c-Myc equivalent to Myc

   - *The c-myc proto-oncogene, which encodes two amino-terminally distinct Myc proteins, acts as a transcription factor.* (Introduction, PMID: 9632747)

2. **M4 Subfact 2**
   Activation of E-cadherin by pRb and c-Myc
   Instance 1

   - *All of these results indicate that RB and c-Myc transactivation of E-cadherin expression is specific to epithelial cells and requires an active RB protein family.* (Results, PMID: 9632747)

   Synonym fact
   pRb equivalent to RB

   - Undefined

3. **E21 Main fact**
   p107 promoter contains E2F recognition elements and can be repressed by pRb and p107
   Instance 1

   - *Differential Roles of Two Tandem E2F Sites in Repression of the Human p107 Promoter by Retinoblastoma and p107 Proteins* (Title, PMID: 7791762)

   Synonym fact
   Retinoblastoma protein equivalent to pRb

   - *The retinoblastoma protein (pRb) is a 105- to 110-kDa nuclear phosphoprotein (48) with tumor suppressor function (4,32) and is believed to be a negative growth regulator (24, 63).* (Introduction, PMID: 7791762)

   Instance 2

   - *Both pRb and p107 can repress expression of the human p107 promoter through the 5' copy of the E2F-binding site.* (Results subheading, PMID: 7791762)

Figure 2.3: Example instances depending on synonym facts

In order to create a corpus with as much coverage as possible for MIM main facts and subfacts, individual instances or parts of them may be associated with different facts. For example, the two consecutive sentences in Example 1 can only support M4 Subfact 1, however the first sentence also supports M4 Subfact 2. Thus the first sentence is also annotated as another instance of Subfact 2.

2.4 Dependencies

A goal of the MIM corpus is to depict how the coherent flow of knowledge is presented or assumed within full-text articles. The dependency annotation is introduced when a main fact or subfact of a MIM description may not be entirely derived from the text of an instance alone. These instances depend on additional factual knowledge (*dependencies*), which may or may not be present in the same article, to allow the original MIM fact to be derived. This section discusses the two types of dependencies annotated: synonym facts and extra facts.

2.4.1 Synonym Facts

The frequent use of synonyms, metonyms, abbreviations and acronyms in biomedical text is a common source of ambiguity that is often hard for automated methods to resolve (Sehgal et al., 2004). Furthermore, manually curated lists of these are difficult to maintain in rapidly moving fields like biology (Lussier et al., 2006). As a result there is considerable interest in developing systems to identify and extract these, e.g. Ao and Takagi (2005), Okazaki and Ananiadou (2006) and McCrae and Collie (2008). However, there has been no investigation into the difficulties which arise from synonym use when automatically identifying cited facts from full-text articles.

In the MIM corpus, all synonyms, metonyms, and abbreviations, acronyms and other orthographic variations of bio-entities, excluding case changes, which need to be resolved to identify the original MIM fact, are annotated as *synonym facts*. For example, the synonyms (1) *E2F4*, (2) *E2F-4* and (3) *E2F1-4* in the corpus refer to the same entity *E2F4*, however the third term

also includes the entities *E2F1*, *E2F2* and *E2F3*. The synonyms for other terms, such as verbs, in the MIM fact are not included.

In the first example in Figure 2.3, the instance which supports M4 Subfact 1 depends on two synonym facts. In the MIM description, the entity terms *pRb* and *c-Myc* are used, but in the relevant cited article (Batschè et al., 1998) their synonymous terms, *RB* and *Myc*, are mentioned. Therefore, there is a need for a synonym fact dependency.

To link the text in Instance 1 to the subfact, we must first identify that *pRb* is synonymously equivalent to *RB*, and form Synonym Fact 1 to represent this knowledge:

pRb is equivalent to RB

The next step is to identify all passages from within the same article which support this synonym fact. However, an instance was not identified and the synonym fact is labelled as *undefined*. This example highlights the ambiguity introduced when authors choose to use terms other than the ones within the articles they cite, and when synonymous terms are assumed to be general domain knowledge. In addition to this ambiguity, the term *RB* is also an abbreviation for *respiratory bronchiolitis*, *repetition blindness*, *ruminal buffer*, and *Rio bravo virus*, to name a few, and a homograph for the gene *ruby* (*rb*), *rabbit* (*rb*) and *rubidium* (*Rb*).

Instance 1 also depends on Synonym Fact 2:

c-Myc is equivalent to Myc

In this instance, the bio-entities *c-Myc* and *Myc* are used interchangeably, where the protein *Myc* is referred to by its gene name, *c-Myc*. The use of metonymy, where an entity can be substituted with another related entity, is common in biomedical literature, and an instance supporting this type of synonym fact was found in the introduction of the article. This synonym instance does not contain any common contextual patterns such as:

1. *X known as Y*
2. *X* (*Y*)

that are often used to extract sets of synonymous terms, such as X and Y (Yu et al., 2002; McCrae and Collie, 2008). Therefore, further processing to identify these synonyms via the causal relationship, *c-myc encodes Myc*, is required. After these synonymous terms are resolved we can directly infer the M4 Subfact from the instance.

Example 3 in Figure 2.3, shows two annotated instances of the E21 Main fact, where only the first instance depends on a synonym fact to be resolved. The authors of the cited article used the long form of the bio-entity *pRb*, *Retinoblastoma Protein*, in the article's title (Instance 1). Thus to link the first instance to Kohn's MIM fact, these terms must be identified as synonymous. An instance supporting this synonym fact was identified in the introduction section of the article, stating the synonym fact clearly:

> *retinoblastoma protein (pRb)*

After this statement, all later references to this protein in the article (excluding coreference expressions) used the shorter form, and thus Instance 2, identified in a result's subheading, does not depend on this synonym fact to infer the main fact.

2.4.2 Extra Facts

Instances in the MIM corpus may also depend on extra information for the MIM fact to be inferred, which cannot be expressed by synonym fact dependencies. *Extra fact* dependencies were created to represent this additional information need. Extra facts include all assertions (excluding synonym definitions) which are necessary to infer a main fact or subfact from an instance. Many extra facts are descriptions or classes of bio-entities, hyponym relationships and compounded terms. For example, in the extra fact:

> S465A-Abl is a mutated form of c-Abl where Serine 465 is substituted for Alanine

the bio-entity *S465A-Abl* is not a synonym of, but a modified form of the protein *c-Abl*.

If additional information is required for an instance to support a MIM fact, an extra fact is created. All supporting instances of these extra facts must then be identified within the same

4. **E13 Subfact**
HDAC1 binds to the pocket proteins p107 and p130 and in turn is recruited to E2F complexes on promoters
Instance

- *The experiments described above indicate that p107$_1$ and p130$_1$ can interact with HDAC1$_2$. We thus reasoned that they$_1$ could repress E2F activity by recruiting histone deacetylase$_2$ activity to E2F containing promoters.* (Results, PMID: 9724731)

Extra fact
HDAC1 is a histone deacetylase

- *We have previously shown that Rb, the founding member of the pocket proteins family, represses E2F1 activity by recruiting the histone deacetylase HDAC1.* (Abstract, PMID: 9724731)

Figure 2.4: Example instances depending on extra facts

article as the original dependent instance. Examples of extra fact dependencies are shown in Figures 2.4 and 2.5.

Example 4 (Figure 2.4) shows an instance of Subfact E13, which depends on the extra fact:

> HDAC1 is a histone deacetylase

to derive the subfact. The first sentence of this instance states the binding relationships between the bio-entities *HDAC1* and *p107*, and *HDAC1* and *p130*. This sentence does not support the entire subfact individually as the second sentence introduces the fourth required bio-entity, *E2F*. The extra fact is required to associate the class of proteins referred to in the second sentence using the term *histone deacetylase*, to the specific protein *HDAC1* in sentence one. This is necessary as the sortal anaphor *they* in sentence two refers to the bio-entities *p107* and *p130* in sentence one, and not *HDAC1*. An instance supporting this extra fact was identified within the article's abstract, and is expressed in the apposition:

> *the histone deacetylase HDAC1.*

Once this extra fact is identified, the coreference expressions can be resolved, and in turn, the E13 Subfact can be inferred.

Two additional examples of extra fact dependencies are shown in Figure 2.5. Annotated instances of the C2 Main fact and one of its subfacts are shown. In Example 5, the instance supports the main fact only after two extra fact dependencies are established and resolved. These extra facts are required to map the bio-entity terms in the main fact to their corresponding terms/phrases in the instance.

The first extra fact represents the mapping between the bio-entities *E2F4* and *E2F4/DP1*. It depicts a common representation of compounded bio-entities by using a slash (/) to represent a complex of multiple entities.[2] This extra fact is identified in the instance by the following apposition:

the heterodimeric transcription factor E2F4/DP1

where the term *heterodimeric* states that *E2F4/DP1* is a complex composed of two different proteins.

The main fact's instance also depends on additional factual knowledge to associate the coreference concept *a pocket protein* to the bio-entities *p130* and *p107* stated in the main fact. To represent this dependency, another extra fact is created:

pRb, p107 and p130 are pocket proteins

and an instance defining this concept was identified within the article's abstract. The extra fact can be extracted from the expanded enumeration:

the 'pocket proteins': pRB, p107, p130

This extra fact's instance also details a hierarchy of bio-entity concepts where *the pocket proteins* are part of the concept *pRB tumor suppressor family*.

In Example 6, the instance of the C2 Subfact also depends on extra facts. The first extra fact is required to identify the bio-entities *p107* and *p130* in the subfact definition as members of the *pRB protein family* stated within the instance. The instance supporting this extra fact is the same text that supports Extra fact 2 of Example 5.

[2]The slash notation is also often used to represent synonymous terms.

5. **C2 Main fact**
CERC is a complex of E2F4, DP1 and either p130 or p107, and an extra unidentified component
Instance

- *Altogether, these experiments show that CERC is a high molecular weight complex, stable in solution, which contains E2F4/DP1, a pocket protein and at least one additional unidentified protein.* (Results, PMID: 10202151)

Extra fact 1
E2F4/DP1 is a complex of E2F4 with DP1

- *(B and C) CERC contains the heterodimeric transcription factor E2F4/DP1.* (Figure legend, PMID: 10202151)

Extra fact 2
pRb, p107 and p130 are pocket proteins

- *They bind to DNA as free heterodimers E2F/DP or associated in larger complexes containing members of the pRB tumor suppressor family (the 'pocket proteins': pRB, p107, p130) and of the cyclin/cdk family (cyclin E/cdk2 and cyclin A/cdk2 associates physically with p107 and p130).* (Abstract, PMID: 10202151)

6. **C2 Subfact**
CERC contains E2F4 and either p130 or p107
Instance

- *CERC contains members of the E2F and pRB protein families* (Figure heading, PMID: 10202151)

Extra fact 1
p107 and p130 are members of the pRB protein family

- *They bind to DNA as free heterodimers E2F/DP or associated in larger complexes containing members of the pRB tumor suppressor family (the 'pocket proteins': pRB, p107, p130) and of the cyclin/cdk family (cyclin E/cdk2 and cyclin A/cdk2 associates physically with p107 and p130).* (Abstract, PMID: 10202151)

Extra fact 2
E2F4 is a member of E2F protein family

- *E2F's transcriptional activity is the result of the heterodimeric association of two families of proteins, E2Fs (E2F1-6) and DPs (DP1-2) (for reviews on E2F, see Sardet et al., 1997; Dyson, 1998; Nevins, 1998).* (Abstract, PMID: 10202151)

Synonym fact required by Extra fact 2
E2F4 is contained in the range of entities E2F1-6

- Undefined

Figure 2.5: Example instances depending on extra facts

The second extra fact is necessary to associate the bio-entity *E2F4*, which is not mentioned in the subfact instance, as a member of the *E2F protein family* stated in the instance. An instance supporting this extra fact was identified, and defines the *E2F protein family* as:

E2Fs (E2F1-6)

This definition introduces additional complexity as the bio-entity *E2F4* is not directly mentioned. The term *E2F1-6* corresponds to multiple bio-entities, including *E2F4*. This information is represented in the corpus as a synonym fact, which defines *E2F1-6* as a synonymous term for *E2F4*, and thus this extra fact instance also has a dependency fact.

2.4.3 Dependency Graphs

The MIM corpus represents each of the main facts and subfacts as a dependency graph of instances, each of which in turn may depend on other factual knowledge from synonym and extra facts. Each edge in the graph links an instance to each of its dependency instances. It is possible for an instance of a dependency fact to also depend on synonym and/or extra facts, as shown in Example 6 in Figure 2.5, where the instance of Extra fact 2 depends on a synonym fact. Thus paths of dependencies may occur, all of which would need to be resolved before the main fact or subfact could be derived from the initial instance.

2.5 Linguistic Phenomena

The previous sections, introduced the process of formulating main facts and subfacts from the MIM descriptions, and identifying supporting instances of these to annotate, along with any synonym or extra fact dependencies they require. This section discusses the linguistic phenomena individual instances are annotated with. In the MIM corpus, only the linguistic constructs in individual instances that need to be resolved to infer a fact are annotated.

7. **A2 Subfact**
ATM phosphorylates c-Abl
Instance

- *Incubation with [mutant ATM kinase] did [not lead to c-Abl phosphorylation] (fig. 3d, lane 2).* (Results, PMID: 9168116)

8. **N4 Main fact**
RPA2 binds XPA via the C-terminal region of RPA2
Instance

- *[Mutant RPA that lacked the p34 C terminus] [failed to interact with XPA], whereas RPA containing the p70 mutant (Delta RS) interacted with XPA (Fig. 2).* (Results, PMID: 9168116)

Synonym fact
p34 is equivalent to RPA2

- Undefined

9. **C30 Subfact**
Gadd45 inhibits Cdk1 activity
Instance

- *With the use of an antisense approach, [reduced Gadd45 expression] [attenuated the suppression of Cdc2/Cyclin B1 activity] in UV-irradiated human cells.* (Abstract, PMID: 10362260)

Synonym fact
Cdc2 is equivalent to Cdk1

- Undefined

Extra fact
Cdc2/Cyclin B1 is a complex of Cdc2 and Cyclin B1

- Undefined

10. **C9 Subfact**
Rapid degradation of Cyclin D1 requires phosphorylation at threonine-286
Instance

- *Although "free" or CDK4-bound cyclin D1 molecules are intrinsically unstable (t1/2 < 30 min), a [cyclin D1 mutant (T286A) containing an alanine for threonine-286 substitution] [fails to undergo efficient polyubiquitination] in an in vitro system or in vivo, and it is markedly stabilized (t1/2 approximately 3.5 hr) when inducibly expressed in either quiescent or proliferating mouse fibroblasts.* (Abstract, PMID: 9136925)

Figure 2.6: Example instances with negated expressions annotated

2.5.1 Negated Expressions

The purpose of the negated expressions annotated in the MIM corpus is different to that of the BioScope corpus (Vincze et al., 2008) and the BioInfer corpus (Pyysalo et al., 2007). In the BioScope corpus, negative terms in sentences, such as *not* and *neither*, and their scope are annotated for the purpose of developing systems that can detect uncertain facts or negative findings (Vincze et al., 2008). The BioInfer corpus is similarly annotated with negated expressions. However, the annotated phrases correspond to those stating an absence of a relationship between entities, such as *X not affected by Y* (Pyysalo et al., 2007).

The negated expression annotation in the MIM corpus extend those in other corpora by focusing on statements that do not directly express a MIM fact, but from which the fact can be logically implied. The negation annotations include logical negatives, as in the BioScope corpus, and lexical negatives, which have not been annotated in either of the other corpora. Logical negatives are realised by a discrete, closed class negative particle like *not* or *no*. In lexical negatives, the negation is built into the lexical item, like *inhibit* or *mutant*. In these cases, the negated expression entails the opposite of a fact that would need to be worded differently.

As the MIM corpus is focused on molecular interactions, the main type of negated expressions identified correspond to statements describing modifications to molecules and their resulting effects. These statements document the outcomes of experiments from which one can identify/infer a molecule's function by modifying the molecule and observing any functional changes. For example, if in a gene knockout experiment we find that removing gene X results in function Y disappearing, we could infer that gene X is responsible in some way for function Y. In the literature, negated expressions are commonly used to describe these types of experiments, from which the normal function is inferred by the author and the reader. This typically requires two or more negated expressions to be processed simultaneously, as will be shown in the following examples.

Figure 2.6 shows four example instances of different MIM facts which require negated expressions to be interpreted for their corresponding fact to be inferred. The negated expressions

are marked by square brackets, and as in Vincze et al. (2008) the full scope of the negated expressions are annotated.

In Example 7, two negated expressions within the instance need to be resolved to form the positive statement of the A2 Subfact. In this instance, the negated expressions are clearly defined. In the first negated expression, the lexical negated form of the bio-entity *ATM* is stated as *mutant ATM kinase*. The second negated expression is a logical negative which states the function the mutated form of *ATM* was unable to perform. Based on knowledge of the experimental aims and the implicit reporting of the results, one can logically combine these two negated expressions to infer the positive fact expressed in the A2 Subfact. If an IE system could not identify these negated expressions, then the incorrect relationship:

ATM does not phosphorylate c-Abl

may be extracted. It is these types of relationships that the MIM corpus aims to capture, with the goal of identifying processing errors like this.

The negated expressions annotated in Example 8, N4 Main fact, are similar in style to those in Example 7, however the processing to resolve the fact from this instance is more complicated. First, the synonymous terms *p34* and *RPA2* need to be resolved. Secondly, the first lexical negative has wider scope than in Example 7, as it states the specific type of mutation. This additional information is required to infer the main fact completely. As in Example 7, the first lexical negative expression is also followed by a logical negative expression, and these two negated expressions must be inverted and then combined to recover the main fact.

Negated expressions in the MIM corpus are not just identified by the presence of negated terms or lexical negative statements. For a negated expression to be annotated, the positive form of the expression needs to be resolved to infer its associated fact. For example, the instance of Example 8 also contains the lexical negative expression:

p70 mutant (Delta RS)

however as its resolution is not required to infer the corresponding MIM fact, it is not annotated.

Many of the negated expressions annotated in the MIM corpus contain the terms *mutant* or *mutation*, which stem from the specific types of experimental studies performed in this domain. However, this is not always the case. Consider, for example, the subfact in Example 9 (Figure 2.6). The subfact description is itself a logical negative expression, and there is no reference to a mutated bio-entity in the supporting instance. The first logical negative expression, states the result of a different experimental technique (*an antisense approach*), which aims to silence or reduce the activity of the *Gadd45* protein. And the second negated expression declares the result of this reduction, i.e. the *reduction* resulted in the *attenuation*. In the second negated expression, the nested negative phrase:

suppression of Cdc2/Cyclin B1 activity

correctly matches the action of *Gadd45* stated in the MIM subfact, and it is thus not annotated. By inverting the two annotated negated expressions, we get the positive expression:

Gadd45 expression suppresses Cdc2/Cyclin B1 activity

and thus the MIM fact can be inferred from these negated expressions.

The last example in Figure 2.6 captures the complexity of negated expressions in the MIM corpus. The first negated expression is similar to the lexical negative expressions in the previous examples stating a mutation of *cyclin D1 at threonine-286* directly. However, the second negated expression states that the mutated protein is unable to be *polyubiquinated*, which is not mentioned in the C9 Subfact. In turn, the inverted forms of these negated expressions do not directly convey the MIM subfact, and thus external domain knowledge is required.

In the first negated expression, we not only need to identify that the *threonine* required in the MIM fact is no longer a part of *cyclin D1*, but that the amino acid, *alanine*, it was substituted with cannot be *phosphorylated*. At this point, there is still no mention of *degradation*, however with domain knowledge this can be inferred from the second negated expression, as *polyubiquitination* of a protein triggers a signal for the protein to be degraded.

11. **A1 Subfact**
c-Abl is in a complex Rad51
Instance

- *Also, the finding that DNase has no effect on the coimmunoprecipitation of c-Abl and HsRad51 indicated that the association between these proteins is not dependent on DNA binding (data not shown).* (Results, PMID: 9461559)

Synonym fact
Rad51 is equivalent to HsRad51

- *The finding that human Rad51 (HsRad51) promotes homologous pairing and strand exchange reactions in vitro has suggested that Rad51 may also play a role in recombinational repair in man (26)* (Introduction, PMID: 9461559)

12. **C36 Main Fact**
Cdc25C is phosphorylated by Cyclin B-cdk1
Instance

- *In the work reported here, we examine the effect of phosphorylation on the human cdc25-C protein (Sadhu et al.,1990). We show that this protein is phosphorylated during mitosis in human cells and that this requires active cdc2-cyclin B.* (Introduction, PMID: 8428594)

Synonym fact 1
Cdc25C is equivalent to cdc-25C

- Undefined

Synonym fact 2
cdc2-cyclin B is equivalent to Cyclin B-cdk1

- Undefined

Synonym fact 3
cdk1 is equivalent to cdc2

- Undefined

13. **A4 Subfact**
c-Abl inhibits Mdm2-mediated degradation of p53
Instance

- *We demonstrate that c-Abl increases the expression level of the p53 protein. The enhanced expression is achieved by inhibiting Mdm2-mediated degradation of p53.* (Abstract, PMID: 10085066)

14. **P20 Subfact**
PCAF acetylates p53
Instance

- *Here we show that p53 is acetylated in vitro at separate sites by two different histone acetyltransferases (HATs), the coactivators p300 and PCAF.* (Abstract, PMID: 9254608)

Figure 2.7: Example instances annotated with coreference expressions

2.5.2 Coreference Expressions

When automatically extracting information about a bio-entity, such as the interactions it is involved in, it is important to identify all textual references to that entity within the text, to ensure all information is retrieved. These textual references, for example, *it, they* and *these*, are called coreference expressions. In biomedical literature, coreference expressions are frequently used to make abbreviated or indirect references to bio-entities or events.

Few biomedical corpora include annotations of coreference expressions. Castaño et al. (2002) annotated 100 MEDLINE abstracts with both pronominal and sortal anaphors and their corresponding antecedents. A total of 116 unique anaphoric expressions were annotated. These annotations were specifically added to the corpus to assist in the development of a biomedical anaphora resolution system. The BioInfer corpus (Pyysalo et al., 2007) is also annotated with anaphoric coreference expressions. The corpus contains 1100 sentences and 145 unique anaphoric expressions annotated, and is one of the most comprehensive in terms of other linguistic annotations. However, the coreference annotations do not span across sentence boundaries. As we will see in Section 3.5, a significant proportion of coreference expressions occur across sentences.

To quantify the importance of coreference expressions, instances in the MIM corpus are annotated with pronominal, sortal and event anaphoric expressions, and apposition, including those referring to terms within another sentence. As in the negated expression annotations, only coreference expressions which need to be resolved to infer the MIM fact are annotated. Examples of annotated coreference expressions are shown in Example 4 (Figure 2.4), 10 (Figure 2.6) and 11–14 (Figure 2.7). The coreferring expressions and their referred terms are underlined with a single line.

In Example 11, the relationship of the A1 Subfact is indirectly stated in the instance as:

association between these proteins

which can be expanded to:

association between c-Abl and HsRad51

which directly states the relationship. For an IE system to identify this relationship, the sortal anaphoric expression *these proteins*, which is syntactically closer to the instance's relationship statement, would need to be linked to the proteins *c-Abl* and *HsRad51*.

A similar sortal expression appears in Example 4 (Figure 2.4), where the pronoun *they* in the second sentence refers to the proteins in the first sentence. However, this anaphoric expression is more complex to resolve. Firstly, the anaphor *they* does not specify what type of bio-entity it is referring to. Secondly, it is used to refer to only two of the three proteins (*p107* and *p130*) in the previous sentence. The third protein, *HDAC1*, is referred to in the second sentence with the anaphoric expression *histone deacetylase*. These anaphoric expressions need to be resolved, along with the dependencies, to link the information in both sentences together to form the MIM fact.

In the MIM corpus, the anaphoric expressions, described above, which refer to single or multiple entities, are distinguished from those that refer to events such as molecular processes. The MIM corpus *event anaphora* annotations differ to those described by Humphreys et al. (1997), who link different sequential events together. The MIM corpus annotations provide links between references to the same events when their resolution is required to identify the MIM relationships.

Two examples of annotated event anaphora are shown in Figure 2.7. Event anaphoric expressions are underlined with dashed lines. Example 12, is complicated as it not only contains an event anaphoric expression, but it contains two *this* terms. The first *this* is guided by the restricting modifier *protein*, and refers to the protein *cdc25-C* in the first sentence. The second *this* is the event anaphoric expression that refers to the phosphorylation event, *phosphorylated*. These two anaphoric expressions help resolve the MIM fact by linking the bio-entities in each sentence to the phosphorylation relation described.

In Example 13, the event statement in the first sentence indirectly details part of the relationship described in the MIM A4 Subfact. It is in the second sentence, that the specific MIM fact relationship is described, however the event anaphoric expression, *the enhanced expression*, needs to be resolved first. This event anaphor links the two interaction relationships together,

which in turn introduces the bio-entity *c-Abl* in the first relationship to be syntactically associated with the second relationship (*inhibiting Mdm2-mediated degradation of p53*).

In the MIM corpus, coreference expressions realised through apposition within instances, which need to be resolved to infer their associated MIM facts, are also annotated. Appositional phrases are typically used to provide an alternative description or name for an entity. An example of coreference realised through apposition is shown in Example 14. In Example 14, to identify the *acetylating* relationship between the bio-entities *PCAF* and *p53*, where *PCAF* is syntactically distant from the relationship statement, the coreference link between the appositional phrase:

the coactivators p300 and PCAF

and the noun-phrase:

two different histone acetyltransferases (HATs)

needs to be resolved. This noun-phase refers to the proceeding bio-entities *p300* and *PCAF*, and is syntactically closer to the relationship statement. Therefore, once the apposition is resolved the relationship between the bio-entities, *p53* and *PCAF* (as well as *p53* and *p300*), can be established.

2.6 Summary

This chapter described the *Molecular Interaction Map* (MIM) corpus, which maps interaction facts to annotated passages (instances) within full-text articles. The corpus uniquely models the process of forming an interaction fact based on the knowledge presented within a full-text article. The annotation schema captures the complexities which arise when extracting interactions from full-text. The MIM corpus annotates two types of factual dependencies that must be resolved to extract a fact completely from an instance — synonyms and extra facts. Instances are further annotated with negated and coreference expressions that must be resolved

to infer its corresponding interaction fact. The MIM corpus is the first to identify the importance of these negated expressions.

The detailed annotations in the current MIM corpus introduce a number of unexplored challenges for extracting biomedical knowledge from full-text articles, and I expect the MIM corpus to be an informative resource for those developing IE systems. The MIM corpus complements the GENIA and BioInfer corpora, which were created specifically for the development of IE tools. As the MIM corpus documents mentions of interactions, an obvious future enhancement would be to annotate the instances with the named entities and relationships specified in either the GENIA or BioInfer ontologies.

In the following chapter, the MIM corpus is analysed in detail. It demonstrates the importance of full-text processing, identifying synonym and extra fact dependencies, and resolving coreference and negated expressions, for accurate IE.

Chapter 3

Corpus Analysis

The primary goal for developing the MIM corpus was to explore the possible advantages of, and complexities which may arise when, extracting molecular interactions from full-text articles as opposed to just abstracts. The MIM corpus complements other biomedical corpora available that focus on fine-grained annotations for training specific IE systems, such as named entities and their sentence-level relationships. Furthermore, it provides the foundation for quantifying the relative importance of various NLP tasks in biomedical IE.

This chapter presents a detailed evaluation of the MIM corpus. My analysis demonstrates the necessity of full-text processing; identifies the article sections where interactions are most commonly stated; and quantifies the proportion of interaction statements requiring coherent dependencies. Further, the relative importance of resolving negated and coreference expressions is calculated.

The MIM corpus also provides the first insight into the level of fact redundancy within biomedical articles. This is possible because the annotation involved exhaustively identifying and annotating all mentions of the MIM facts in each cited article. This property also facilitates the use of the corpus as the first gold-standard test set for a full-text biomedical sentence retrieval system, which is presented in Section 3.6. This test set allows the exploration of not only the characteristics of the false positives but also the false negatives.

3.1 Overview

The MIM corpus consists of text segments taken from 78 full-text articles used as references by Kohn (1999). In total, I have identified and annotated 2162 sentences from these articles, which document the facts contained in 76 MIM summaries of molecular interactions by Kohn. Table 3.1 shows the distribution of the various fact types which have supporting instances identified and annotated. To reverse engineer the knowledge presented in the 76 summaries, 134 different main facts were created. Of these, 107 main facts had supporting text identified in their corresponding articles, and a total of 363 different instances of these main facts were annotated. The 27 unidentified main facts stated complex information that was not expressed within a single passage of text. However, each of these 'unidentified' main facts are supported by instances of subfacts that express part of the knowledge they convey. Note that since subfacts were only created when instances supporting part of a fact or subfact are identified, all 247 subfacts have supporting instances by definition. There are a total of 729 different facts created, including 135 synonym facts and 213 extra facts, with only 67% (492) of these facts having instances identified. This low percentage primarily results from identifying only 39 of the synonym facts required. The proportion of missing synonym and extra facts shows the importance of creating external resources, such as ontologies and lexicons, and tools for recognising orthographical variants, for the use of IE and IR systems.

3.2 Fact Redundancy

The MIM corpus, unlike other biomedical corpora, consists of annotated instances supporting interaction facts that are repeated in a full-text article, often in different contexts. This is possible, as the annotation process was not restricted to abstracts (GENIA, Ohta et al., 2002; Kim et al., 2008) or single sentences (BioInfer, Pyysalo et al., 2007), which are limited in space and thus the information they can convey. Using full-text articles, which tend to repeat the main findings numerous times, all instances of individual facts can be identified and annotated. As a result, the MIM corpus has a high level of fact redundancy, and this type of redundancy can be

Fact Type	No. Created	No. Identified	No. Instances
Main fact	134	107	363
Subfact	247	247	1468
Synonym fact	135	39	48
Extra fact	213	99	125
Total	729	492	2004

Table 3.1: Distribution of fact types in the MIM corpus

Fact Type	Total dependencies	Synonym fact		Extra fact	
Main fact	76.9	54.0	(10.2)	35.0	(19.6)
Subfact	57.5	34.8	(4.4)	31.9	(14.9)
Synonym fact	10.4	6.2	(2.1)	4.2	(0.0)
Extra fact	19.2	13.6	(0.0)	6.4	(5.6)

Table 3.2: Distribution of instances in the MIM corpus annotated with dependencies
The number in parentheses corresponds to the percentage of instances for which an instance of the dependent fact was identified.

incorporated into systems to improve the extraction process. For example, Brill et al. (2001) and Clarke et al. (2001) showed that redundancy can be exploited by Question Answering Systems by aiding the passage selection components, as retrieved potential answers with high redundancy within documents are often more correct than others. Imperfect systems can also benefit from fact redundancy, as the chances of extracting a fact repeated in different contexts increases. In the MIM corpus, the most redundancy occurs in main facts and subfacts, with on average 3.4 and 5.9 instances each, respectively, while the synonym and extra facts have almost no redundancy. This is another reason for creating biomedical semantic resources.

3.3 Dependencies

Table 3.2 shows the percentage of fact type instances which depend on synonym and extra facts in the corpus. Each dependency type has two percentage measurements associated with it. The first measurement corresponds to the percentage of instances which depend on that fact type. The second measurement, in parentheses, corresponds to the percentage of instances whose dependency is also defined within the same article. These instances are considered to be completely contained within the cited article, requiring no external resources to resolve them.

Breadth	Main fact (%)		Subfact (%)		Synonym fact (%)		Extra fact (%)	
1	160	(44.1)	554	(37.8)	3	(6.2)	19	(15.2)
2	99	(27.3)	246	(16.8)	1	(2.1)	4	(3.2)
3	16	(4.4)	41	(2.8)	0	(0.0)	0	(0.0)
4	1	(0.3)	1	(0.1)	0	(0.0)	0	(0.0)
6	1	(0.3)	0	(0.0)	0	(0.0)	0	(0.0)

Table 3.3: Breadth of instance dependencies

Depth	Main fact (%)		Subfact (%)		Synonym fact (%)		Extra fact (%)	
1	243	(66.9)	762	(51.9)	4	(8.3)	22	(17.6)
2	34	(9.3)	75	(5.1)	0	(0.0)	1	(0.8)
3	0	(0.0)	5	(0.3)	0	(0.0)	0	(0.0)

Table 3.4: Depth of instance dependencies

In total, 76.9% of main fact instances have at least one dependency, with 54.0% and 35.0% depending on at least one synonym fact or extra fact, respectively. However, only 10.2% of main fact instances which depend on a synonym fact have it defined within the same article. Many subfact instances also depend on synonym and extra facts, however fewer of these instances, in particular those depending on synonym facts, are completely contained within the articles — a direct result of the small number of synonym facts created which had a supporting instance identified. Interestingly, some synonym and extra facts depended on other synonym and extra facts, where the majority of these additional dependencies were undefined.

The MIM corpus contains more synonym than extra fact dependencies (Table 3.2), however there are more unique extra facts and more instances of these identified in the articles (Table 3.1). A large proportion of main fact and subfact instances have dependencies (Table 3.2). Since only a small percentage of these dependencies are identified, many of these main facts and subfacts are not completely contained within the articles. This further demonstrates the importance of automatically extracting resources for these types of dependency facts.

As seen in the annotation examples in Chapter 2, a single instance can depend on multiple synonym and extra facts for the original MIM fact to be inferred. For a given instance, the number of dependencies spanning from the instance is referred to as its *dependency breadth*. Table 3.3 shows the degree of dependency breadth for the instances within the corpus. The

first measurement corresponds to the raw frequency count of the fact type, and the second, in parentheses, corresponds to the percentage of instances. In total, 277 of the main fact instances and 842 of the subfact instances have one or more dependencies. Many instances of main facts (44.1%) and subfacts (37.8%) depend on only one fact.

As many instances depend on other facts, it is fortunate that most of the instances depend on less than three different facts. This is because each additional dependency will reduce the likelihood of an instance being identified by an automated system. However, considering that the instances and their dependency facts may occur anywhere within an article, automatically extracting them is still a very challenging task.

In the MIM corpus, an instance of a dependency fact may also depend on synonym and/or extra facts. These are known as *dependency chains*. The facts within a dependency chain must all be resolved before the original fact can be inferred. An example of a dependency chain is shown in Example 6 in Figure 2.5, where Extra fact 2 of the C2 Subfact depends on a synonym fact. For a given instance, the maximum length of the dependency chain is referred to as its *dependency depth*. Instances that depend on an instance, which has no further dependencies, have a dependency depth of 1, and instances with no dependencies have a depth of 0.

Table 3.4 shows the distribution of the dependency depths spanning from instances of each fact type. The majority of main fact (66.9%) and subfact instances (51.9%) have a dependency depth of one. This means that very few instances of dependency facts also rely on additional dependencies — only 34 main fact instances and 75 subfact instances require a chain of two dependencies to be resolved. This distribution is also fortunate, as the introduction of dependency chains is likely to significantly impair an IE system's performance.

3.4 Locating Facts

This section evaluates the relative importance of processing each article as a complete discourse for fact extraction. Each instance in the MIM corpus is annotated with its location within the cited article. These locations include specific article sections, such as abstracts and

Fact location	Main fact		Subfact		Synonym fact		Extra fact	
Title	8.2	(1.5)	8.9	(3.3)	0.0	(0.0)	0.0	(0.0)
Abstract	42.5	(18.7)	38.1	(23.9)	13.3	(13.3)	14.6	(13.1)
Introduction	20.9	(6.7)	32.8	(18.6)	13.3	(12.6)	9.9	(8.0)
Results	43.3	(20.9)	73.7	(35.2)	3.0	(3.0)	16.9	(13.1)
Discussion	36.6	(14.9)	44.5	(21.1)	0.7	(0.0)	3.8	(3.8)
Figure heading	9.7	(1.5)	30.8	(13.4)	0.7	(0.7)	0.0	(0.0)
Figure legend	7.5	(0.7)	16.2	(9.7)	0.0	(0.0)	5.2	(5.2)
Table data	0.0	(0.0)	0.4	(0.0)	0.0	(0.0)	0.0	(0.0)
Methods	0.7	(0.0)	2.0	(0.8)	0.0	(0.0)	4.2	(3.3)
Conclusion	1.5	(0.7)	1.6	(1.2)	0.0	(0.0)	0.0	(0.0)
Footnotes	0.0	(0.0)	0.0	(0.0)	3.7	(3.0)	0.0	(0.0)
Headings	11.2	(1.5)	22.3	(9.7)	0.7	(0.7)	0.5	(0.5)

Table 3.5: Locations of facts within full-text articles
The number in parentheses corresponds to the percentage of facts for which all of their dependencies are identified within the same section.

conclusions, as well as other structures, like the article's title, or headings and captions. Using this data, I can evaluate the informativeness of each section and structure for identifying molecular interactions and specific fact types. By incorporating the detailed dependency annotations, the number of instances that depend on additional facts defined in different sections can be determined.

Table 3.5 shows the percentage of facts that have at least one supporting instance identified within particular article sections. The number in parentheses corresponds to the percentage of instances that are completely contained within a section, that is, the instance and all of its dependencies are identified in the same section. Note that as each fact may have multiple instances, which may be identified in different sections, the percentages do not sum to 100.

Many of the main facts are identified in the results and abstracts of articles. However, these individual sections account for less than 45% of the main facts with instances annotated. More than 70% of subfacts had a substantiating instance within the results section, whereas only 38.1% were identified within abstracts. A large proportion of subfacts were also identified within the section and figure headings, despite their restricted lengths. This provides a clear indication that systems which process only abstracts will be disadvantaged by the significant information loss.

Inst. location	Main fact		Subfact		Synonym fact		Extra fact	
Title	3.9	(0.5)	1.5	(0.5)	0.0	(0.0)	0.0	(0.0)
Abstract	19.0	(7.7)	9.5	(5.9)	37.5	(37.5)	26.4	(24.0)
Introduction	9.1	(2.5)	9.5	(5.1)	37.5	(35.4)	17.6	(14.4)
Results	30.3	(11.3)	39.4	(17.7)	8.3	(8.3)	34.4	(25.6)
Discussion	24.8	(8.3)	19.1	(8.4)	2.1	(0.0)	6.4	(6.4)
Figure heading	4.4	(0.6)	8.9	(4.6)	2.1	(2.1)	0.0	(0.0)
Figure legend	2.8	(0.3)	5.4	(3.3)	0.0	(0.0)	8.8	(8.8)
Table data	0.0	(0.0)	0.1	(0.0)	0.0	(0.0)	0.0	(0.0)
Methods	0.3	(0.0)	0.3	(0.1)	0.0	(0.0)	7.2	(5.6)
Conclusion	0.6	(0.3)	0.3	(0.2)	0.0	(0.0)	0.0	(0.0)
Footnotes	0.0	(0.0)	0.0	(0.0)	10.4	(8.3)	0.0	(0.0)
Headings	5.0	(0.6)	6.0	(3.1)	2.1	(2.1)	0.8	(0.8)
Entire article	100.0	(32.0)	100.0	(49.0)	100.0	(93.8)	100.0	(85.6)

Table 3.6: Locations of instances within full-text articles
The number in parentheses corresponds to the percentage of instances for which all of their dependencies are identified within the same section.

The best sections for identifying synonym facts were the abstract and introduction sections, with few being identified within other locations. This is not too surprising, as it is more appropriate to introduce abbreviations and other synonyms when they are first used within an article. This finding is similar to that identified by Yu et al. (2002).

The majority of extra facts were located in the results and abstract sections. Interestingly, the conclusion and methods sections of the articles rarely contributed any facts. The conclusion sections predominantly discussed speculative ideas and future directions, while the methods sections detailed experimental procedures and conditions.

The location analysis so far indicates the usefulness of each section for expressing different fact types. However, it does not consider the degree of instance redundancy, which can be exploited by systems. More specifically, Table 3.5 only considers if a fact type can be identified within a section, not whether it appears multiple times.

Table 3.6 shows the percentage of instances that are identified in particular article sections, and indicates the level of redundancy within sections. For synonym facts, which often have no instance redundancy, the abstracts and introductions are still the most useful sections for identifying them. Not too surprisingly, the section and figure headings do not express many instances as they are limited in both number and length.

The best sections for finding repeated instances of main facts and subfacts are the results and discussion sections. This contrasts with the results in Table 3.5, where the abstracts were shown to have a significant number of facts identified. This difference is due to the different lengths of these sections and their purposes. For example, abstracts are limited in size and thus the facts they present are rarely stated more than once. Therefore, an IE system restricted to only abstracts must cover all possible ways a molecular interaction can be stated to ensure its extraction, as the system cannot rely on redundancy for validation or for catching a missed fact in another context later.

For the 2002 KDD Cup Challenge, Regev et al. (2002) developed an IR system which specifically focused on limited text sections, such as titles and figure headings. In the MIM corpus, these sections are poorly represented. However, when they do state an interaction fact they do so very concisely.

The following analysis considers the dependencies of different fact types and their instances. If an IE system is restricted to a particular section, as they often are to abstracts, all instance dependencies must also be identified within the same section. When we take into account each instance's dependencies, the results for each section drop dramatically (those in parentheses in Table 3.5 and 3.6). For example, main fact instances which are predominantly expressed in the results section, decreases from 30.3 to 11.3% (Table 3.6). That is, only 11.3% of main fact instances could be completely recovered within the results section alone. This is a direct result of the synonym and extra fact dependencies that are mainly defined in the abstract and introduction sections. This is similarly observed with instances in the abstract section, where the majority of dependency facts are defined elsewhere. These results further demonstrate the need for processing an article as one discourse, rather than as individual disjoint sections, to allow the resolution of synonyms and extra facts stated in different sections, while gaining redundancy coverage from the results and discussion sections.

Annotated Expressions	Instances
Negated	5.5
Coreference	13.0
Anaphora	9.4
Event Anaphora	2.6
Apposition	2.0

Table 3.7: Distribution of annotated expressions within instances

3.5 Negated and Coreference Expressions

In this section, I discuss the relative importance of resolving negated and coreference expressions for extracting molecular interaction mentions from text. Table 3.7 shows the percentage of instances annotated with negated and various coreference expressions in the MIM corpus. The coreference expressions have been separated into three main groups: pronominal and sortal anaphora (anaphora), event anaphora, and appositional phrases. Each of the individual annotated expressions appear in less than 10% of the instances, with standard anaphoric and negated expressions the most predominate. Very few (2.0%) instances are annotated with apposition.

Negated expressions are the second most common linguistic property annotated in the MIM corpus following standard anaphoric expressions. These expressions appear in 5.5% of instances, and pose an interesting NLP task for the automatic extraction of these molecular interaction relationships. The identification of mutation mentions has been investigated by Caporaso et al. (2007) and Erdogmus and Sezerman (2007). However, the cause and effect of mutations with respect to molecular interactions in text has not been investigated.

In total, 13.0% of the instances are annotated with coreference expressions, which are necessary to identify their corresponding facts. The MIM corpus is annotated with 346 unique coreference expressions, with 86 of these crossing sentence boundaries. However, this is only a subset of those appearing in the MIM corpus. In fact, 20% of the instances contain at least one of the following coreferring terms: *these*, *These*, *they* and *They*. Although less than 15% of instances require an anaphoric expression to be resolved, an IE system with an anaphora resolution component must attempt to resolve all anaphoric expressions as it is not known in advance which expressions need resolving. These results suggest that we would expect the

greatest improvement when systems incorporate anaphora resolution components and little improvement from apposition analysis.

3.6 Full-text Sentence Retrieval System

The primary goal motivating the annotations of other biomedical corpora, including GENIA (Ohta et al., 2002; Kim et al., 2008) and BioInfer (Pyysalo et al., 2007), was to create a resource for the development of specific supervised NLP and IE tools. The GENIA corpus was recently the focus of the BioNLP'09 Shared Task on Event Extraction (Kim et al., 2009). One of the main objectives was to identify biological events in text marked with bio-entities. The state-of-the-art performance of 51.95% (F-score) was achieved using a combination of sophisticated NLP methods such as parsing, Support Vector Machines with graph-based features, and rule-based techniques (Björne et al., 2009).

In this section, I consider the related task of identifying sentences and passages that are likely to contain scientific results. I present the first use of the MIM corpus as a gold standard evaluation dataset for a full-text sentence retrieval system. Oracle experiments are carried out to estimate the performance upper bound for different types of keyword queries, and to investigate how much improvement could be achieved if systems can accurately process linguistic phenomena, such as anaphora and negation. Note that this system was developed only to perform oracle experiments to gain performance bounds, and as such it is not a complete functioning IR system.

The task of the retrieval system is to identify sentences from the cited full-text articles that report relationships documented in the MIM. The system takes as input queries composed of different sets of keywords specifically associated with a main fact or subfact in the MIM corpus, and retrieves all sentences matching the keywords. Each search is restricted to a particular article (or articles), which is known in advance to contain the instances supporting the relevant MIM fact the query is linked to. The system does not apply any ranking criteria to the identified sentences, and thus all retrieved sentences are considered equally relevant. The current system is not capable of searching within PDF documents, and as such the MIM corpus is slightly reduced

to include only instances identified within articles in HTML format (63 full-text articles). This reduced corpus will be referred to as the MIM IR corpus and is described in Section 3.6.2.

Since I have exhaustively identified and annotated all of the sentences supporting the MIM facts in the cited articles, the MIM corpus can be used as a gold standard dataset to reliably identify all relevant and irrelevant retrieved sentences and report on the accuracy of the system. Any retrieved sentence that matches the keyword queries for a specific fact, and also appears as an annotated instance of the fact is considered a relevant result, that is, a true positive (TP). Any other sentence retrieved by the system is irrelevant, and is referred to as a false positive (FP). Finally, any fact's annotated instances that are not retrieved by its corresponding queries are false negatives (FN). Based on these counts, the systems performance can be reported in terms of *precision* (P), *recall* (R) and *F-score* (F), which are defined as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.1}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.2}$$

$$\mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}} \tag{3.3}$$

3.6.1 Keywords and Queries

Each of the main facts and subfacts in the MIM IR corpus is assigned a set of keywords associated with their fact and instances. These lists were created semi-automatically by first obtaining the most frequent terms from the instances, excluding any stop words, such as *the* and *it*.[1] This ensures that all main verbs associated with a fact (not only those within Kohn's description) are included. Each list was then manually reduced by a domain expert to include only those associated with the fact. These remaining terms were then divided manually into three keyword classes: bio-entities, verbs, and auxiliary terms.

[1] The stop word list consists of 545 terms.

A2 Subfact ATM phosphorylates c-Abl
Incubation with mutant ATM kinase did not lead to c-Abl phosphorylation (fig. 3d, lane 2).
Keywords
Bio-entities
- *ATM*
- *c-Abl*

Verbs
- *phosphorylate* (*phosphorylation*)

N4 Main fact RPA binds XPA via the C-terminal region of RPA2
Mutant RPA that lacked the p34 C terminus failed to interact with XPA, whereas RPA containing the p70 mutant (Delta RS) interacted with XPA (Fig. 2).
Keywords
Bio-entities
- *RPA*
- *RPA2* (*p34*, *RPAp34*)
- *XPA*

Verbs
- *bind* (*associate*, *complex*, *interact*)

Auxiliary terms
- *C-terminal* (*carboxyl-terminus*, *C-terminus*)

Figure 3.1: Example keywords for A2 Subfact and N4 Main fact

The *bio-entities* consist of all terms referring to the molecules involved, and their synonyms, in the molecular interactions stated in their corresponding MIM fact. Bio-entities that frequently occur in instances, but do not appear in the MIM fact, are excluded. The set of *verbs* includes all terms that describe the interaction relationship, as well as their synonyms and nominalisations. The *auxiliary* term list contains additional terms that were considered necessary by the domain expert to fully identify the entire MIM fact. Auxiliary terms often refer to specific structures within the bio-entities involved in the interaction, and are added manually if they are not identified by the semi-automatic approach. In many cases, no auxiliary terms are specified.

For example, Figure 3.1 shows the instances and keyword lists for the MIM A2 Subfact and N4 Main fact. These correspond to Example 7 and 8 in Figure 2.6. The synonyms are shown in parentheses. Note that, the A2 keywords do not have any auxiliary terms, and the bio-entity

p70 in the N4 instance is not included in the N4 keyword list because it is not part of the molecular interaction fact.

From these keyword categories, five query types are specified:

1. *bio-ent*: sentences must contain all main bio-entity terms or their synonyms;
2. *verb*: sentences must contain all main verbs associated with a MIM fact;
3. *verb syn*: as above, but sentences may contain synonyms for each main verb;
4. *auxiliary*: if a MIM fact is associated with auxiliary terms, sentences must contain these or their synonyms;
5. *any verb*: sentences must contain at least one verb from the set consisting of all verbs associated with all MIM facts in the corpus.

These five classes are then combined to construct queries with various levels of relaxation, such as *bio-ent + verb syn*, which will retrieve sentences matched by both the *bio-ent* and *verb syn* queries.

3.6.2 Preparing the MIM IR corpus

The IR experiments are based on 63 of the 78 full-text articles that were used to construct the MIM corpus. These articles correspond to those which are available in HTML format rather than only PDF. The HTML articles were converted into plain text using the World Wide Web browser Lynx, followed by some manual post-processing to filter out any remaining noise. Individual sentences were identified using a boundary detector based on the MXTerminator (Reynar and Ratnaparkhi, 1997). Manual post-processing was carried out to correct consistently mistaken boundaries, such as *et al.* These sentences were tokenised, ensuring single term entities with punctuation, like *E2F-4*, were not split into multiple tokens.

The reduced final dataset contains 19,117 sentences and 363,130 tokens. There are 316 different facts and subfacts identified within this dataset, corresponding to 1635 individual

instances and 1736 annotated sentences. Only 92 instances consist of two or more adjacent sentences, which are all required to infer their associated fact.

3.6.3 Results and Analysis

The performance of the sentence retrieval system is shown in Table 3.8. The number of sentences retrieved, and the precision (P), recall (R), F-score (F), and the distribution of true positive (TP), false positive (FP) and false negative (FN) sentences of each query type, are shown.

The first experiment in Table 3.8 corresponds to the most restrictive query set, requiring all of the keywords for a MIM fact: bio-entities, main verbs, and auxiliary terms to be present in the retrieved sentences. This query is unrealistic because it requires a user's knowledge of the exact relationship stated in each of the MIM facts, including the specific verbs and auxiliary terms used. As a result, this query composition identifies the least number of sentences, and achieves the highest precision of 34.0%, but the lowest recall of 34.6% (65% of the annotated sentences in the MIM IR corpus are not identified, with 1135 FN).

Each subsequent experiment shown in Table 3.8 relaxes the search criteria. These increase the number of sentences retrieved and recall, with the expected or usual decrease in precision. The least restrictive search, *bio-ent*, results in the largest recall and the lowest precision, and returns an enormous number of FP sentences.

There is an improvement in F-score, from 32.2 to 37.3%, when the corresponding verb lists are expanded to include their synonyms (*bio-ent + verb syn*). There are approximately 50% fewer FN, however the number of FP increases by a similar percentage. The best performance of 43% F-score is achieved with the *bio-ent + verb syn + auxiliary* set. The search restriction enforced by the *auxiliary* terms reduces the number of FP by 34%. However, including these terms unrealistically models a user's search style as it relies heavily on prior knowledge of the exact fact. Unfortunately, the most realistic query setting is *bio-ent + any verb*, since it is feasible to enumerate possible interaction verbs without prior knowledge of the specific interaction.

Query type	# Retrieved	P	R	F	TP	FP	FN
bio-ent + verb + aux.	1769	34.0	34.6	34.3	601	1168	1135
bio-ent + verb	2287	28.3	37.3	32.2	647	1640	1089
bio-ent + verb syn + aux.	3277	33.2	62.7	43.4	1089	2188	647
bio-ent + verb syn	4507	25.8	67.1	37.3	1165	3342	571
bio-ent + any verb	8856	14.4	73.4	24.1	1274	7582	462
bio-ent	10232	12.7	74.9	21.7	1300	8932	436

Table 3.8: Sentence retrieval performance

Query type	Neg.	Ana.	Event ana.	Appos.	Extra dep.	None
bio-ent + verb + aux.	6.3	10.0	3.2	1.1	37.5	30.5
bio-ent + verb	6.6	9.9	3.3	1.1	37.0	30.7
bio-ent + verb syn + aux.	5.9	14.3	5.6	1.4	48.5	18.0
bio-ent + verb syn	6.5	14.9	6.5	1.6	47.6	16.8
bio-ent + any verb	6.7	17.2	7.4	2.0	46.5	14.5
bio-ent	6.6	17.6	7.7	2.2	48.9	12.1

Table 3.9: Distribution of linguistic phenomena in false negative instances

Characteristics of False Negatives

Using the MIM corpus, we can determine the linguistic phenomena or dependencies each FN has that are likely to be responsible for it being undetected. This enables us to evaluate the potential impact these phenomena have on the sentence retrieval task and potentially other NLP tasks. Examples 15–17 (Figure 3.2), are FN instances of their respective facts.

Table 3.9 shows the percentage of FN instances containing different types of linguistic phenomena. Note that as the *bio-ent* query contains all entity synonyms, no FN will arise due to lack of synonym knowledge. In all experiments, the majority of the FN instances can be linked to some form of linguistic phenomena. Only 12.1% of the FN resulting from the most relaxed search (*bio-ent*) do not have any linguistic annotations or extra fact dependencies. For example, Instance 15 in Figure 3.2, consists of two consecutive sentences without any coreference expressions. Few FN need negated expressions or apposition to be resolved to identify the sentences. However, these are required to extract the relationship between the bio-entities once the sentences are identified.

15. **P36 Subfact**
TBP binds to an acidic domain in central Mdm2
FN Instance

- *We show that MDM2 binds to the general transcription factor TFIID in vivo. The C-terminal Ring finger interacts with TAF[II]250/CCG1, and the central acidic domain interacts with TBP.* (Abstract, PMID: 9388200)

16. **N6 Subfact**
XPF binds to the C-terminal region of ERCC1
FN Instance

- *Previous mutagenesis studies showed that a 'Rad10-like' [ERCC1 protein, with a stop at residue 214], was functionally inactive (27). This can now be explained by the [inability of this protein to form a complex with XPF].* (Discussion, PMID: 9722633)

17. **P21 Subfact**
p300 acetylates p53
FN Instance

- *Note that incubation with DNA-PK produced a new p53 isoform (labeled 3) that is phosphorylated on Ser-37 as well as on Ser-33. This isoform was preferentially acetylated by p300.* (Figure legend, PMID: 9744860)

18. **A5 Main fact**
c-Abl phosphorylates tyrosines in the C-terminal domain of RNA polymerase II
FP Instance

- *Given the fact that Arg and Abl are highly divergent in the C-terminal region except for the CTD-interacting domain, it is possible that these two kinases may transduce different signals to mediate the tyrosine phosphorylation of RNA polymerase II.* (Discussion, PMID: 9228069)

19. **B3 Subfact**
DNA-PK can bind dsDNA without Ku
FP Instance

- *It seems likely that DNA-PK does not recognize DNA alone initially if Ku is present; rather it binds to some part of Ku or both Ku and DNA in the Ku:DNA complex.* (Discussion, PMID: 9742108)

20. **C43 Main fact**
p16 associates with TFIIH and RNA pol II CTD
FP Instance

- *The possibility that p16[INK4A] might associate with the RNA pol II, a protein substrate of the CTD kinase of TFIIH, was next examined.* (Results, PMID: 9488660)

Figure 3.2: Examples of false negative and false positive instances

Term list	Example terms
Modal verbs	could, should, shouldn't, may, might
Epistemic adjectives	apparent, improbable, likely, possible, unlikely
Epistemic nouns	assumption, claim, doubt, opinion, suggestion
Epistemic adverbials	apparently, maybe, perhaps, presumably, surely
Epistemic lexical verbs	appear, assume, hypothesize, predict, suggest
Indefinite quantifiers	about, generally, often, predominately, sometimes
Speculative terms (Light et al., 2004)	insight, likely, may, suggest, possibly, propose
Commitment terms	confirm, demonstrate, established, indicating, proves

Table 3.10: Examples of hedging and commitment terms

The main linguistic phenomena associated with the FN instances are the anaphoric expressions and extra fact dependencies. For example, 17.6% of FN instances resulting from the *bio-ent* query contain an anaphoric expression, and 48.9% require at least one extra fact dependency to be resolved. Example 16 and 17 show instances where no single sentence mentions each of the queried bio-entities, and an anaphoric expression is required to identify the MIM facts that are conveyed across multiple sentences. Therefore, an anaphora resolution system has the potential to not only improve the relationship extraction process, but also to improve the number of relevant sentences retrieved. Furthermore, the large proportions of FN depending on extra facts, even with the most restrictive search (37.5%), shows that there is a need for systems to identify these dependencies, and treat full-text documents as a complete discourse rather than a set of individual sentences.

Characteristics of False Positives

Results in scientific literature contain various levels of certainty, ranging from speculation to complete confidence. *Hedging* is frequently used in scientific literature to indicate any lack of commitment to a fact (Hyland, 1996). For example, the following sentences express the same proposition between two proteins *RPA1* and *DNA-PK*, however only the first does so with certainty:

1. *RPA1 was sufficient to form a complex with DNA-PK.*
2. *These results suggest that RPA1 interacts directly with DNA-PK.*

This section explores the presence of hedging and commitment within the FP sentences to determine if these characteristics can be exploited to help distinguish between relevant and irrelevant sentences. Hedging has been studied in the citation analysis of scientific literature (Mercer and Marco, 2004), and is also annotated within the BioScope corpus (Vincze et al., 2008). This study is the first to investigate the impact of hedging on biomedical sentence retrieval. For the analysis reported here, the presence of hedging and commitment terms within the TP, FP and FN sentences is considered.

Hedging is typically realised using modal verbs, epistemic adjectives, nouns and adverbials, lexical verbs and indefinite quantifiers (Holmes, 1988). Light et al. (2004) explored the use of speculative language within MEDLINE abstracts and composed a set of 14 speculative terms commonly used. For the analysis, a list of commitment terms, which express a high degree of certainty, was created. Example terms from these lists are shown in Table 3.10.

Examples 18–20 in Figure 3.2 are FP of their respective facts and contain at least one hedging expression. In these instances, the authors are indicating their research aims and hypotheses. However, this is only directly stated in Example 20, indicated by the phrase *was next examined*. In Example 17, the epistemic adjective *possible* and modal verb *may* are used to express open-mindedness about the MIM main fact. This instance also expresses some certainty, using the phrase *Given the fact*, but about another statement.

Epistemic adjectives and modal verbs are also used in Examples 19 and 20, respectively. If we ignore the notion of hedging, only Examples 18 and 19 would match the MIM facts. Although Example 20 contains all bio-entities and the exact verb used to describe their relationships (*associate*), it only substantiates part of the fact (*p16 associates with RNA pol II*).

Table 3.11, shows the results of the hedging and commitment analyses on the output of the sentence retrieval system with the *bio-ent + verb syn* query sets. The majority of the hedging categories occur in less than 7% of TP, FN and FP, with little class discrimination. Epistemic lexical verbs and the speculative words identified by Light et al. (2004) are the most frequently occurring hedging terms within the literature. However, these terms are also not discriminative. The most significant class discrimination is identified with the modal verbs, with a high 15%

Term list	TP	FN	FP
Epistemic adjectives	2.4	2.5	4.6
Epistemic nouns	2.3	2.2	4.0
Epistemic adverbials	2.5	2.5	4.1
Indefinite quantifiers	3.6	6.8	5.3
Modal verbs	8.3	9.5	15.0
Epistemic lexical verbs	12.7	16.3	18.2
Speculative terms (Light et al., 2004)	13.0	14.2	16.1
Any hedging term	25.0	32.0	37.5
Any commitment term	40.3	48.3	36.7
Only hedging terms	15.3	15.4	25.0
Only commitment terms	30.7	31.7	24.0

Table 3.11: Distribution of hedging and commitment terms within instances and false positives

of FP (corresponding to 246 FP) containing at least one modal verb. There is approximately 7% and 6% difference between the TP and FP, and FN and FP, respectively.

The overall importance of hedging terms can be realised by combining the hedging categories into one term list (*any hedging term*), and identifying those TP, FN and FP which contain any of these terms. Approximately 12% more FP contain hedging terms than TP.

When we consider the terms expressing commitment, the TP and FN are discriminated from the FP more. As expected many of the TP and FN contain commitment terms, 40.3% and 48.3% respectively. However, almost as many FP contained positive terms (36.7%) as those containing hedging terms (37.5%).

So far the analyses have not considered the possibility of both hedging and commitment terms appearing in the same sentences, as in Example 18. When we consider sentences with hedging terms and no commitment terms (*only hedging terms*), the FP are separated from the TP and FN almost as much as using the *any hedging term*, with fewer TP and FN matching.

These results indicate that hedging and commitment is common within the MIM corpus, as well as in the FP, and thus these categories cannot be used directly to detect and filter FP. However, they may be useful, in combination with other features, in the development of statistical NLP models for distinguishing between TP and FP.

3.7 Summary

This chapter presented a detailed analysis of the MIM corpus, which in turn has provided invaluable guidance for developers of biomedical IR and IE systems. The corpus analysis demonstrates that full-text processing is crucial for extracting biomedical knowledge. Less than 45% of individual facts and 20% of instances within the MIM corpus were contained within the abstract, and the majority of instances with dependencies rely on statements within different article sections. Further, full-text systems will be able to exploit any fact redundancy that is present.

By analysing the MIM corpus, I have quantified the proportion of interaction instances requiring dependencies and also the amount of external knowledge required. Only 29% of synonyms and 46% of extra facts were identified within the articles. The extra fact dependencies are also the most significant factor preventing the retrieval of instances ($> 35\%$). These results highlight the importance of automated methods for extracting both synonyms and extra facts.

The MIM corpus also provided a means for reporting the relative importance of NLP tools for resolving coreference and negated expressions — 13% and 5.5% of instances require at least one coreference or negated expression to be resolved, respectively, for the original MIM fact to be extracted. This large proportion of instances indicates that these phenomena will also need to be processed for the accurate extraction of many relationships.

These first two chapters have investigated and quantified the importance of numerous NLP challenges, such as resolving negated expressions and identifying dependencies within full-text, which in itself is a significant contribution to the community. Based on these analyses, it is expected that systems will gain significantly by processing full-text articles instead of individual abstracts, and by incorporating coreference resolution components. Furthermore, the automatic identification of semantic resources, such as synonym lists, as well as extra facts and mutations, will be critical for the identification of molecular interactions. In the remaining chapters, I focus on automatically extracting biomedical semantic lexicons, including mutations, that are pertinent to identifying molecular interactions and other biomedical knowledge from text.

Chapter 4

Extracting Semantic Lexicons

Minimally supervised bootstrapping algorithms are commonly exploited to extract large semantic lexicons or relations between semantic categories from a small set of examples. This chapter describes the most influential and successful bootstrapping algorithms. It begins by exploring some of the canonical work of Hearst and Riloff to this task, which have been very influential in the development of many sophisticated and effective bootstrapping methods. Some of the common bootstrapping algorithms used for this task are described, including the single-category and multi-category approaches. Each approach shares a common goal — to extract large precise semantic lexicons or lists of relations, while preventing semantic drift from occurring. I focus specifically on the more recent multi-category algorithms, BASILISK (Thelen, 2001; Thelen and Riloff, 2002) and Mutual Exclusion Bootstrapping (MEB, Curran et al., 2007), as both their advantages and disadvantages have motivated the design of my new algorithm *Weighted Mutual Exclusion Bootstrapping* presented in Chapter 6.

4.1 Information Extraction with Patterns

The central idea of using patterns for extracting semantic lexicons and relations automatically from text was pioneered by Hearst (1992). Hearst exploited manually identified patterns to acquire hyponym relations in text. She identified six common lexico-syntactic patterns, shown below, which can extract a hypernym and its hyponyms from text:

1. <NP_0> *such as* <NP_1> , <NP_2>, . . . , *and* | *or* <NP_n>

2. *such* <NP_0> *as* <NP_1> , <NP_2>, . . . , *and* | *or* <NP_n>

3. <NP_1> , . . . , <NP_n>, *or other* <NP_0>

4. <NP_1> , . . . , <NP_n>, *and other* <NP_0>

5. <NP_0> , *including* <NP_1> , . . . , *and* | *or* <NP_n>

6. <NP_0> , *especially* <NP_1> , . . . , *and* | *or* <NP_n>

From these patterns, the noun phrases NP_1, NP_2, $\ldots NP_n$, are identified as hyponyms of the hypernym noun phrase NP_0. Patterns 1-3 were identified by Hearst by observing hyponym mentions in text. The additional patterns were identified using the seed relations *England* is a *country* (patterns 4-5), and *tank* is a *vehicle* (pattern 6), and finding commonalities between their mentions in text. Using these six patterns, 152 possible hypernym relations were automatically extracted from text in Grolier's American Academic Encyclopedia (Grolier, 1990), which was first shallow parsed. The extracted relations were evaluated by comparing them to the noun hierarchy in WordNet. From the relations, 226 unique words were extracted and 180 of these were identified in WordNet. For these 180 terms, WordNet contained 106 relations for which Hearst's system identified 61 (Hearst, 1992).

Based on the work presented by Hearst, Berland and Charniak (1999) devised five lexico-syntactic patterns for extracting whole-part relationships from text. For example:

1. <NN_0> *in a* | *the* <NN_1>

2. <NN_0> *in* <NN_1>

3. <NN_0> *of a* | *the* <NN_1>

4. <NN_0> *of* <NN_1>

5. <NN_1> *'s* <NN_0>

These patterns extract pairs of single terms, which correspond to the head nouns (NN) of noun phrases, where NN_0 is assumed to be a part of NN_1. For example, the fifth pattern will match the phrase *the building's basement*, and extract a whole-part relationship between the terms *basement* and *building*.

Hearst (1992) concludes by suggesting an iterative process for utilising the newly extracted pairs to automatically identify new patterns for the purpose of extracting more term pairs. Since then, many others have considerably extended her work, and her suggested framework is now commonly referred to as *bootstrapping*. For example, Brin et al. (1998) developed an automated bootstrapping method, DIPRE, to extract book titles and their authors from Web pages. Bootstrapping has also been shown to be especially well suited to acquiring semantic lexicons (Riloff and Jones, 1999; Thelen and Riloff, 2002), which is the main focus of the following discussions.

4.2 Single Category Bootstrapping

4.2.1 Iterative Bootstrapping

Riloff and Shepherd (1997) introduced one of the first bootstrapping algorithms for extracting semantic lexicons from a parsed version of the MUC-4 development corpus (Sundheim, 1992). Their algorithm exploits the observation that members of a semantic category often co-occur with each other in common syntactic constructions: *conjunctions* (*lions and tigers and bears*), *lists* (*lions, tigers, bears, ...*), *appositives* (*the stallion, a white Arabian*) and *nominal compounds* (*Arabian stallion; tuna fish*). The algorithm takes as input a set of five seed words that are members of the semantic category of interest and a parsed corpus, and iteratively identifies additional terms that are hypothesised to be members of the same category. Herein, this algorithm will be referred to as *iterative bootstrapping* (IB).

The first iteration starts by identifying all sentences within the parsed corpus that contain at least one of the five seed terms as a head noun in a noun phrase. Each occurrence of the seeds as head nouns in the corpus is then assigned an extracting contextual pattern. A seed's pattern

consists of two words from its surrounding context that correspond to the closest nouns to its left and right. The set of all matching extracting patterns, P, are then used to identify new unseen candidate terms which may be added to the lexicon. Each candidate term must correspond to a head noun in the corpus that is matched by at least one of the extracting patterns, and have a frequency greater than five in the corpus. From these candidate terms, the *best* terms are added to the lexicon. To identify the best new terms, all candidate terms are scored according to the conditional probability that the term can be identified by the extracting patterns:

$$score(term) = \frac{\sum_{j=1}^{P} \text{freq}(term, P_j)}{\text{freq}(term)} \tag{4.1}$$

The top five scoring candidate nouns, which are most strongly associated with the extracting patterns, are added to the semantic lexicon. This expanded semantic lexicon is then used to initialise the next bootstrapping iteration, in place of the original seed terms. The iterative process terminates after a fixed number of iterations or if no new terms are identified.

Riloff and Shepherd (1997) evaluated IB on five categories in the MUC-4 terrorism corpus (Sundheim, 1992): *energy*, *financial*, *military*, *vehicle*, and *weapon*. The top 200 terms extracted for each semantic category contained on average 35 correct terms ($\sim$17%). Although this low precision reflects a limited degree of success, their work proved to be highly influential. Following Riloff and Shepherd (1997), Roark and Charniak (1998) extended the approach using complete parse trees for constructing patterns and used a log-likelihood metric to re-rank the final extracted lexicon. This method yielded a precision approximately twice that achieved by Riloff and Shepherd (1997) on the *weapon* category.

Both of Riloff and Shepherd's (1997) and Roark and Charniak's (1998) systems are language dependent as they rely on patterns composed of lexico-syntactic information, and restrict the terms that can be extracted to noun-phrases. However, by utilising these types of patterns and terms the search space is pre-filtered of many incorrect terms and generic patterns.

4.2.2 Mutual Bootstrapping

Another variant of IB is known as *mutual bootstrapping* (MB) (see Riloff and Jones (1999)). In IB, all of the matching patterns are used to identify new candidate terms, and only the top scoring terms are extracted. However, in MB all of the extracting patterns are scored and all of the terms matching the top scoring pattern are extracted.

In each iteration of MB, the set of terms in the growing lexicon are used to identify candidate patterns. To identify the best extraction pattern, all candidate patterns are scored using the *RlogF* metric, introduced in Riloff (1996a):

$$RlogF(pattern_i) = \frac{F_i}{N_i} \times \log_2(F_i) \tag{4.2}$$

where F_i is the number of unique terms in the seed lexicon that can be extracted by pattern i, and N_i is the total number of unique terms that pattern i can extract. The *RlogF* metric was designed for IE tasks, where a balance between reliable extraction patterns and those which frequently yield new terms are preferred (Riloff, 1996a). The new lexicon terms correspond to all of the terms that can be extracted by the top scoring pattern, without any scoring or selection restrictions.

The performance of MB was not evaluated completely, and thus there is no available performance comparison between MB and IB. However, Riloff and Jones (1999) noted that the performance of MB deteriorated rapidly once incorrect terms were extracted — that is, semantic drift occurred. MB is one of the main components of *Multi-level bootstrapping* which is described in the following section.

4.2.3 Multi-level Bootstrapping

Multi-level bootstrapping (MLB), described in Riloff and Jones (1999), extends and improves upon MB by incorporating both candidate term and candidate pattern scoring functions. In MLB two separate bootstrapping loops are introduced. The inner bootstrapping loop

corresponds to the MB algorithm described previously. The inner MB loop is normally repeated until 10 different patterns are identified, unless one of two pattern score thresholds are met. If the best pattern's score is below 0.7 or above 1.8, the inner loop can stop before or continue after the 10th iteration, respectively. This inner loop then generates a temporary semantic lexicon, containing all terms that can be extracted by any of the selected patterns.

As noted by Riloff and Jones (1999), the inner MB loop is subject to noise. Noise enters the temporary lexicon when a high scoring pattern, which extracts many correct terms, also extracts incorrect terms. To reduce the degradation of the lexicon an outer bootstrapping loop, referred to as *meta-bootstrapping*, is added. The meta-bootstrapping loop first identifies the top five items in the temporary semantic lexicon to add to the final semantic lexicon, and then restarts the MB loop using the expanded lexicon as the new input seeds. To identify the top five terms, each term is scored using the following function:

$$\text{score}(term_i) = \sum_{k=1}^{N_i} 1 + (0.01 \times RlogF(pattern_k)) \tag{4.3}$$

where N_i corresponds to the set of patterns identified in the MB loop that extracted term i. This scoring function assigns higher scores to terms that are identified by more patterns. The small factor $(0.01 \times RlogF(pattern_k))$, which takes into account the scores of the extracting patterns, is used to break-ties when ranking the terms.

MLB generates semantic lexicons with greater precision than the previous approaches. Using the same parsed, MUC-4 text collection and the same pattern constructs used in Riloff and Shepherd (1997), MLB was shown to boost performance of each semantic category. In particular, the *weapon* category improved from 36% (Roark and Charniak, 1998) to 51% (Riloff and Jones, 1999).

4.3 Multi-category Bootstrapping

When bootstrapping semantic lexicons for a single semantic category, polysemous terms and/or patterns that weakly constrain the semantic class are eventually extracted, causing semantic drift within the lexicon. This ambiguity can occur as the bootstrapping process is not aware of other semantic categories existing and their possible overlaps.

In this section, I describe three multi-category bootstrapping algorithms, BASILISK (Thelen, 2001; Thelen and Riloff, 2002), NOMEN (Lin et al., 2003b; Yangarber, 2003b), and *Mutual Exclusion Bootstrapping* (MEB, Curran et al., 2007). These algorithms aim to reduce semantic drift by extracting multiple semantic categories simultaneously. This enables the algorithms to utilise information about other semantic categories in an attempt to reduce the categories from drifting towards each other while extracting new terms and patterns. This strategy is similar to the *one sense per discourse assumption* (Gale et al., 1992), which states that a term only has a single sense within a set of documents. Multi-category bootstrapping algorithms have been shown to significantly outperform the single-category approaches, and thus the experiments in this thesis focus on these algorithms.

Each semantic category is assigned an individual bootstrapping instance, which is initialised with a small set of seeds and iteratively alternates between two phases: 1) identifying and selecting the best candidate extraction patterns; and 2) identifying and selecting the best candidate terms to add to the lexicon. Figure 4.1 shows the architecture of each individual bootstrapping instance. In multi-category bootstrapping, each iteration of a bootstrapping instance is performed in parallel with the other categories' instances, depicted in Figure 4.2. During the term and pattern selection phases, any conflicts between the semantic categories may be handled. A category *conflict* occurs when two or more categories attempt to select the same term or pattern in an iteration. The top candidate terms and patterns are selected using scoring metrics and knowledge of the other semantic categories to handle any conflicts. In Figure 4.2, the conflict resolution stages are indicated by the two larger boxes surrounding the individual instances. This is the only point when the individual bootstrapping instances interact.

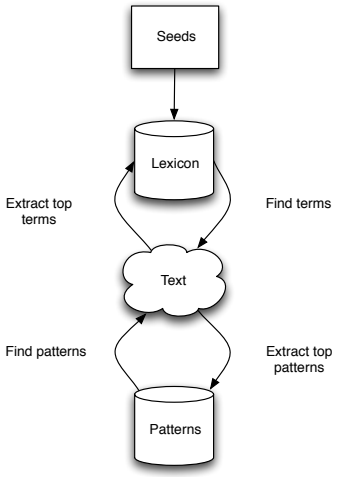


Figure 4.1: Architecture of an individual bootstrapping instance for BASILISK, NOMEN and MEB

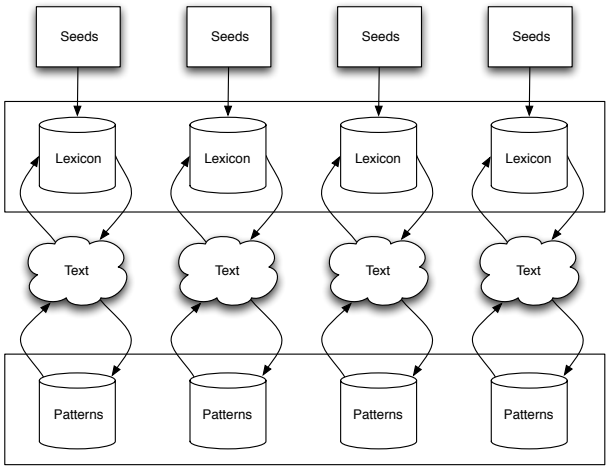


Figure 4.2: Architecture of multiple category bootstrapping for BASILISK, NOMEN and MEB

Each algorithm differs in the types of patterns utilised, their scoring metrics for terms and patterns, and how they incorporate knowledge from multiple semantic categories. Both BASILISK and MEB use fixed patterns, whereas NOMEN generalises the patterns to increase coverage. Category conflicts are only handled during the term selection phase in BASILISK, and at the pattern selection phase in NOMEN. Whereas, in MEB, both term and pattern conflicts are considered. Each algorithm also removes the necessity of the meta-bootstrapping loop of MLB, by accumulating evidence from multiple high scoring patterns within each iteration, to identify new candidate terms. Each algorithm is discussed in detail in the following sections.

4.3.1 BASILISK

The multi-category bootstrapping algorithm *Bootstrapping Approach to SemantIc Lexicon Induction using Semantic Knowledge* (BASILISK, Thelen, 2001; Thelen and Riloff, 2002) was developed to extract higher quality lexicons than MLB (Riloff and Jones, 1999). In Thelen and Riloff (2002), BASILISK is used to extract six semantic lexicons from the MUC-4 corpus: *building*, *event*, *human*, *location*, *time*, and *weapon*. Like MLB, BASILISK focusses on extracting the head nouns of noun phrases.

After each category's bootstrapping instance is initialised with a set of seed terms (single-term nouns), BASILISK utilises heuristics from the AutoSlog system (Riloff, 1996b) to identify candidate patterns. These AutoSlog patterns represent linguistic expressions that identify noun phrases in one of three syntactic roles: subject, direct object, or prepositional phrase (PP) object, and specify either the left or right linguistic context of a term. For example, the following patterns could all extract mentions of *humans*:

<subject> *was murdered*
killing <direct object>
collaborated with <PP object>

In each iteration of BASILISK, all candidate extraction patterns are identified using AutoSlog, and are ranked as in MB, using the *RlogF* metric (Equation 4.2). The top-k scoring patterns are then used to extract all matching candidate terms from the corpus. These can include patterns from previous iterations. In each iteration, the number of extracting patterns (k) is incremented by one. This ensures that at least one new pattern is contributing candidate words in each iteration, which prevents the selected patterns from becoming stagnant (Thelen, 2001; Thelen and Riloff, 2002).

The key idea in BASILISK is that multiple categories are exploited to reduce semantic drift. BASILISK extracts candidate terms which have strong evidence relating it to one semantic category, and little or no evidence for the other categories. Each of the candidate terms for a given

category is assigned a preliminary score ($AvgLog$) according to:

$$AvgLog(term_i, cat_a) = \frac{\sum_{j=1}^{P_i} \log_2(F_j + 1)}{P_i} \tag{4.4}$$

The $AvgLog$ function considers all patterns (P_i) in the corpus which can identify $term_i$, not just the top-k patterns that identified it during the term selection phase. F_j corresponds to the number of terms in the category's (cat_a) expanded lexicon that pattern j can extract.

The final score of a candidate term for a given category (Equation 4.5) penalises a term that is identified by multiple semantic categories. A term's score is reduced by its maximum score obtained in the other categories.

$$score(term_i, cat_a) = AvgLog(term_i, cat_a) - max_{a \neq b}(AvgLog(term_i, cat_b)) \tag{4.5}$$

A candidate term for a category will receive a high score if its scores in the other categories are low. If a term is identified by multiple categories in an iteration, it is only added to the category for which it scores the highest. Using this scoring function and applying this restriction if necessary, the top five candidate terms in a category are added. After the top terms are added, the bootstrapping process in BASILISK is reinitialised using each of the categories' expanded lexicon terms as the seeds.

The term scoring function in BASILISK, typically favours less frequent terms resulting in the extraction of rare terms. This is because terms with low frequency will match far fewer patterns, and are less likely to match patterns assigned to multiple categories, and in turn are more likely to receive a higher score than more frequent terms.

BASILISK considers conflicts during the term selection phase, but not during the pattern selection phase. In addition, patterns which extract terms for a category in one iteration, may extract terms for another category in a different iteration. This may also introduce semantic drift. BASILISK is reportedly 10 times faster than MLB on the basis of the exclusion of the 10

iterations of the inner MB loop (Thelen, 2001). However, his analysis does not consider the increase in complexity of the new term scoring metric used (see Equation 4.5). Firstly, to score an individual candidate term $term_i$, for a given category cat_a, the $AvgLog$ for the term with *all* categories must be calculated. This is required to determine the maximum $AvgLog$ value over all competing categories. Secondly, the $AvgLog$ of a candidate term and category pair is more expensive than the score used in MLB. In MLB, only the extracting patterns are used (up to 10), whereas in BASILISK *all* possible patterns P ($\gg$ 10,000) must be checked to see if they can extract the $term_i$. Lastly, the scoring metric cannot be calculated offline before bootstrapping begins — F_j for each candidate term and category pair may change in each iteration.

Thelen (2001) and Thelen and Riloff (2002) showed that bootstrapping multiple competing semantic categories improved the accuracy of most of the extracted lexicons, in particular, for the *building*, *location* and *weapon* categories. For the *human* category, there was no appreciable difference. They noted the improvements, if any, usually appeared in the later stages of bootstrapping — after the first 200 terms have been extracted. This is because few category conflicts occurred in the earlier iterations since the categories they used are not semantically close. Further, as bootstrapping continues the lexicons become progressively noisier and are in turn more likely to collide.

Meij and Katrenko (2007) applied BASILISK to identify biomedical entities and their patterns from full-text documents in the TREC Genomics Track QA task (Hersh et al., 2007). Their re-implementation of BASILISK utilises candidate patterns that are formed from the two tokens either side of a candidate term. This pattern simplification removes the necessity for a shallow parser. The set of patterns extracted by BASILISK were then used to find potential answer sentences for their QA system. The accuracy of their extraction process was not evaluated. However, their QA system had performance gains using these extraction patterns in unambiguous and common entity types, where little semantic drift is likely to occur.

Note that, in my re-implementation of BASILISK the patterns do not incorporate syntactic information. The BASILISK patterns used are the same for MEB (see Section 4.3.3 and 6.3).

4.3.2 NOMEN

Another multi-category bootstrapping approach called NOMEN was developed by Yangarber and is described in Lin et al. (2003b) and Yangarber (2003b). The development of NOMEN was motivated by the need for domain specific lexicons of *diseases*, *causative agents*, *transmission vectors* and *drugs*, for automatically extracting events related to infectious disease outbreaks and epidemics from the ProMED corpus (ProMED, 2003).[1]

In NOMEN, additional categories of *no* interest, referred to as *negative categories*, are introduced to boost the lexicon extraction of the categories of interest. The negative categories provide further negative examples of the target categories and as such they can improve precision substantially (Lin et al., 2003b). As stated previously, category conflicts in BASILISK are handled during the term selection phase. In NOMEN, conflicts are considered during the pattern selection phase, where patterns are penalised if they identify terms from multiple categories.

NOMEN bootstraps semantic lexicons from sets of seed terms for each semantic category of interest as well as the negative categories, and aims to extract *noun groups* for each category from corpora that are tagged with lemmas and parts of speech. Each category is seeded with the 10 most frequent related terms appearing in a large unspecified medical database. For each seed term instance in the corpus, NOMEN generates two independent extraction patterns that are formed from either the three lemmas to the left or right of the term. For example, from the following context:

$$l_{-3} \quad l_{-2} \quad l_{-1} \quad \text{< noun group >} \quad l_1 \quad l_2 \quad l_3$$

the two patterns, which independently identify the beginning and end of the seed instances, are generated:

$$l_{-3} \quad l_{-2} \quad l_{-1}$$
$$l_1 \quad l_2 \quad l_3$$

[1] ProMED is a mailing list concerned with outbreaks of infectious diseases and toxins around the world.

The extraction patterns for each category are then generalised by substituting each token from the patterns with a wildcard token. These pattern generalisations are then added to the set of candidate patterns for each category. This generalisation is performed to obtain reliable statistics as the literal patterns may not occur with sufficient frequency in the corpus.

For each semantic category, the set of noun groups that are identified by each candidate pattern p are divided into three groups: 1) *positive* (pos) terms correspond to the terms already extracted into the category's lexicon; 2) *negative* (neg) terms are terms that have been extracted into another category's lexicon, and 3) *unknown* (unk) terms that are yet to be extracted into a lexicon.[2] Based on these lists, each candidate pattern of a given category is assigned measures of *accuracy*, *confidence* and a final *score* for ranking. The accuracy of a pattern for category cat is defined as:

$$\text{accuracy}_{cat}(pattern) = \frac{|pos_{cat}|}{|pos_{cat}| + |neg_{cat}|} \tag{4.6}$$

If a pattern's accuracy is less than a pre-determined threshold θ_{prec}, it is discarded. In their experiments, Yangarber (2003b) set θ_{prec} to 0.8. For the remaining candidate patterns, the algorithm calculates their *confidence* level and then ranks them according to the *score* metric:

$$\text{confidence}(pattern_{cat}) = \frac{|pos_{cat}|}{|pos_{cat}| + |neg_{cat}| + |unk_{cat}|} \tag{4.7}$$

$$\text{score}(pattern_{cat}) = \text{confidence}(pattern_{cat}) \cdot \log(|pos_{cat}|) \tag{4.8}$$

Only patterns that identify at least two category lexicon items, and identify more positive than negative examples, receive a positive score. The top five scoring patterns for each category are added to the extracting patterns, P, and are used to identify new candidate terms.

[2]The noun groups to the left or the right of a pattern are identified using regular-expressions, such as the simple noun-group [Adj* Noun+] (Lin et al., 2003b).

The initial set of candidate terms corresponds to the sets of unk terms identified by each pattern in P. Candidate terms are discarded if they are identified by less than two patterns, and if they are not identified by both a left and a right contextual pattern. The remaining candidate terms for each category are then scored based on how many different patterns identify the term and the *confidence* of these patterns (Equation 4.9). Unlike BASILISK, which uses all possible patterns for scoring terms, in NOMEN only the top five extracting patterns associated with each individual category, P, are used to score terms. The top 5% of the scoring terms (up to a maximum of five terms) are extracted into the category's lexicon. The bootstrapping process then restarts using the lexicons for each category as the new seeds.

$$\text{score}(term) = 1 - \prod_{p \in P} (1 - \text{confidence}(p)) \tag{4.9}$$

Using the unk sets as the initial candidate terms, ensures that a term already in a category's lexicon cannot be extracted by another category. However, it does not consider the case where multiple categories attempt to extract the same term in the same iteration. In BASILISK, this case is handled in the term scoring function (see Equation 4.5).

Like Thelen and Riloff (2002), Lin et al. (2003b) also observed the usefulness of bootstrapping multiple semantic categories simultaneously. However, Lin et al. (2003b) also considered the use of *negative* semantic categories to provide more guidance in each category's search space. The negative categories are used to extract terms that belong to none of the categories of interest. In their experiments, Lin et al. (2003b) focused on extracting lexicons of *disease* and *location* names from the ProMED corpus. They first utilised one additional negative category, which was seeded with the 10 most frequent noun groups, excluding *diseases*, *locations* and those related to those categories, in the corpus.[3] This single negative category resulted in substantial improvements in precision. Their final experiment, incorporated six negative categories into the bootstrapping process: *symptom*, *animal*, *human*, *institution*, *time*, and *other*. The reason for selecting these categories was not stated. However, it is assumed that these categories were notable sources of semantic drift within the *disease*

[3]The 10 negative seeds were: *case, health, day, people, year, patient, death, number, report* and *farm.*

and *location* categories. The addition of these negative categories, significantly improved the precision of both the *disease* and *location* lexicons.

4.3.3 Mutual Exclusion Bootstrapping

Curran et al. (2007) introduced the algorithm Mutual Exclusion Bootstrapping (MEB) which aims to reduce semantic drift by extracting multiple semantic categories with individual bootstrapping instances in parallel and by forcing the categories to be mutually exclusive. MEB more actively defines the semantic boundaries of the lexicons and patterns of the categories than BASILISK and NOMEN. In MEB, conflicts between candidate terms and patterns are both considered, and the conflict resolution is based on the key assumption that terms can only belong to a single semantic category and that patterns can only extract terms for a single category. This effectively excludes general templates that can occur frequently with multiple categories and reduces the chance of assigning ambiguous terms to their less dominant sense. Curran et al. (2007) also incorporated additional negative categories, which they termed *stop categories*, to reduce the search space of the categories further.

The primary goals for developing MEB, was the necessity for a more scalable and efficient bootstrapping algorithm, that is completely language independent, for extracting large semantic lexicons from very large and noisy, unlabeled corpora, such as the Google WEB 1T corpus (Brants and Franz, 2006). Curran et al. (2007) use MEB to extract semantic lexicons from the raw WEB 1T 5-grams. The semantic categories are based on those defined in the BBN Pronoun Coreference and Entity Type Corpus: *female name*, *male name*, *last name*, *honorific title*, *facility*, *organisation*, *geo-political entity*, *location*, *date*, *language*, and NORP (*nationality, other, religion, political*) (Weischedel and Brunstein, 2005).

MEB does not require any linguistic processing, such as the shallow parsing required by AutoSlog to identify patterns in BASILISK. The set of possible candidate terms were restricted to the middle tokens of the 5-grams, and as they were extracting proper nouns, the candidate terms also needed to be capitalised. The candidate patterns were formed from the two tokens either side of the middle tokens:

t_{-2} t_{-1} < term > t_1 t_2

Even though MEB was optimised to be space efficient, significant filtering was still required to reduce the WEB 1T 5-grams to a loadable size (666MB on disk). All 5-grams which included digits or where the middle token was not capitalised were removed. Filtering also removed all patterns which only appeared with one term, since they could not identify new terms.

MEB takes as input a set of manually labelled seed terms for each semantic category. Each category's seed set forms its initial lexicon. For each term in the category lexicon, MEB extracts all candidate patterns that the term matches. To enforce mutually exclusive patterns, the candidate patterns that are identified by multiple categories are excluded from the candidate set. The remaining candidate patterns are then ranked first according to the *reliability* measure, and ties are broken using the *productivity* measure. The *reliability* of a pattern for a given category, is the number of input terms in an iteration that can identify the pattern. Thus, the maximum reliability score of a pattern is the size of the category's current lexicon. The *productivity* of a pattern is the number of non-lexicon terms it can extract and thus possibly contribute in later iterations.

The top-n patterns for each category are then used to identify new lexicon terms. Each non-lexicon term that can be extracted by at least one top pattern is identified as a candidate term. In their experiments, n was set to 5 or 10 (Curran et al., 2007).

The process of ranking and selecting new terms is symmetrical to that of the patterns. The candidate terms are first filtered using mutual exclusion — all terms that are candidates for multiple categories are excluded for the iteration. The remaining terms for each category are then ranked using the reliability and productivity measures. As only n patterns are used to identify candidate terms, the maximum reliability of a term is bounded by n. The top-m terms are added to the category's lexicon, and each category's expanded lexicon is then used as seeds for the next iteration of MEB. In the MEB experiments, m was set to 5 (Curran et al., 2007).

In MEB, terms and patterns that match the most input instances and have the potential to generate the most new candidates are preferred (Curran et al., 2007). More reliable patterns

and terms would theoretically have higher precision, while highly productive instances will have a high recall. Unfortunately, these productive instances could potentially introduce drift.

The scoring of terms and patterns in MEB is very efficient compared to both BASILISK and NOMEN. In Curran et al.'s implementation of MEB, each unique term that appears with a pattern requires only 4 bytes of storage. The terms and patterns that co-occur are completely cross-indexed, which makes updating the term and pattern extraction counts very efficient. The semantic class of each term and pattern is represented implicitly using flags, and so identifying colliding candidate terms and patterns is also extremely fast. The mutual exclusion filtering also means that the reliability and productivity measures only need to be based on a single category's counts, and thus MEB is also more memory efficient than BASILISK. In particular, in BASILISK each term is associated with floats to store its $RlogF$ values for each category, and floats to store its final $score$ values for each category (as the term may be a candidate for multiple categories).

As in Yangarber (2003a) and Lin et al. (2003b), Curran et al. (2007) also identified the need to reduce the semantic search space of categories by introducing additional competing stop categories. Certain categories of interest to Curran et al. (2007) naturally compete together, such as *geo-political entities* and *locations*, and *languages* and NORP. However, for other categories stop categories were required to help bound the topic of the competing category. These were manually selected based on an observed semantic drift in those categories (Curran et al., 2007). If a category tended to drift into another category, a stop category representing the incorrect category was created. For example, in MEB, the *jewel* stop category was added to prevent the *female name* lexicon from drifting when names such as *Ruby* and *Pearl* are extracted. In total, Curran et al. (2007) incorporated eight stop categories: *address*, *body part*, *chemical*, *colour*, *drink*, *food*, *jewel* and *web terms*.

In Curran et al.'s (2007) experiments, multi-category MEB was shown to improve on single-category MEB, and the inclusion of the stop categories resulted in further improvement. This was measured by manually evaluating the top-50 terms extracted for each of the 11 categories of interest. In Murphy and Curran (2007), MEB was used to extract bi-grams of most of their 11

original categories[4], and the additional BBN categories *event* and *law*, from the WEB 1T corpus. In the unigram and bigram experiments, different pattern context geometries were explored:

1. t_{-2} t_{-1} < unigram > t_1 t_2
2. t_{-2} t_{-1} < unigram > t_1
3. t_{-1} < unigram > t_1 t_2
4. t_{-1} < unigram > t_1
5. t_{-2} t_{-1} < bigram > t_1
6. t_{-1} < bigram > t_1 t_2
7. t_{-1} < bigram > t_1

The first pattern corresponds to the original context pattern used in Curran et al. (2007) to extract unigrams. Patterns 2–4 correspond to new unigram extracting patterns, and patterns 5–7 correspond to the bigram extracting patterns. Murphy and Curran (2007) showed that limiting the size of the left context in the patterns (Patterns 3–4, and 6–7) seriously impaired the extraction precision, while limiting the size of the right context made very little difference. MEB was also shown to be as effective in extracting bigrams.

4.4 Relation Extraction Based Bootstrapping

There are several other bootstrapping algorithms that have been developed primarily for the purpose of extracting tuples of terms for which there is a specific relationship between them. This section briefly discusses the most influential and successful relation extraction bootstrapping algorithms: DIPRE (Brin, 1998), Snowball (Agichtein and Gravano, 2000), and the large-scale approach of Paşca et al. (2006). Each of these algorithms bootstrap for a single relationship type and no multi-relationship algorithm has been devised.

[4]The individual categories for *female, male* and *last names* were merged to form one category *Person name*.

In the lexicon bootstrapping approaches discussed previously, single terms are identified as members of a semantic class. As only single terms are searched for, the extracting patterns only have one slot for a possible lexicon member. In comparison, the relation extraction methods discussed next, utilise patterns with multiple slots to identify tuples.

4.4.1 DIPRE

The *Dual Iterative Pattern Relation Expansion* (DIPRE) bootstrapping system was developed by Brin (1998) to extract tuples of *authors* and *book titles* from raw HTML pages. The DIPRE bootstrapping process is similar to the iterative process in *Mutual Bootstrapping* (MB, Riloff and Jones, 1999), whereby all relation tuples which match the identified extraction patterns are extracted without any selection scoring. Each extraction pattern generated by DIPRE is represented as a five tuple:

<*order, url-prefix, prefix, middle, suffix*>

The *order* flag determines whether the author appears before or after the title in the matching text fragment. The *url-prefix* corresponds to the URL of the document the pattern is identified in. The *prefix* corresponds to the 10 characters preceding the author (or title). The *middle* is the text between the author and title pair, and the *suffix* consists of the 10 characters after the title (or author). This five tuple will match an author-title pair if there is a document matching the *url-prefix* and contains text matching the regular expression:

prefix <author | title> *middle* <title | author> *suffix*

where both *author* and *title* strings must also match regular expressions:

Author: [A-Z][A-Za-z .,&]$^{5;30}$[A-Za-z.]
Title: [A-Z0-9][A-Za-z0-9 .,:'#!?;&]$^{4;45}$[A-Za-z0-9?!]

As each extraction pattern is associated with a *url-prefix*, each pattern can only be applied to certain web pages.

DIPRE bootstrapping begins by identifying all sentences for which both elements of a seed tuple occur. In Brin's (1998) experiments, five author and title pairs were used. Using these sentences a set of patterns are generated. For all seed mentions, candidate patterns which share the same *order* and *middle* context are clustered together and combined into one candidate pattern by identifying the longest common *url-prefix*, *prefix*, and *suffix* they share. If the resulting combined length of the *url-prefix*, *prefix*, and *suffix* is below a threshold, or the pattern matches only one seed tuple, subclusters are formed to generate more specific patterns. As each pattern is linked to a url-prefix, the extracting patterns formed are highly specific for certain web-pages. For example, the pattern:

<LI><B>*title*</B> by *author* (

is used to extract all matching tuples from the website www.sff.net/locus/c.* (Brin, 1998). As in MB (Riloff and Jones, 1999), in DIPRE all tuples that can be identified by the extracting patterns are extracted and then used as seeds in the subsequent iterations.

To evaluate DIPRE, Brin (1998) extracted *author-title* pairs from a repository of 24 million web-pages totaling 147 gigabytes. The initial five seeds, identified 199 occurrences and generated three patterns, one of which is shown above. These patterns then extracted 4047 unique *author-title* pairs. The following two iterations were not identical to the first iteration. In the second iteration, Brin (1998) restricted the document sets to a smaller sample (2 million documents), and in the third to those containing the term *books* (156,000 documents). Manual filtering of the extracted pairs was also performed to remove incorrect author names such as *Conclusion*. Over the three iterations, a total of 15,257 unique books were extracted. A random sample of these books (unknown sample size) estimated that $\approx 95\%$ were true books.

4.4.2 Snowball

Agichtein and Gravano (2000) developed the Snowball system to bootstrap tuples of *organisations* and their *locations* from newspaper articles, without HTML tags. A similar method was independently developed by Yangarber et al. (2000). Snowball extends the DIPRE system by incorporating pattern and tuple scoring functions. Before bootstrapping can begin, a named entity tagger is applied to the document set to tag mentions of *organisations*, *locations* and *persons*. Snowball utilises the MITRE Corporation's Alembic Workbench named-entity (NE) tagger (Day et al., 1997).

Patterns in Snowball, are presented as a 5-tuple:

<prefix, tag1, middle, tag2, suffix>

which contains one organisation NE and one location NE at either tag1 or tag2, and vectors associating weights with terms appearing in the prefix, suffix and middle contexts. For example:

the <organisation> *'s headquarters in* <location> *is*

This pattern representation is used for generating patterns and extracting new tuples. As in DIPRE, during the bootstrapping process, patterns are identified using a clustering method over the set of seed occurrences. An occurrence is identified if the location and organisation in a seed tuple are identified in the same sentence with the correct NE tags. Therefore, the performance of Snowball is dependent on the performance of the NE tagger. The centroids of each of the occurrence clusters are then used as the extracting patterns. No pattern ranking or selection is performed at this stage.

To identify new tuples, Snowball scans all sentences containing a *location* and *organisation* tag to identify text segments that are similar to at least one centroid pattern. If the similarity between the sentence and a centroid pattern is above a minimum similarity threshold, the *location-organisation* pair is considered a candidate tuple. Each candidate tuple is then scored using the *confidence* metric (Equation 4.10), where P is the set of patterns that generated the

tuple and C_i is the context associated with an occurrence of the tuple that matched P_i with a degree of similarity, $Match(C_i, P_i)$.

$$\text{confidence}(tuple) = 1 - \prod_{i=0}^{|P|} 1 - (\text{confidence}(P_i) \cdot \text{Match}(C_i, P_i)) \tag{4.10}$$

This metric incorporates the confidence values of the extraction patterns that identified the tuple (P_i) which is based on the *RlogF* metric in Riloff (1996a) (Equation 4.11). The sets pos and neg correspond to the correct and incorrect seed matches identified by the pattern, respectively.

$$\text{confidence}(pattern) = \frac{|pos|}{|pos| + |neg|} \cdot \log_2(|pos|) \tag{4.11}$$

Using these metrics, Snowball discards all candidate tuples with low confidence. The remaining tuples are added to the list of extracted relations, and the bootstrapping process repeats using the entire list of relations as seed tuples.

Agichtein and Gravano (2000) compared Snowball to their own re-implementation of DIPRE on a large collection of newspapers from the North American News Text Corpus (LDC, 1995). The systems were seeded with five *organisation-location* tuples. The confidence thresholds in Snowball for selecting candidate tuples after each iteration resulted in slightly higher precision in comparison to DIPRE after the first iteration. DIPRE does not have a mechanism for rejecting matching tuples and so suffers from semantic drift in the following iterations. In Snowball, the confidence threshold can be varied in each iteration depending on the precision or recall requirements. The performance of Snowball was evaluated on 100 randomly selected tuples, and the majority of the errors (47/48 with no confidence threshold; 7/7 with confidence set to 0.8) were related to the performance of the named entity tagger. Therefore, bootstrapping systems which process raw text and do not depend on the accuracy of NLP tools are favourable, especially for domains where the tools are significantly less accurate than those for newswire.

Yu and Agichtein (2003) applied Snowball to extract synonymous relationships between *gene* and *protein* terms from 52,000 biomedical full-text articles.[5] Their approach also required the prior tagging of the semantic categories of interest in the text before bootstrapping could begin. Snowball was seeded with a large set of 650 known gene and protein synonym tuples and only two bootstrapping iterations were used. A confidence threshold of 0.8 was used to obtain a precision of 90.0% over an unreported sample size.

4.4.3 Large-scale Fact Extraction

Paşca et al. (2006) present a large-scale bootstrapping algorithm inspired by DIPRE and Snowball, for extracting facts regarding people and their year of births (Person-BornIn-Year) from the Web. Their main focus is on fast-growth extraction for very high scalability compared to the more conservative approaches such as BASILISK and MEB. Using 10 seed facts and 100 million web documents, their algorithm expands the list to 100,000 facts and then to 1,000,000 facts, after the first and second iteration, respectively (Paşca et al., 2006). They demonstrate that it is possible to generate one million facts with a precision of approximately 88%.

For each seed tuple, a set of initial candidate patterns is identified by first matching the tuple's pair of terms within sentences. Each pattern is represented by the tuple:

<prefix, term1, middle, term2, suffix>

Each fact tuple mention is then assigned a pattern that first only consists of the sequence of terms between the pairs of seed terms (*middle*, up to length 6). Candidate patterns composed of only stop-words are discarded. From the set of remaining candidate patterns, a set of generalised patterns are acquired, and both sets of patterns are used for identifying new candidate facts. Paşca et al. (2006) incorporates Lin's (1998a) distributional similarity tool to generalise the candidate patterns. The tool identifies similar terms to those within the middle sections of the set of patterns. These similar terms are then used to generalise the middle section and induce new patterns. For example, in the pattern:

[5] These articles are not publicly available and are maintained by the GeneWays project (Friedman et al., 2001).

<prefix> <person> *was born in June ,* <year> <suffix>

which matches the text:

<prefix> Chester$_{\text{NNP}}$ Burton$_{\text{NNP}}$ Atkins$_{\text{NNP}}$ *was born in June ,* 1924$_{\text{CD}}$ <suffix>

the term *June* would be generalised using a set of similar terms identified by Lin's distributional similarity approach, such as *April, Aug., February, Jul, October*, etc. This allows the automatic induction of new patterns, such as:

<prefix> <person> *was born in April ,* <year> <suffix>
<prefix> <person> *was born in Aug. ,* <year> <suffix>

Due to the aggressive expansion of candidate patterns, it is impossible to score and rank them based on the 10 seed facts, as is done in smaller scale algorithms. In large-scale extraction, candidate patterns are ranked highly if they contain words that are indicative of the facts being extracted (Paşca et al., 2006). For example, patterns containing the word *birthday* are also likely to extract correct candidate Person-BornIn-Year facts, whereas patterns containing the term *graduated* are not. Terms that are related to the specific fact of interest are identified as those occurring most frequently in the set of candidate patterns. It is assumed that all patterns containing fact related terms are retained.

At this stage, the candidate patterns' *prefix* and *suffix* parts have not been defined. Each extracting pattern is then associated with a sequence of consecutive POS tags to its left and right, which bound the extremities of the associated seed fact. These form the *prefix* and *suffix* parts, respectively. Candidate fact tuples are identified in sentences if they match a complete pattern. That is, if the *middle* context matches, and the *prefix* and *suffix* POS tag sequences of the pattern also match, than the tuple is extracted. Each pattern can acquire up to 600,000 candidate fact tuples.

As with the pattern ranking, it is difficult to rank candidate facts based on the extracting patterns. Paşca et al. (2006) rank candidate facts based on their accumulated distributional

similarity scores to each of the seed facts. If a term within an extracted tuple is not similar to any of the seed fact terms, the candidate fact is discarded. For example, Paşca et al.'s system eliminates the correct fact *Jethro Tull* born in *1947*, as *Tull* was not similar to any of the surnames in the seed facts. The top $1/3$ of the remaining candidate facts are added to the list of facts. The expanded fact list is then used as seeds in the second/final iteration.

Of the 1 million facts extracted, a sample of 1299 facts were evaluated by Paşca et al. (2006).[6] This sample contained 88.3% of correct facts, which was then extrapolated to be the average precision over the entire 1 million set (Paşca et al., 2006). These results demonstrate that with only a set of 10 seed Person-BornIn-Year facts, it is possible to generate 1 million facts with high precision. However, this result is not too surprising. Given the expected enormous set of sentences mentioning a Person-BornIn-Year fact on the web and their limited use of language for expressing these facts, and the rigid extracting patterns containing fact specific terminology and strict POS boundaries, it is expected that many correct facts will be identified with the initial set of patterns.

The portability of Paşca et al.'s (2006) approach is also questionable. It is not clear to what extent fast growth is applicable to other facts or different domains, especially those which are less numerous on the web. In fact, Paşca et al. (2006) state that they chose this specific fact due to the known many mentions of these on the web. Indeed as there are far fewer companies and countries than people in the world, it is expected there would be significantly fewer mentions of Person-CeoOf-Company or City-CapitalOf-Country facts available to extract. Within the biomedical domain, rarer relationships and entities from smaller document collections are of critical importance.

[6] This sample is generated from the top of the extracted list of facts, by retaining a fact and skipping the next N facts, where N is incremented at each step. This retains 1414 facts, however 115 of those facts were not associated with any documents after a web query and were thus discarded (Paşca et al., 2006).

4.5 Summary

This chapter presented an overview of the existing minimally supervised bootstrapping methods described in the literature, which are related to this thesis. The canonical work of Hearst and Riloff have motivated the development of these bootstrapping algorithms for automatically extracting semantic lexicons and their patterns (or relationships between entities), from a small seed sample of domain knowledge.

As discussed in this chapter, semantic drift of the lexicons, which occurs when polysemous terms and/or extracting patterns are identified during the bootstrapping process, is a major concern and prevents the extraction of large yet precise lexicons. The early algorithms described by Riloff and Shepherd (1997) and Riloff and Jones (1999) bootstrapped lexicons for single semantic categories, and aimed to improve precision by incorporating new term and/or pattern scoring and selection methods. However, as the literature demonstrates these single-category methods are still affected by semantic drift.

In Section 4.3, I introduced the multi-category bootstrapping algorithms, BASILISK, NOMEN and MEB, which outperform the single-category frameworks. These approaches reduce semantic drift further by utilising information about other semantic categories to prevent the categories drifting into each other's semantic space. In conjunction with their different term and pattern scoring metrics, each algorithm differs in their use of patterns and conflict resolution: BASILISK utilises fixed patterns and considers category conflicts between candidate terms; NOMEN applies pattern generalisations and considers conflicts between candidate patterns; and MEB uses fixed token-based patterns and forces mutual exclusion between terms and between patterns. The experiments in the following chapters show that both BASILISK and MEB are still susceptible to semantic drift. This motivates the development of a new multi-category bootstrapping algorithm (*Weighted Mutual Exclusion Bootstrapping*, WMEB), which is presented in Chapter 6.

To conclude the literature review, the closely related and influential work of bootstrapping relationships between two semantic categories from text is discussed. For example, the Snowball system by Agichtein and Gravano (2000) has been successfully used to extract relationships between genes and proteins (Yu and Agichtein, 2003). The possibility of bootstrapping multiple relationships simultaneously is an open question, to be addressed in future work.

Chapter 5

Evaluation Methodology

This thesis focuses on extracting biomedical semantic lexicons and their patterns from raw text. Section 5.1 describes in detail the ten biomedical semantic categories used to compare the performance of the bootstrapping algorithms. For each category, examples of correct terms and common incorrectly extracted terms are provided. Section 5.2 discusses briefly the stop categories used in the bootstrapping experiments in this thesis. This is followed by a description of the raw biomedical text the bootstrappers extract the biomedical lexicons from.

One of the main obstacles to developing minimally supervised bootstrapping systems is evaluating the quality of the lexicons extracted from raw text. Section 5.4 details the limitations of the available biomedical resources which render them unsuitable for evaluating the extracted biomedical lexicons. I describe the manual evaluation methodology and, in Section 5.5 discuss the reliability of the manual evaluation and present the kappa statistics for evaluator agreements. The extracted lexicons are evaluated in terms of *precision*, *inverse rank score*, and degree of *overlap*. These measures are discussed in Section 5.6. This chapter concludes by presenting the evaluation guidelines for manually judging the quality of the patterns extracted during bootstrapping.

Category	Hand picked seeds
ANTIBODY	*MAb IgG IgM rituximab infliximab*
CELL	*RBC HUVEC BAEC VSMC SMC*
CELL LINE	*PC12 CHO HeLa Jurkat COS*
DISEASE	*asthma hepatitis tuberculosis HIV malaria*
DRUG	*acetylcholine carbachol heparin penicillin tetracyclin*
FUNCTION	*kinase ligase acetyltransferase helicase binding*
MUTATION	*C677T C282Y 35delG Leiden nul* l (MEDLINE)
	T47D K44A F442A G93A null (TREC)
PROTEIN	*p53 actin collagen albumin IL-6*
SYMPTOM	*anemia hypertension hyperglycemia fever cough*
TUMOUR	*lymphoma sarcoma melanoma neuroblastoma*

Table 5.1: Hand-picked seeds for each biomedical semantic category

5.1 Biomedical Semantic Categories

In this thesis, I evaluate and compare the effectiveness of various bootstrapping algorithms for extracting single-term lexicons of ten biomedical semantic categories: ANTIBODY, CELL, CELL LINE, DISEASE, DRUG, FUNCTION (functions and processes), MUTATION, PROTEIN (proteins and genes), SYMPTOM (signs and symptoms) and TUMOUR. The categories were inspired by the TREC Genomics 2007 named entities, with some modifications to those descriptions (Hersh et al., 2007). The TREC categories *Genes* and *Proteins* are combined into one category, PROTEIN, as there is a very high degree of metonymy between these, particularly once out of context. The TREC category *Strains* was excluded due to the difficulty for biologists to distinguish between strains and organisms. The categories *Toxicities*, *Pathways* and *Biological Substances*, which are predominantly multi-term entities, were also excluded because only single-term lexicons are extracted. However, single term entities from these categories may be correctly extracted by the other semantic categories, such as FUNCTION. I was also interested in the fine grain distinction between types of *cells* and *cell lines*, so I split the *Cell or Tissue Type* category into CELL and CELL LINE entities. The hand-picked seeds for each semantic category are shown in Table 5.1. Separate MUTATION seeds for the MEDLINE and TREC Genomics experiments were used, as some of the original seeds used for MEDLINE do not appear in the TREC Genomics document set.

This section describes each of the ten biomedical semantic categories in detail, including correct and incorrect examples. The general evaluation guidelines and inter-evaluator agreement follow in Section 5.4 and 5.5, respectively.

5.1.1 CELL

The simplest self-stabilising unit in an organism is the cell. It must integrate the activity of its components to form a highly specialised functional entity and respond to multiple signals in a robust manner. Multicellular organisms (organisms with multiple cells) contain a vast array of highly specialized cells. For example, humans have neurons and rod cells in the retina of the eye, and plants contain Sclerenchyma cells which provide structural support. This category includes all mentions of distinct morphological or functional forms of cells. It does not include cell lines, locations where cells are taken from or general modifiers of cells. Examples:

Correct terms

- *erythrocytes*, *fibroblasts*, *lymphocytes*, *mast*, *neutrophils*, *T-cell*, *CD4+*
- *RBCs* is an abbreviation for *Red Blood Cells*.
- *HBFC* is an abbreviation for *human bronchial fibroblastic cells*.
- *MNC(CD34+)* is a mononuclear cell marked with CD34 surface molecules.

Incorrect terms

- locations of cells: *glial*, *epithelial*, *mouse*, *liver* and *neuroblastoma*
- *BVMC* is non-specific and an abbreviation for *bacteria, viruses, and mammalian cells*.
- modifiers: *cultured, infected, lysed, modified*, and *leukemic*

5.1.2 CELL LINE

A cell line is a population of cells propagated in culture that are totally derived from a single common ancestor cell, typically from cancerous tissue. Cell lines are valuable for research as

they can be reproduced indefinitely and are genetically identical. This category includes all mentions of cell lines. It does not include cell types or cancers, or the organism or location, the cell lines are derived from. Examples:

Correct terms

- *PC12, CHO, HeLa, Jurkat*
- *COS-7* and *COS*. *COS-7* is a cell line, which is often referred to as *COS*.

Incorrect terms

- *mouse*, as in '*the mouse cell line*', refers to the organism the cell line was extracted from.
- *NHL*, as in '*NHL cell lines*', refers to cell lines derived from *Non-hodgkins Lymphoma* cells.

5.1.3 PROTEIN

Cells are comprised of DNA, RNA and protein molecules. DNA molecules are composed of specific sequences of nucleic acids known as genes, which in turn specify the structure and composition of proteins. Gene expression refers to the process of transcribing a gene's DNA sequence into a messenger RNA (mRNA) transcript which serves as a template for protein synthesis. During protein synthesis the resulting mRNA is translated into chains of amino acids, which are then transformed into a three dimensional functional protein.

Automatically extracting gene, RNA and protein names in text is an open research area. In particular, the correct classification of such terms as either genes, RNA or proteins is a difficult task, in which human experts only agree on $\sim 78\%$ of the time (Hatzivassiloglou et al., 2001). This is predominantly due to terminological economy, which leads to a large degree of metonymy between these entities — every protein has an associated gene and RNA transcripts, which often share the same name. Once these entities are taken out of context, any disambiguation is almost always impossible. Therefore, since we evaluate out of context this category includes terms from the two TREC Genomics categories, Genes, and Proteins, and

terms referring to RNA molecules. It also includes classes of proteins, protein domains and complexes. Examples:

Correct terms

- *p53, actin, collagen, albumin, IL-6*
- *kinase*, *reductase* and *enzyme* are types of proteins.
- *E6:P53* is the complex formed between *E6* and *P53*.
- *p21/p53* is the complex formed between *p21* and *p53*.
- *SH2* is a protein domain (*Src homology 2 domain*).

Incorrect terms

- Single characters or digits, which may be part of multi-term protein/gene names, but cannot be used on their own to represent the protein/gene.

  S in *Protein S*

  D in *Glycophorin D*

- Single amino acids or DNA bases are excluded.

5.1.4 MUTATION

Genes and their proteins are essential components of all cells. They are responsible for many vital functions and processes that must occur without errors for a cell to perform correctly. Mutations, which can occur within a gene or protein sequence, can result in their functions being modified. Other sequence mutations can affect if and when proteins are formed. Mutation types specify the kind of genetic change that occurred. For example, a *null* mutation of a gene eradicates the function of the gene's protein, and a point mutation indicates that a base pair in the gene's sequence was substituted by another.

Point mutations are often mentioned using a common notation: *wNm*, where *w* and *m* correspond to the original (wild-type) and mutant nucleic or amino acids, respectively, and *N*

refers to the sequence position of the substitution. For example, the mutation *G1691A* states that the nucleic acid *Guanine* at position 1691 was mutated to *Adenine*.[1] In both the MutationFinder (Caporaso et al., 2007) and mSTRAP (Kanagasabai et al., 2007) systems, regular expressions are used to automatically extract point mutations in this format. However, as noted by Caporaso et al. (2007), this notation can also identify many other entities, such as the cell line *T98G*.

Many mutations also have generic names. For example, the mutation *G1691A* is also commonly referred to as the *Leiden* mutation of the *Factor V* gene, and the mutation *Ala242Val* in mice is also known as *jimpy*. Mutations in an organism (mutants) which result in observable characteristics are often referred to by their mutant phenotype names. For example, the *scid* (severe combined immune deficiency) phenotype is due to a mutation resulting in the failed development of lymphocytes (a type of cell), and the *nude* phenotype results in a genetic mutation in mice causing an inhibited immune system and a lack of body hair.

This category includes all mentions of gene and protein mutations, chromosomal mutations and mutants. It does not include names of the mutated genes, proteins or organisms. Examples:

Correct terms

1. Point mutations: *wNm*
    - *A200C* is an amino acid mutation at residue 200 where *Alanine* is mutated to *Cysteine*.
    - *Ala200Cys*, *Ala200C*, *A200Cys* are accepted synonyms, where the last two are uncommon and likely due to poor editing.
    - *Asn-45Ser*
2. Amino acid deletions: the Greek letter $\triangle$ or *delta* (*del* for short) often indicates a deletion of an amino acid.
    - $\triangle$*F508* is a deletion mutation where *Phenylalanine* (*F*) in the wild-type at position *508* is removed.

[1] The mutation G1691A can also be represented by the terms: 1691G>A, Arg534Gln, and R506Q

- *999delAla* is a deletion mutation where *Alanine* (*Ala*) in the wild-type at position *999* is removed.
- *deltaR208* is a deletion mutation where *Arginine* (*R*) in the wild-type at position *208* is removed.
- *4710delAG* is a deletion mutation where two nucleic bases (*Adenine* (*A*) and *Guanine* (*G*)) are removed from position 4710.

3. Nonsense mutations: refer to mutations resulting from a point mutation in the DNA sequence that results in a premature *stop codon*, indicating the end of the protein sequence.[2] This often results in a *truncated* and nonfunctional protein. *wNSTOP* is a common pattern specifying the mutation of a residue *w* at position *N* to a stop codon.

   - *Cys456STOP* and *C456stop* is a truncated mutation where *Cysteine* at residue position *456* is replaced with a stop codon.

4. Insertion mutations: refer to mutations that arise from the addition of one or more nucleic acids into a genetic sequence, which often result in frameshift mutations.

   - *87insC* corresponds to the nucleic acid *Cytosine* (*C*) inserted at position 87.
   - *1350insG* corresponds to the nucleic acid *Guanine* (*G*) inserted at position 1350, and this results in the frameshift mutation *V451GfsX453*.

5. Chromosomal translocations: The International System for Human Cytogenetic Nomenclature (ISCN), which has been revised numerous times since 1963 (Shaffer et al., 2009), is used to describe translocations between chromosomes. Translocations arise when parts of two different chromosomes are exchanged. The notation *t(A;B)(pN;qN)* is used to denote a translocation between chromosome *A* and chromosome *B*. The second set of parentheses, if given, specifies the precise location within the chromosomes *A* and *B*, respectively, with *p* indicating the short arm of the chromosome, *q* indicating the long

[2] A stop codon is a set of three consecutive nucleic acids in the RNA or DNA sequence which signal the end of the gene/protein sequence. There are three standard stop codons: UAG (in RNA) / TAG (in DNA), UAA/TAA, and UGA/TGA.

arm, and the numbers after p or q refers to chromosome regions. Note there are slight variations to this notation in the literature.

- *t(2;5)(p23;q35)* – causes *anaplastic large cell lymphoma*
- *t(X;18)(p11.2;q11.2)* – causes *Synovial sarcoma*
- *t(15;17)* – causes *acute promyelocytic leukemia*
- *t(14;21q)* – causes *Down Syndrome*
- *t(1;11)(q42.1;q14.3)* – causes *Schizophrenia*

6. Null and knock-out mutations: a *null* mutation eradicates the function of a gene/protein, and a knock-out mutation refers to the direct disruption of a single gene within an organism for research purposes. Common representations include *gene(-/-)*.

   - *TLR4(-/-)*, *eNOS(-/-)*, *apoE(-/-)*
   - *AhRKO* is a knock-out (KO) of *AhR*.

7. Names designated to mutations and mutant organisms

   - *whirler* – *whirler* (*wi*) is a mouse mutation on chromosome 4 which results in neuroepithelial deafness.
   - *clock* – mouse *clock* mutants have a mutation which changes the overall circadian clock mechanism.
   - *orange-eyed* – the *Drosophila melanogaster* (fruit flies) have a mutation in the *white* gene, which normally produces the red pigments in the eye.
   - *ebony* – these *Drosophila melanogaster* have a dark and almost black body. They carry a mutation in the *ebony* gene which is responsible for the tan-coloured pigments in wild-type flies. If the *ebony* gene is defective, the *ebony* fly results.

Incorrect terms

1. Do not include the names of the mutated genes or proteins.

2. Do not include references to a single amino acid.

    - *Trp1506* refers to the amino acid Tryptophan of a peptide/protein at position 1506.
    - *F54* refers to the amino acid Phenylalanine at position 54.

5.1.5 ANTIBODY

Antibodies are a class of immunological proteins (immunoglobulins) produced by *B-cells*, which identify and bind with a specific molecule (antigen). In the body, antibodies are used to identify and neutralize foreign objects, such as bacteria and viruses. For example, the naturally occurring antibody *IgG*, is used to neutralize toxins and prevent infections by blocking bacterial and viral entry into cells. Externally produced antibodies can also be used in the treatment of disease. For example, *Infliximab* is used to bind and block the action of the protein *TNF* which increases inflammation in *rheumatoid arthritis*.

Antibodies developed to specifically bind to different cellular molecules are also commonly used in molecular research applications. Antibodies, labelled with fluorescent dyes, which specifically bind surface proteins of a specific cell type, can be used in flow cytometry to differentiate cell types (BD Biosciences, 2002). For example, the antibody *2H7* recognises the *CD20* antigen present on *human pre B lymphocytes* and *B lymphocytes* at all stages of maturation, except on plasma cells (Liu et al., 1987).

This category includes antibody types, naturally occurring antibodies and those used in disease diagnosis, disease and prenatal therapy, and research applications. Examples:

Correct terms

1. Naturally occurring antibodies

    - *IgA, IgG, IgM*

2. Antibodies used as treatments, including generic and brand names.

    - *rituximab, infliximab, lintuzumab, galiximab, adalimumab*
    - *CAMPATH-1* is a brand name for the antibody *Alemtuzumab*

3. Experimental antibodies

    - *5C6* is an *anti-FAM monoclonal antibody*.
    - *mH18A* is an antibody which specifically binds *anti-Lewis Y*.
    - *Ab4* is an antibody which specifically binds *anti-HLA-DR*.

4. Abbreviations

    - *MAb* is an abbreviation for *Monoclonal antibodies*.
    - *HAb* is an abbreviation for *heparin-associated antibodies*.
    - *HBeAb* is an abbreviation for *hepatitis B e antibody*.
    - *cANCA* is an abbreviation for *Antineutrophil cytoplasmic antibody*.
    - *IA-2-Ab* is an abbreviation for *IA-2 antibody*.
    - *aCLAs* is an abbreviation for *anticardiolipin antibodies*.

5. Conjugations of antibody terms

    - *BCP7-10* corresponds to four antibodies – *BCP7*, *BCP8*, *BCP9* and *BCP10*.
    - *VAK-3-5* corresponds to three antibodies – *VAK-1*, *VAK-2* and *VAK-3*.

Incorrect terms

1. Do not include references to the antigens which form part of multi-term antibodies.

    - *MAP* in *the MAP antibody*, refers to the antigen the antibody binds.

2. Do not include references of being positive or negative for a specific antibody.

    - *HAb+* refers to a cell that is *positive* for *heparin-associated antibodies*.

5.1.6 DISEASE

A disease is a definite pathologic process with a characteristic set of signs and symptoms that affects humans, animals, and/or plants. The category also includes disabilities, disorders, syndromes, deviant behaviors, infections, and viruses causing infectious diseases. Examples:

Correct terms

1. Common diseases
    - *asthma, hepatitis, tuberculosis, HIV, malaria*
2. Terms that are part of embedded disease entities, such as part of a long form ending in *disease*, are correct if the term can unambiguously refer to the same disease.
    - *Creutzfeldt-Jakob* in *Creutzfeldt-Jakob Disease*
    - *Parkinson's* in *Parkinson's Disease*
    - *Huntington's* in *Huntington's Disease*
    - *mad* or *cow* as in *mad cow disease* are too generic and are thus incorrect.
3. Abbreviations and acronyms
    - *CHD* is an acronym of *Coronary heart disease*.
    - *CVD* is an abbreviation for *Cardiovascular disease*.
4. Nouns referring to people, animals or plants, suffering from diseases
    - *diabetic*, *hypoxemic*, *anaemic*
5. Viruses known to cause infections in humans, animals or plants.
    - *ADV* is an abbreviation for *Aleutian Disease Virus*, and causes Aleutian Disease.
    - *AHFV* is an acronym of *Alkhurma hemorrhagic fever virus*.
    - *TSHV* is an abbreviation for *Tree Squirrel Hepatitis Virus*.
    - *TYLCV* is an acronym for *Tomato yellow leaf curl virus*, which is a virus infecting tomatoes.

- *Flavivirus* (also known as *viral hemorrhagic fever virus* (*AHFV*) is the agent of a viral hemorrhagic fever. Both *Flavivirus* and *AHFV* are correct.

6. Pathogens, only if the disease caused by the pathogen shares the same name.

   - The genus *Chlamydia* is correct as it causes *Chlamydia Infection* which can be referred to as the single term entity *Chlamydia*.
   - The genus *Aspergillus* is *incorrect* as its pathogenic form *Aspergillus fumigatus* causes the disease *Aspergillosis*, which is not referred to as *Aspergillus*.

Incorrect terms

1. Do not include viruses which are not known to cause diseases.

   - *AAV* is an abbreviation for *Adeno-associated virus*, which is not currently known to cause disease.
   - *R6004* is a vacinna virus.
   - *vSIDK* is a recombinant vaccinia virus.

2. Do not include viruses conjugated with other entities.

   - *MHV-3CLpro* refers to the *3C-like proteinase (3CLpro)* of *mouse hepatitis virus (MHV)*.
   - *HIV-1LTR* refers to the transactivated *LTR gene* of *HIV-1*.

3. Do not include adjectival modifiers and those pertaining to disease locations.

   - *recurrent, chronic, persistent, infectious, rare*
   - *respiratory, gastrointestinal, liver, plant, leaf*

5.1.7 TUMOUR

A tumour or tumor is the name of an abnormal growth of cells, originating from a specific tissue of origin or cell type, and having defined characteristic properties, such as recognized histology.[3] Tumors can be benign, pre-malignant or malignant, and only malignant tumours are cancerous. This semantic category does not include the names of cell lines that are derived from tumour cells. Examples:

Correct terms

1. Single term tumours
    - *lymphoma*, *melanoma*, *neuroblastoma*, *angioreticuloma*, *osteosarcoma*
2. Abbreviations and acronyms
    - *SCLC* is an acronym for *Small cell lung carcinoma*.
    - *DLBCL* is an acronym for *Diffuse large B-cell lymphoma*.
    - *SLCT* is an acronym for *Sertoli-Leydig cell tumors*.
    - *NSGCTT* is an abbreviation for *nonseminomatous germ cell tumors of the testis*.

Incorrect terms

1. Do not include adjectival modifiers and terms referring to tumour locations.
    - *recurrent, chronic, rare*
    - *brain, thyroid, penial, pituitary, trophoblastic, squamous*
2. Do not include abbreviations or acronyms that include the term tumour but do not specifically refer to tumours.
    - *GAT* is an abbreviation for *galactosyltransferase associated with tumor*.
    - *otu* is an abbreviation for the *ovarian tumor locus*.

[3] Histology is the study of the microscopic anatomy of cells and tissues.

5.1.8 SYMPTOM

Any objective indications of some medical fact that are observed or measured by a physician during an examination, such as *hypertension*, are known as *Signs*. *Symptoms* on the other hand include the sensations or subjective changes in health function that are experienced by patients, such as *pain*. This category includes both signs and symptoms. Examples:

Correct terms

1. Single terms referring to signs and symptoms.
    - *aggression, anemia, bellyache, cough, diarrhoea, fever, unsteadiness*
    - *hypertension, hypercholesterolemia, hypoglycemia, hypoventilation*
2. Conditions and diseases that are also signs and symptoms.
    - *anemia, insomnia, infections*
    - *Polycythemia* – an increase in blood volume as a result of an increase in the number of erythrocytes. It may result from a blood-forming disease that increases cell production, or it may be a physiologic response to an increased need for oxygenation in high altitudes, cardiac disease, or respiratory disorders.

5.1.9 DRUG

A pharmaceutical preparation intended for human or veterinary use, including discontinued drugs and those in clinical trials. Includes chemical, generic and brand names. Examples:

Correct terms

1. Common medications
    - *acetylcholine, carbachol, heparin, penicillin, morphine, codeine*
    - *detomidine* is used as a large animal sedative.

2. Antibodies used as treatments

   - *Infliximab* (brand name *Remicade*) is used to treat autoimmune disorders.
   - *Trastuzumab* (brand name *Herceptin*) is used in breast cancer therapy.

3. Chemical name variations

   - *d,1-amphetamine*
   - *(-)-noradrenaline*

4. Obvious parts of multi-term drug names.

   - *4'-ethyl-2-methyl-3-piperidinopropiophenone* is part of
     4'-ethyl-2-methyl-3-piperidinopropiophenone hydrochloride
     Generic name: *Eperisone hydrochloride*. Brand name: *Myonal*.

5. Abbreviations and acronyms

   - *CQ* is an abbreviation for *chloroquine*.
   - *SP* is an abbreviation for the combination of *sulphadoxine* and *pyrimethamine*.
   - *INPEA* is an abbreviation for the beta-adrenergic blocking drug
     1-p-nitrophenyl-2-isopropylaminoethanol.

Incorrect terms

1. Do not include chemicals used only in scientific experiments.

   - *5-aza-2'-deoxycytidine* (*5-azadC*) is a widely used potent inhibitor of the protein DNA *methyltransferase*.

5.1.10 FUNCTION

This semantic category broadly covers all activities describing the actions of functional bio-entities, such as antibodies, proteins, drugs, or bioactive substances, and the pathways and processes formed by consecutive activities. This category includes entities from the *Molecular Functions* and *Pathways* entities from the TREC Genomics task (Hersh et al., 2007). Examples:

Correct terms

1. Single term molecular functions
    - *kinase, ligase, helicase, bind*
2. Molecular pathways and processes
    - *apoptosis, dephosphorylation, replication, transcription, ubiquitination*
3. Nouns describing the functional categories of bio-entities
    - *adrenoreceptor, acetyltransferase, catalyst, pesticidal, steroid*
    - *antiinflammatory, antispasmodic, bronchoconstrictor, vasocontrictor*
4. Lexeme variations
    - *bound, bind, binding, binds*
    - *activate, activated, activates, activation*

Incorrect terms

1. Do not include specific bio-entities.
    - *nucleotidase* is a functional category, however *ecto-5'-nucleotidase* is a specific type of *nucleotidase* protein and is thus incorrect.

Category	Seed terms
AMINO ACID	*arginine cysteine glycine glutamate histamine*
ANIMAL	*insect mammal mice mouse rats*
BODY PART	*breast eye liver muscle spleen*
ORGANISM	*Bartonella Borrelia Cryptosporidium Salmonella toxoplasma*

Table 5.2: Hand-picked seeds for each biomedical stop category

5.2 Stop Categories

Following Yangarber (2003a) and Curran et al. (2007), I also introduce additional semantic categories referred to as *stop categories* to help reduce the semantic search space and thus semantic drift of the categories of interest. I used four stop categories: AMINO ACID, ANIMAL, BODY PART, and ORGANISM (collectively STOP), and the hand-picked seeds for each of these are shown in Table 5.2. These STOP categories were identified as common sources of semantic drift in preliminary experiments with MEB on the biomedical categories. This is discussed further in Chapter 6.

5.3 Biomedical Corpora

The corpora used in this thesis consist of MEDLINE abstracts and the TREC Genomics 2007 full-text articles. These corpora are significantly larger than any annotated biomedical corpora, and cover a broad range of biological topics. I used both corpora to compare and evaluate the bootstrapping methods in the following chapters.

The MEDLINE database, from the National Library of Medicine (NLM), is the primary literature collection in the medicine and life science domains. The database currently contains more than 17 million references to articles published in 5000 scientific journals. This set includes 9 million article abstracts, and article titles for the remaining references, all of which are freely available. The MEDLINE database is expanding rapidly at a double-exponential rate. In 2007, 670,000 additional references were added (NLM, 2009).

The MEDLINE document set used in the experiments consists of all titles and abstracts of all research documents indexed by MEDLINE up to and including October 2007 (16 140 000

Type	MEDLINE	TREC
No. Terms	1 347 002	1 478 119
No. Contexts	4 090 412	8 720 839
No. 5-grams	72 796 760	63 425 523
No. Unfiltered tokens	6 642 802 776	3 479 412 905
No. Documents	16 140 000	162 259

Table 5.3: MEDLINE and TREC statistics

documents). This set also includes references to articles for which only the title is available. The XML markup added by MEDLINE was removed and the documents were converted into raw text. Using bio-specific NLP tools (Grover et al., 2006), with methods to detect chemical names, these documents were then tokenised and split into sentences.

The growth rate of available biomedical literature is also increasing due to the inclusion of Open Access full-text articles in the NLM's PubMed Central database. In the full-text experiments, the TREC Genomics collection is used, which contains some Open Access articles, and articles made available through their publishers specifically for the task (Hersh et al., 2007).

The TREC collection contains 162 259 full-text biomedical documents in HTML format from 49 journals. The full collection is approximately 12.3 GB when uncompressed. After removing the HTML markup, these documents were processed the same way as the MEDLINE set. No further text processing, such as part-of-speech tagging or parsing, was applied.

5.3.1 Term and Pattern Sets

The bootstrapping algorithms implemented in this thesis take as input a set of candidate terms to be extracted into semantic lexicons, and the candidate contextual patterns the algorithms may use to identify new lexical items. Unlike Riloff and Jones (1999) and Yangarber (2003a), I do not incorporate any syntactic knowledge within the patterns or terms, as I am deliberately taking a purely language and domain independent approach. For the experiments, each corpus is converted into a set of 5-grams (t_1, t_2, t_3, t_4, t_5). The 5-grams do not cross sentence boundaries. From these 5-grams, the set of possible candidate terms and contextual patterns are identified. The set of terms corresponds to all of the middle tokens (t_3) of the 5-grams, and the patterns are formed from the tuple of the surrounding tokens (t_1, t_2, t_4, t_5).

Filtering was necessary to reduce the number of patterns and terms to a loadable size for BASILISK (Thelen and Riloff, 2002), the most memory expensive bootstrapping algorithm used in the experiments. The 5-grams in each corpus were filtered to remove patterns appearing less than 7 (MEDLINE) or 3 times (TREC), and these cut-offs were set empirically to permit the largest number of terms and patterns loadable by BASILISK in 4GB of RAM. Candidate terms consisting of only non-alphanumeric tokens, such as punctuation, were also removed.

The statistics of the resulting data-sets are shown in Table 5.3. Before filtering, the MEDLINE corpus is twice as large as the TREC corpus. After filtering, the number of possible terms that can be extracted from either MEDLINE or TREC is about 1.5 million. Note that for a similar number of terms, the TREC corpus results in more than double the number of patterns than the MEDLINE corpus. This makes these data-sets considerably different for bootstrapping.

5.4 Manual Evaluation of Lexicons

In recent years, the increasing interest in Information Extraction within the biomedical domain has prompted the development of numerous domain specific challenges, such as the BioCreative tasks (Hirschman et al., 2005; Wilbur et al., 2007), and additional annotated corpora, such as GENIA (Ohta et al., 2002; Kim et al., 2008) and BioInfer (Pyysalo et al., 2007). These resources have enabled significant progress in the development and automatic evaluation of domain specific NLP tools, including named entity recognition systems (Ando, 2007) and parsers (Pyysalo, 2008). As gold-standards, they also allow fair comparisons between different approaches for specific NLP tasks, using the standard measures of *precision*, *recall* and *F-score*. In terms of evaluating semantic lexicons, precision is the percentage of correct lexicon terms that have been extracted against the total number of extracted terms, and is the main measure used. Recall is the percentage of correctly extracted lexicon terms against the total number of correct terms in a gold-standard set. F-score is the harmonic mean of precision and recall.

Unfortunately, the limited coverage of gold-standard biomedical corpora and their associated semantic lexicons make them unsuitable for facilitating the automatic evaluation of

extracted lexicons. Many corpora are limited in size and are biased towards specific topics within biomedicine. For instance, the GENIA corpus is limited to 1999 MEDLINE abstracts containing three MeSH terms, *Human*, *Blood Cells*, and *Transcription Factors* (Ohta et al., 2002; Kim et al., 2008). As a result, the gold-standard corpora do not accurately represent the language used across the whole biomedical domain, and thus the semantic lexicons automatically extracted from these will be severely limited. In this thesis, I aim to extract semantic lexicons from all available MEDLINE abstracts, with no topic restrictions (see Section 5.3). Also, the annotated named entities across the available corpora only represent a small fraction of the ten biomedical semantic categories of interest, and it is not possible to determine whether a corpus contains all instances of a semantic category. These limitations of the annotated biomedical corpora and resources make them infeasible for evaluating the precision of the systems, and necessitates manual evaluation of the extracted lexicons to determine the precision.

The evaluation presented in this thesis involves manually inspecting each semantic lexicon, by checking each extracted term and judging whether it is a true member of the lexicon it is assigned to. This process is extremely time consuming, and requires evaluators with domain specific knowledge. To facilitate the work of the evaluators, all evaluators' decisions for each category are cached. This makes later evaluations of new lexicons, which include terms evaluated previously, more efficient as only unseen terms need to be checked.

During the evaluation of a lexicon, all unfamiliar terms are checked using online resources including Pubmed, Medical Subject Headings (MeSH), Wikipedia and Google. The following guidelines were used:

1. Ambiguous terms extracted correctly into one of their semantic categories are correct.
    - *lymphoma* is a member of the TUMOUR and DISEASE categories.
    - *insulin* is a member of the DRUG and PROTEIN categories.
2. Abbreviations of correct terms are correct.
    - *AZT* is an abbreviation for the drug *zidovudine (Retrovir)*.
    - *CPZ* is an abbreviation for the drug *proclorperazine (Compazine)*.

3. Acronyms of correct terms are correct.

   - *BEC* is an acronym for the cell type *Bovine Epithelial Cell*.
   - *HBFC* is an abbreviation for *human bronchial fibroblastic cells*.
   - *DMT1* is an acronym for the disease *Diabetes Mellitus Type 1*.

4. Typographical variations of correct terms are correct.

   - *CIP-1*, *Cip1* and *cip-1* all correspond to the same protein.

5. Obvious misspelt terms are considered correct.

   - *nuetrophils* instead of *neutrophils* (a type of CELL)

6. Terms that are unambiguously part of a correct multi-word term are considered correct.

   - *Parkinson's* in *Parkinson's Disease*

7. Terms that are conjunctions of multiple entities from the same category are correct.

   - *Ig(A/G/M)* refers to a group of Ig antibodies – *IgA*, *IgG* and *IgM*.
   - *p53/p21* is the protein complex formed by the proteins *p53* binding *p21*.

8. Modifiers that are not specific for the individual semantic category are incorrect.

   - *gastrointestinal* may be incorrectly extracted for TUMOUR, as part of the entity *gastrointestinal carcinoma*. However, the modifier may also be used for terms in the DISEASE and CELL lexicons.

   Other non-specific modifiers include *chronic* in *chronic arthritis* (DISEASE) and *chronic pain* (SYMPTOM), and *cultured* in *cultured lymphocytes* (CELL).

5.5 Evaluator Agreement

Annotation and term evaluation in the biomedical domain is a very difficult task, even with adequate guidelines. The manual evaluations were performed by two evaluators with considerable backgrounds in biomedicine. As the evaluations were based on judgements from

a biological viewpoint, the evaluators did not need training in linguistics. I was the primary evaluator (Evaluator 1) with four years of training in molecular biology and pharmacology, and broad knowledge of each of the categories. In particular, ANTIBODY, CELL, CELL LINE, FUNCTION, MUTATION, and PROTEIN. Evaluator 2, is a medical doctor with previous training in molecular biology, and thus has expert knowledge with respect to the DRUG, DISEASE, and SYMPTOM categories, and adequate knowledge for the remaining categories. As stated previously, any term for which an evaluator was not 100% sure of was checked exhaustively using online resources. The second evaluator reevaluated sections previously judged by Evaluator 1, and helped judge difficult terms. This allowed us to determine the inter-evaluator agreement and adjust the guidelines where necessary.

The kappa coefficient of agreement (Landis and Koch, 1977; Siegel and Castellan, 1988) was used as a measure for the correctness of the judged lexicon terms. Kappa measures pairwise agreement between each evaluator, and represents the proportion of agreements, after taking into account the likelihood of chance agreement between the evaluators:

$$\text{kappa} = \frac{Pr(A) - Pr(E)}{1 - Pr(E)} \tag{5.1}$$

where $Pr(A)$ is the proportion of times that the evaluators agree and $P(E)$ is the proportion of times that we would expect them to agree by chance. When evaluating a lexicon, each term's category membership can only be labelled as correct or incorrect. Therefore, the likelihood of agreement by chance, $P(E)$ is set to 0.5. A kappa value of one, reflects complete agreement among the evaluators, and a kappa value of zero reflects no agreement other than that expected by chance. A kappa score above 0.8 is considered to represent "almost perfect" agreement (Landis and Koch, 1977). Carletta (1996) discusses the use of kappa in NLP in detail.

For each category, the correctness of the evaluated lexicon's is measured by calculating the degree of agreement for the two evaluators on 300 extracted terms. These terms correspond to the first 100 terms extracted by three bootstrapping algorithms, BASILISK, MEB and WMEB (described in the following chapter), with the hand-picked seeds (Table 5.1). Evaluator disagreements were discussed and checked extensively using online resources, and the agreement

between the two evaluators before (Kappa_1) and after (Kappa_2) the discussions was measured. The number of unique terms evaluated and the resulting kappa values for each category are shown in Table 5.4. As an algorithm can identify the same terms as another algorithm, the number of unique terms is often less than 300.

5.5.1 Agreement Analysis

Each kappa score is above 0.8 before and after evaluator discussions, reflecting an agreement strength of "almost perfect" (Landis and Koch, 1977). The majority of disagreements before the discussions were considered to be clear errors by either one of the evaluators. During the discussions an evaluator was able to defend their decision with adequate supporting documentation, and as a result most errors were corrected leading to Kappa_2 scores of 1.0.

Agreements before discussions were very high for PROTEIN (0.99), SYMPTOM (0.95), DISEASE (0.95), CELL LINE (0.94), and TUMOUR (0.93). This is reassuring as Evaluator 1 has less domain knowledge than Evaluator 2 for the DISEASE and SYMPTOM categories, and vice versa for the PROTEIN and CELL LINE categories. This was notable in the different times taken to evaluate these categories. For instance, Evaluator 1 took $\sim$ 50 mins to check the SYMPTOM terms and required external resources significantly, whereas Evaluator 2 took only $\sim$ 15 mins using only their domain knowledge. The main concern noted by Evaluator 1, was the difficulty in distinguishing between SYMPTOM and medical conditions that could also be signs. For example, Evaluator 1 labelled the term *anorexia* as incorrect for SYMPTOM, however Evaluator 2 noted that *anorexia* could also be a symptom of *cancer* or *drug abuse*. Another SYMPTOM disagreement occurred with the term *rhabdomyolysis*. Evaluator 2 considered this term not to be a sign but a syndrome/condition. However, Evaluator 1 who was unfamiliar with this term identified MEDLINE abstracts referring to it as a symptom.

The category with the least agreement was ANTIBODY (0.86). There were a total of 15 unique terms for which a disagreement occurred. The majority of terms corresponded to references to immunoglobulin molecules, including the terms *immunoglobulins* and *Ig*. Evaluator 2 considered these terms too general, however after discussions both evaluators agreed they

Category	Number of unique terms	Kappa_1	Kappa_2
ANTIBODY	213	0.86	1.00
CELL	192	0.90	1.00
CELL LINE	217	0.94	1.00
DISEASE	265	0.95	1.00
DRUG	269	0.87	1.00
FUNCTION	250	0.88	1.00
MUTATION	234	0.88	1.00
PROTEIN	235	0.99	1.00
SYMPTOM	172	0.95	1.00
TUMOUR	195	0.93	0.99

Table 5.4: The Kappa statistics for each semantic category

should be included. Another discrepancy occurred with the term *IgAIC*. *IgAIC* refers to *circulating immune complexes* which contain the antibody *IgA* bound to its specific antigen. Although this term is not specifically a reference to an individual antibody, it was agreed that the semantic category should include terms referring to *complexes*.

Discrepancies between the evaluators for the DRUG category tended to be on terms which are both DRUG and PROTEIN. For example, the DRUG *Cholecystokinin* was known by Evaluator 1 as a naturally occurring PROTEIN formed in the small intestine, whereas Evaluator 2 was aware of its use as a diagnostic aid injected into patients to determine if the gallbladder and pancreas are functioning normally. During the evaluation, Evaluator 2 also noted DRUGs they were unfamilar with, that they checked using other resources. For example, the term *enrofloxacin*, is an antibiotic for ornamental fish (Reimlinger et al., 1990), and American brand names of drugs.

After the discussions, all categories except TUMOUR had 100% agreement. The evaluators still disagree on the term *MDR* as a TUMOUR. *MDR* is an abbreviation for the modifier *multidrug resistant*. The evaluators disagreed on whether the modifier was specific enough for describing only TUMOUR.

5.6 Performance Measures

In this section, I describe the performance measures used to compare the bootstrapping algorithms. These include precision, inverse rank score, and lexicon overlap.

5.6.1 Precision

The *precision* of the generated lexicons (Eq 5.2) is commonly used to compare the performance of bootstrappers. $N_{correct}$ is the number of correct lexicon items extracted by the system, and $N_{incorrect}$ is the number of incorrect lexicon terms. Precision scores assume values in the range of 0 to 100%, where low values reflect less accurate lexicons, and a value of 100% corresponds to a perfect set of terms.

$$\text{precision} = \frac{N_{correct}}{N_{correct} + N_{incorrect}} \tag{5.2}$$

For comparing the performance of the bootstrapping algorithms, the precision of the top-n terms (P(n)) extracted by each semantic category is reported, from the top 100 terms (P(100)) up to the top 1000 terms (P(1000)). For many experiments, the average precision for the top-n terms over the ten semantic categories (Av(n)) is reported. The precision of term samples within the top 1000 terms, such as the 801-1000 term sample which corresponds to the last 200 terms in each lexicon (P(801-1000)) is also measured. Evaluating term samples allows us to identify when semantic drift becomes a significant problem or has a significant impact.

5.6.2 Inverse Rank Score

The *inverse rank score* of the top-n terms (InvRank(n)), which is the sum of the inverse rank of each correct term in the top-n set (Eq 5.3), is also used. The maximum inverse rank score of a lexicon of 1000 terms, which is achieved when precision is 100%, is ~ 7.485 and is calculated as: $\frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{1000}$. One characteristic of the inverse rank measure is that it rewards correct terms extracted in the early iterations more than those in the middle or towards

the end of the lexicons. Thus, inverse rank is a good measure to identify subtle differences between extracted lexicons with similar overall precisions. For example, if terms at ranks 4, 16 and 32 are the only correct terms, then the inverse rank score is ~ 0.344, whereas if the correct terms are at ranks 2, 14 and 30, the inverse rank score is ~ 0.605.

$$\text{InvRank} = \sum_{i=1}^{N_{correct}} \frac{1}{rank_i} \tag{5.3}$$

5.6.3 Lexicon Overlap

The degree of term *overlap* between different lexicons can also be used to compare algorithms. The overlap of two lexicons that have been initialised with the same set of seeds, A and B, is given by:

$$\text{overlap} = |A \cap B| \tag{5.4}$$

The level of overlap can indicate how similar the term and pattern ranking methods are between different algorithms. Overlap can also be used to determine the sensitivity of an algorithm to the seeds by comparing lexicons generated by different seed sets. A sensitive algorithm would extract lexicons with very little term overlap.

5.6.4 Statistical Significance Testing

To detect statistical significant differences in precision of two bootstrapping methods, I apply computationally-intensive randomisation tests, as described in Noreen (1989) and Cohen (1995, §5.3). I use a randomised version of the paired sample t test (Cohen, 1995), which is a type of stratified shuffling (Noreen, 1989, §2.7).

Under the null hypothesis, the two algorithms are considered to perform similarly, so any lexicon item extracted by one of the algorithms could have equally likely been extracted by the other. The null hypothesis is tested by shuffling the terms in the lexicons, and reassigning each term randomly to one of the two lexicons. This is performed nc times, and the precision of each new lexicon is recalculated. A count (nt) is kept of how many times the difference between the

recalculated precision is greater than or equal to the difference between the original lexicons' precisions. Following Noreen (1989), the significance level (p-value), is at most:

$$\mathrm{p} = \frac{nc + 1}{nt + 1} \tag{5.5}$$

Ideally, we would consider all possible term permutations — 2^n, where n is the total number of different lexicon terms. As this is infeasible, I set nc to 10,000. The null hypothesis is rejected if the p-value is greater than 0.05.

5.7 Manual Evaluation of Patterns

The manual evaluation also includes judging the quality of the patterns extracted by each algorithm. Unlike the extracted lexicons, the patterns generated during the bootstrapping process are often ignored and never evaluated. Despite this, they have been used within question answering systems, to identify potential answer strings. The winning system at the TREC-10 QA Evaluation (Voorhees, 2001) used an extensive list of manually constructed patterns which may be indicative of answers to certain question types (Soubbotin and Soubbotin, 2001). For example, the pattern:

<city name> , <country name>

is likely to answer some *Where is* type questions. Following Soubbotin and Soubbotin's success, others have utilised automatically discovered patterns, such as, Ravichandran and Hovy (2002) on the TREC-10 QA collection, and Meij and Katrenko (2007) on the TREC Genomics QA task (Hersh et al., 2007).

In this thesis, the patterns are evaluated in order to gain intuition about the behavior of the algorithms. The pattern evaluation involved manually classifying patterns extracted by each semantic category into three classes: true matches, multi-matches and non-matches. Patterns where the context is semantically consistent with terms only from the assigned semantic category are classified as *true matches*. These patterns are considered to accurately define the

category. Common true match patterns involve lists of entities from the same semantic category. For example:

- *IgG1 , <entity> and IgE*

  This pattern corresponds to the end of a list. Both *IgG1* and *IgE* are antibodies, and thus the wildcard entity is most likely also an antibody.

- *vomiting , <entity> , fatigue*

  This pattern corresponds to part of a list. Both *vomiting* and *fatigue* are signs and symptoms, and thus the wildcard entity is also most likely to be a sign or symptom.

- *such as <entity> , hepatitis*

  This pattern corresponds to the start of a list, with only one entity, the disease *hepatitis*. Therefore the wildcard entity is most likely to be a sign or symptom too. This pattern is reminiscent of those used by Hearst (1992).

- *erythromycin , <entity> , and*

  This pattern corresponds to the later part of a list, and as such only contains one entity, the drug *erythromycin*. Therefore, the wildcard entity is most likely to also be a drug.

Other true match patterns extract acronym or abbreviations of semantic categories, e.g.:

- *virus (<entity>) infections*
- *antibody (<entity>) activity*
- *antibody (<entity>) treatment*

Finally, patterns for which the context surrounding the wildcard is specific to the semantic category are also considered true matches. For example:

- *pharmacokinetics of <entity> after intravenous*

  The wildcard in this pattern can only belong to the semantic category drugs, as the term *pharmacokinetics* refers to the action of drugs in the body over time, and drugs are often administered directly into a vein (intravenously).

- *regulation of <entity> mRNA levels*

  The wildcard in this pattern corresponds to the semantic category Genes and Proteins. Only genes have mRNA transcripts, and the verb regulate is often associated with members of this category.

- *correlation between <entity> expression and*

  The term *expression* in molecular biology refers to the process of transcribing a gene's DNA sequence into its respective proteins mRNA sequence. Therefore the wildcard can only belong to the semantic category PROTEIN.

- *outbreak of <entity> in the*

  Even though the context to the right of the wildcard is not category specific, only entities within the disease semantic category can be in reference to outbreaks.

Multi-matches correspond to patterns where not only terms from the assigned semantic category can be inserted in the wildcard position, but terms from other categories as well. For example:

- *mechanism of <entity> action .*

  Many entities have specific actions they perform, including antibodies, genes and proteins, and drugs.

- *in malignant <entity> patients .*

  The adjective *malignant* is often used to describe types of tumours. However, it is also used to describe other diseases and conditions which are resistant to treatment, such as *malignant hypertension* and *malignant hyperthermia*.

- *We transfected <entity> cells with*

  This pattern was extracted by the CELL LINE category, however the wildcard may also be replaced with terms referring to body parts such as, *liver* and *epithelial*.

Note that, patterns may be true or multi-matches if a general English term can be inserted at the wild card position at low frequencies. For instance, *the* can be inserted into the above pattern.

Patterns that do not provide any contextual information for a specific semantic category of interest are *non-matches*. For example:

- *of the <entity> subclass in*
- *compared with <entity> for the*
- *Value of <entity> in the*
- *concentration of <entity> and complement*

Using this pattern classification scheme we can further investigate and compare the behavior of boot- strapping algorithms, by comparing the number of true, multi- and non-match patterns each algorithm extracts.

5.8 Summary

This chapter introduced the NLP task of automatically extracting biomedical semantic lexicons from raw text, which is the focus of the following chapters. The ten semantic categories of interest and the biomedical corpora their lexicons are extracted from were described. This chapter also presented the methodology used in this thesis for evaluating the precision of the extracted lexicons and patterns. This is followed by a detailed discussion regarding the reliability of the manual evaluation by two domain experts using the kappa statistic for inter-evaluator agreement. For each of the categories, the kappa scores before and after disagreement discussions reflected near perfect agreement and therefore this task is well defined.

The categories, corpora and evaluation methods described here will be used in this thesis to evaluate the multi-category bootstrappers BASILISK, MEB, and my bootstrapping approaches that are described in the following chapters.

Chapter 6

Weighted Mutual Exclusion Bootstrapping

This chapter introduces and evaluates my minimally supervised algorithm, *Weighted Mutual Exclusion Bootstrapping* (WMEB). WMEB simultaneously extracts multiple semantic lexicons and their patterns from raw text. The first two sections describe WMEB, and correspond to the work presented in McIntosh and Curran (2008). WMEB extends Mutual Exclusion Bootstrapping (MEB, Curran et al., 2007), by incorporating candidate term and pattern weighting functions (Section 6.2.3), and a cumulative pattern pool (Section 6.2.1), while still enforcing mutual exclusion between the categories' lexicons and patterns.

WMEB is evaluated on the biomedical extraction task from Chapter 5 and compared to two state-of-the-art multi-category bootstrapping algorithms — MEB and BASILISK (Thelen and Riloff, 2002). In Section 6.3.1 and 6.3.2, the new components of WMEB are shown to reduce the semantic drift occurring in MEB significantly. I demonstrate that WMEB is capable of extracting larger lexicons from raw biomedical text with significantly higher precision than both BASILISK and MEB.

6.1 Motivation

WMEB was motivated by a desire to automatically extract large biomedical semantic lexicons from raw biomedical text with little manual effort from domain experts. The general framework of bootstrapping, discussed in Chapter 4, is appealing for this task. In particular, many of the algorithms require only a few example seed terms of the categories of interest to initiate the extraction process. For example, both BASILISK and NOMEN were seeded with ten terms for each category (Thelen and Riloff, 2002; Lin et al., 2003b). Apart from these initial seed terms, the majority of the bootstrapping algorithms presented also require no manual intervention as the candidate terms and patterns are ranked and selected automatically. Despite this, many of the algorithms are not directly suitable for extracting lexicons from biomedical raw text. For example, for both MLB (Riloff and Jones, 1999) and BASILISK (Thelen and Riloff, 2002) the input text needs to be shallow parsed. This means that a POS tagger and chunker must be available for the target domain. As one of the aims of WMEB is to also be domain independent, WMEB utilises the patterns exploited by MEB which are completely language independent (Curran et al., 2007).

The precision of the lexicons extracted by bootstrapping multiple semantic categories in parallel, as in BASILISK, NOMEN and MEB is encouraging and has prompted the use of this bootstrapping framework in WMEB. WMEB extends MEB (Curran et al., 2007) and exploits the assumption that both the terms in the semantic lexicons and their patterns are mutually exclusive. MEB's framework also considers both term and pattern collisions, and is both time and memory efficient — in WMEB the same term and pattern cross-indexing used in MEB is utilised. Although many of the features of MEB are promising, it suffers from semantic drift in the early iterations, and is thus not suitable for extracting large yet precise biomedical semantic lexicons. The development of WMEB particularly focused on reducing semantic drift.

In each of the bootstrapping algorithms described in Chapter 4 only a small number, k, of extracting patterns are selected in each iteration to identify new candidate terms or facts (Paşca et al. (2006) is an exception). In my initial experiments, I observed that as the lexicons grow,

more general patterns can drift into the top-k patterns. This was also noted by Jones et al. (1999). As a result, the earlier precise patterns lose their extracting influence, and the lexicon's meaning drifts. WMEB overcomes this by incorporating a cumulative pattern pool, to retain precise patterns.

Another cause for semantic drift in MEB is the naïve term and pattern scoring metrics used. MEB prefers terms and patterns that 1) match the most input instances (high reliability); and 2) have the potential to generate the most new candidates (high productivity). More reliable terms and patterns would theoretically have higher precision, while more productive instances will have a high recall. As noted in Chapter 4, the first criteria is bounded by the fixed small number of patterns (k) selected (for terms) or the lexicon's size (for patterns). The smaller the bound, the more likely ties in the reliability score will occur, which will result in the second criteria being used more often for ranking. As the number of extracting patterns is fixed, this occurs very often during candidate term ranking. Unfortunately, as the productivity measure aims to increase recall, the selected terms/patterns are then highly likely to introduce drift.

In the development of WMEB, I have addressed these issues and in doing so, have reduced semantic drift significantly. In WMEB, MEB's *productivity measure* is replaced with a new weighting scheme which aims to extract terms and patterns that are highly associated with their input instances. Furthermore, the introduction of the cumulative pattern pool also enables the *reliability measure* to be more distinguishing with respect to the candidate terms.

6.2 Algorithm

WMEB is a multi-category bootstrapping algorithm that builds semantic lexicons and identifies their extracting patterns from unlabeled raw text. In WMEB, each semantic category's lexicon and patterns are extracted using a single bootstrapping instance that is initialised with a small set of seed terms. Each bootstrapping instance iterates between two phases simultaneously with the other semantic categories. In Phase I, WMEB expands the cumulative pattern pool with the top-m scoring patterns that identify the terms in each category.

If each of the top-m patterns are already members of the cumulative pool, the next top non-member pattern is added. Therefore, in each iteration at least 1 new pattern is added. In Phase II, WMEB selects the top-n new terms to add to each category's lexicon, based on the cumulative pattern pool expanded in Phase I. In each phase, mutual exclusion between the categories is utilised to handle category conflicts. In the experiments reported here, m and n are set to 5. Pseudo-code for WMEB is provided in Algorithm 1 (page 141).

This section describes the architecture of WMEB in detail. WMEB employs a new weighting scheme, which identifies candidate patterns and terms that are strongly associated with the lexicon terms and the patterns in the cumulative pool, respectively. These techniques reduce the semantic drift in WMEB much more effectively than MEB.

6.2.1 Phase I - Pattern Extraction and Selection

WMEB takes as input a set of manually labelled seed terms for each category, and the number of terms (n) and patterns (m) to add in each iteration, and the maximum number of bootstrapping iterations. The bootstrapping process begins with each category's seed set forming its initial lexicon of terms, and an empty cumulative pattern pool for each category.

For each term in a category's lexicon, WMEB identifies all of the contexts surrounding each occurrence of the term in the corpus. Each term occurrence is assigned a pattern that consists of the pre-computed sequence of the two tokens before and after the term. These patterns are collected to form the set of candidate patterns for the category, which are then checked for collisions with other categories. A collision occurs if the candidate pattern appears in another category's cumulative pool or set of candidate patterns. Following the mutual exclusion assumption, all colliding patterns are excluded from the current bootstrapping iteration. The remaining candidate patterns for each category are then ranked according to their *reliability* measure and their *relevance weight* (see Section 6.2.3). In each iteration, the candidate patterns must be re-scored, as the scores are based on the current state of the category's lexicon.

Cumulative Pattern Pool

When only the top-m patterns are used to extract new lexical items, the highly precise patterns used in the earlier iterations can become out-ranked by more generic patterns in later iterations. This will lead to semantic drift. WMEB aims to reduce this by accumulating all of the top-m patterns from the current and all previous iterations in the cumulative pattern pool. This ensures all previous patterns can still contribute in the later iterations.

In WMEB, the top-m candidate patterns are selected for addition into the cumulative pattern pool. If all of the top-m candidate patterns are already members of the pool, the pool will become stagnant. To overcome this, WMEB ensures at least one new pattern is added to the pool in each iteration — the next best pattern that is not already in the pool is added. This process is similar to that of BASILISK, where the number of extracting patterns used is incremented by one in each iteration (Thelen and Riloff, 2002). However, their approach does not maintain all patterns identified in previous iterations that are no longer in the top-m. Each of the patterns in the pool have equal weight in all subsequent iterations for extracting candidate terms.

6.2.2 Phase II - Term Extraction and Selection

After generating the extraction patterns for each target category, WMEB scans the corpus to discover new lexical items for each category. For each category, WMEB first identifies all candidate terms that match the extracting patterns in the category's cumulative pool. Many terms are likely to be identified by multiple patterns, and possibly by patterns from multiple categories. Like the candidate patterns, terms which are extracted by multiple semantic categories within the same iteration are also excluded. Note that a colliding term will collide in all consecutive iterations due to the cumulative pool and thus WMEB creates a stricter term boundary between categories than MEB.

The remaining candidate terms in each category are then ranked with respect to their *reliability* and *relevance weight* (see Section 6.2.3). Candidate terms are re-scored in each iteration, as the scores are based on the patterns in the cumulative pool, which expands in each iteration.

After scoring, the top-n terms are added to the category's lexicon. Highly ranked terms that are not selected in this iteration, can potentially be selected in a later iteration of the algorithm. The expanded lexicons for each category are then used to seed the next bootstrapping iteration, by identifying new candidate patterns in Phase I.

6.2.3 Term and Pattern Relevance Weighting

In MEB, candidate terms and patterns are ranked according to their *reliability* measure and ties are broken using the *productivity* measure. The *reliability* of a term for a given category, is the number of input patterns in an iteration that can extract the term. The *productivity* of a term is the number of potentially new patterns it may add in the next iteration. These measures are symmetrical for both terms and patterns. As mentioned previously, in MEB, the reliability score for terms is bounded by the fixed number of extracting patterns (set to 5 or 10). This frequently results in ties when ranking terms, which leads to the use of the productivity measure, and is thus likely to introduce semantic drift. By incorporating the cumulative pattern pool into WMEB, the discriminatory power of the reliability measure is increased.

In WMEB, the productivity measure from MEB is replaced with a new *relevance weight*. I have investigated various scoring metrics which prefer candidate terms that are strongly associated with the patterns in the cumulative pool, and candidate patterns that are strongly associated with terms in the growing lexicon. These include three variations of the Pointwise Mutual Information measure (PMI) and the chi-squared statistic (χ^2). The fundamentals of these measures are described in Manning and Schütze (1999, Chapter 5). Each of these measures estimates the strength of the co-occurrence of a term and a pattern pair. They do not give the likelihood of the candidate instance being a member of a semantic category, only the chance of seeing the term and pattern pair together in the corpus.

These co-occurrence measures, also allow WMEB to maintain MEB's scoring efficiency as much as possible — based on the efficient representation of term and pattern pairs in WMEB, all scores for each possible term and pattern pair can be pre-calculated before the bootstrapping process begins. The scoring metrics are also symmetrical for both terms and patterns, and thus

only need to be stored and calculated once for each pair. In BASILISK, the scoring metric is more computationally expensive, and each individual calculation is dependent on the current state of the bootstrapping process, and therefore scores cannot be pre-calculated. Therefore, WMEB is much more efficient than BASILISK.

The overall *relevance* weight for a candidate term, t, is the sum of the scores of the term with each matching pattern c in the pattern pool P (Equation 6.1), and the relevance weight for a candidate pattern, c, is the sum of the scores of the pattern with each term t in the expanding lexicon T (Equation 6.2):

$$\text{relevance}(t) = \sum_{c \in P} \text{score}(t, c) \tag{6.1}$$

$$\text{relevance}(c) = \sum_{t \in T} \text{score}(c, t) \tag{6.2}$$

where score corresponds to one of the scoring metrics, described below. The candidate terms and patterns are ordered by their *reliability*, and ties are broken by their *relevance* weight.

Point-wise Mutual Information

Point-wise Mutual Information (PMI) (Fano, 1963) is commonly used in NLP to calculate dependencies between two random events, such as co-occurring terms.[1] PMI compares the probability of observing two events together with the probabilities of observing two events independently. Therefore, it can be used to estimate whether the two events have a genuine association or were observed together by chance. The PMI measure of a term t and a pattern c is defined as:

$$\text{PMI}(t, c) = \log_2 \frac{p(t, c)}{p(t)p(c)} \tag{6.3}$$

[1]The term *mutual information* is commonly used to mean *point-wise mutual information*. The *mutual information* defined in Fano (1963) corresponds to the *point-wise mutual information* measure now used, and the term *mutual information* now refers to the *expectation of the mutual information* defined in Fano (1963).

where $p(t)$ and $p(c)$ is the probability of the term t and pattern c occurring in the corpus, respectively, and $p(t, c)$ is the probability that t and c co-occur. Each of the probabilities are calculated directly from the relative frequencies without smoothing. These probabilities can also be pre-calculated during the loading stages of WMEB, and thus the PMI between all possible term and pattern pairs can be determined before bootstrapping begins.

If there is a genuine relationship between t and c, then $\text{PMI}(t, c) \gg 0$ as the joint probability $p(t, c)$ will be much larger than chance $p(t)p(c)$. If there is no significant association between t and c, then $\text{PMI}(t, c) \approx 0$, as $p(t, c) \approx p(t)p(c)$. PMI can also take on negative values if $p(t, c) \ll p(t)p(c)$, which suggests that t and c must occur at very high frequencies yet do not occur together often.

The PMI measure is prone to overestimating co-occurrence events with low observed frequency counts. This is because PMI does not incorporate the evidence for when t occurs without c, and vice versa. The first variation of PMI investigated, PMI^2, is used to help overcome this (Equation 6.4). In PMI^2, the ratio of probabilities is scaled by the joint probability of the term and pattern co-occurring, which ensures more frequent combinations of terms and patterns have a greater weight:

$$\text{PMI}^2(t, c) = \log_2 \frac{p(t, c)^2}{p(t)p(c)} \tag{6.4}$$

The PMI and PMI^2 measures are sensitive to low frequencies, and thus are set to 0 if the observed frequencies of t or c are less than 5.

Negative values of PMI tend to suggest that t and c are co-occurring less often than by chance and negative values tend to be unreliable unless the corpus is enormous. Therefore, PMI values are often restricted to non-negative values as in Dagan et al. (1993) and Lin (1998b). Following this suggested restriction, the final version of PMI I experimented with is *truncated* PMI (PMIT, Equation 6.5).

$$\text{PMIT}(t,c) = \begin{cases} \text{PMI}^2(t,c) & : \quad \text{PMI}(t,c) > 0 \\ 0 & : \quad \text{PMI}(t,c) \leq 0 \end{cases} \tag{6.5}$$

The pointwise mutual information measure has also been incorporated in the instance ranking measures of other bootstrapping algorithms, such as KNOWITALL (Etzioni et al., 2005) and Espresso (Pantel and Pennacchiotti, 2006). In the KNOWITALL system, the likelihood that a term belongs to a semantic class is determined by computing the PMI of the term and associated *class discriminator phrases* (such as "X is a city" for the class CITY). The probabilities are estimated from Web search engine hit counts (Etzioni et al., 2005). In Espresso, PMI is used in a similar way to WMEB: for ranking candidate patterns based on the previously extracted relations; and ranking candidate relations based on the top extracting patterns (Pantel and Pennacchiotti, 2006).

Pearson's Chi-squared Statistic

The Pearson's χ^2 statistical test (χ^2-test) of independence can be used to test if a term and pattern pair in the corpus are independent of each other. Here, the null hypothesis is that there is no relationship or dependence between the term and pattern pair — that is, the observed frequencies of their co-occurrences are what one would expect by chance. From the corpus frequencies, a 2 by 2 contingency table can be formed, which stores the frequencies of four different combinations of events involving the term and the pattern in individual cells. The four cells store the observed frequency of the term t and pattern c co-occurring (O_{tc}), the term appearing without the pattern ($\text{O}_{t\bar{c}}$), the pattern appearing without the term ($\text{O}_{\bar{t}c}$), and neither the term or pattern occurring ($\text{O}_{\bar{t}\bar{c}}$). Using this contingency table, the χ^2 value for a term (t) and pattern (c) pair is given by:

$$\chi^2(t,c) = \frac{\text{N}(\text{O}_{tc}\text{O}_{\bar{t}\bar{c}} - \text{O}_{t\bar{c}}\text{O}_{\bar{t}c})^2}{(\text{O}_{tc} + \text{O}_{t\bar{c}})(\text{O}_{tc} + \text{O}_{\bar{t}c})(\text{O}_{t\bar{c}} + \text{O}_{\bar{t}\bar{c}})(\text{O}_{\bar{t}c} + \text{O}_{\bar{t}\bar{c}})} \tag{6.6}$$

where N is the total number of term and pattern pairs in the corpus. When using the χ^2 statistic for hypothesis testing, the value is compared to a critical value. If the χ^2 value is below the critical value, the term and pattern pair is considered to be statistically independent and the null hypothesis is accepted. If the value is above the critical value, the null hypothesis is rejected. In WMEB, the χ^2 values are used directly without considering the critical value cutoff. The χ^2 value of a term and pattern pair is set to 0 if their individual observed frequencies are less than 5, as this estimate is also sensitive to low frequencies. In the relevance weighting, the χ^2 value for each candidate term with all patterns in the pool is determined, and summed together. A candidate term that has a higher total than another is ranked more highly. This is symmetrical for pattern ranking.

Based on the efficient representation of term and pattern pairs in WMEB, before the bootstrapping process begins, all contingency table cell values, and thus the χ^2 value for each possible term and pattern pair can be pre-calculated. Therefore, during WMEB only the summation of the relevant stored χ^2 values is required to determine the relevance weight of a candidate term or pattern. The calculation of the χ^2 value is also fast.

6.3 Results

This section presents the results related to the performance of WMEB, for extracting biomedical semantic categories from raw biomedical texts. The first set of results evaluate the individual components of WMEB. In Section 6.3.1, the different scoring metrics which can be used for calculating the term and pattern relevance weights are compared. Section 6.3.2 investigates the impact of the individual weighting and cumulative pattern pool components of WMEB. These results are followed by detailed comparisons of WMEB with BASILISK and MEB.

Section 6.3.3 discusses the effectiveness of the STOP categories in BASILISK, MEB and WMEB. This is followed by a detailed analysis of the individual biomedical lexicons extracted by each algorithm from MEDLINE, and Section 6.3.5 considers the performance of the algorithms on TREC. My analysis concludes with an evaluation of the extracted patterns.

Weight function	1-100	101-200	201-300	301-400	401-500	1-500
PMI	86.0	90.3	86.9	88.8	82.5	86.7
PMI^2	85.7	82.8	81.2	77.8	66.0	78.7
PMIT	87.3	84.5	85.2	81.2	70.0	81.6
χ^2	**88.1**	**92.2**	**89.0**	**91.0**	**83.4**	**88.7**

Table 6.1: Evaluation of term and pattern weighting functions in WMEB

For each of the experiments, unless otherwise stated, all algorithms extract lexicons from the MEDLINE dataset and are initialised with seeds from the 10 biomedical semantic categories and the STOP categories listed in Table 5.1 and 5.2. The maximum number of terms and patterns that can be added in each iteration is set to 5.[2] A re-implementation of the BASILISK system is carried out for the evaluations. This version of BASILISK, does not use syntactic information for forming patterns and uses the same window-based patterns as MEB and WMEB. My implementation also incorporates the speed and memory optimisations used in MEB.

6.3.1 Relevance Weighting

Table 6.1 summarises the variation in performance of WMEB with each of the four term and pattern weighting metrics. The table shows the average precision over the 10 categories for various sections of the extracted lexicons. The last column presents the average precision over the first 500 (1-500) terms extracted. The first three rows show the performance of WMEB using the different Point-wise Mutual Information measures (Section 6.2.3). The traditional PMI score, which can overestimate infrequent co-occurrences between terms and patterns, significantly outperforms the other PMI measures in the later iterations ($p \leq 0.0001$).

The PMIT measure, which excludes negative scores from the total weight (negative values are truncated to 0), compares well with the other metrics over the first 100 extracted terms. However, its performance drops below PMI in the later iterations. This indicates that the negative scoring terms are important in bootstrapping, whereas for distributional similarity the truncated measures are more effective (Curran, 2004).

[2]The number of patterns is incremented by 1 in each iteration of BASILISK.

Score	1-100	101- 200	201-300	301-400	401-500	1-500
MEB	77.6	65.9	56.5	50.0	49.6	59.6
WMEB-weight	84.7	76.6	71.2	69.1	65.8	73.4
WMEB-pool	83.1	77.7	76.6	75.1	72.3	77.0
WMEB	**88.1**	**92.2**	**89.0**	**91.0**	**83.4**	**88.7**

Table 6.2: Evaluation of WMEB's pattern pool, and term and pattern weighting

The last row in Table 6.1 shows the performance of incorporating the χ^2 statistic in WMEB's relevance weight. Over each range of extracted terms, χ^2 performed higher than the other measures, and is most similar in terms of performance consistency to PMI. The χ^2 measure is significantly more effective than PMI (1-100: $p \leq 0.0023$; 1-500: $p \leq 0.0001$). As χ^2 outperforms the PMI measures, the χ^2 weighting function is used for the remaining WMEB experiments throughout this thesis.

6.3.2 Components

Table 6.2 shows the effectiveness of each of the individual components of WMEB. The first row corresponds to MEB, which is equivalent to WMEB with no term and pattern relevance weighting or cumulative pattern pool. The second row, WMEB-weight, corresponds to WMEB without the pattern pool, and shows the effect of replacing MEB's productivity measure with the relevance weighting (χ^2). The next figures, WMEB-pool, correspond to WMEB with the cumulative pool, but no relevance weighting (using MEB's productivity measure). The final row (WMEB) shows the results when both the relevance weighting and cumulative pattern pool are included in WMEB.

The addition of the weighting function (WMEB-weight) is more effective than the cumulative pool (WMEB-pool) over the first 100 extracted terms, and both components substantially improve upon MEB in the later iterations. The cumulative pattern pool is more effective in reducing semantic drift, with a significant precision increase over the last 100 extracted terms (+12.7% at 401-500; $p \leq 0.0001$).

The combination of the components (WMEB) improves upon their individual precision gains, and together significantly outperforms MEB. A notable difference of more than 10%

is achieved within the first 100 terms. Further significant improvements occur in the later iterations, such as the 33.8% improvement over the last 100 extracted terms. These results show that WMEB significantly outperforms MEB, and demonstrates that both components of WMEB successfully reduce semantic drift.

6.3.3 Stop Categories

This section discusses the precision of the individual lexicons extracted by BASILISK, MEB and WMEB, with and without STOP (nostop) categories. Note that the STOP categories were selected for MEB, after manually inspecting the MEB-nostop lexicons to identify potential sources of semantic drift. The first STOP category created was BODY PART, and was specifically formed with the intention of preventing semantic drift in the CELL, DISEASE and SYMPTOM categories. In MEB-nostop, the SYMPTOM category quickly extracted modifiers of SYMPTOMS e.g. *increases, rise*, and *variations*, and also modifiers pertaining to BODY PARTS. In the later iterations of MEB-nostop, the CELL and DISEASE categories also drifted into BODY PART and their modifiers. The ANIMAL and ORGANISM STOP categories were also created with these three difficult categories in mind, and were also a source of semantic drift for the DISEASE category in BASILISK. The final STOP category, AMINO ACID, was created with an aim to filter common MUTATION errors that occurred in each algorithm.

The performance of BASILISK, MEB and WMEB, with and without STOP categories, is shown in Table 6.3. These figures include the first comparison of WMEB and MEB with BASILISK, and show that BASILISK is more effective than MEB at reducing semantic drift, but less effective than WMEB.

Without the STOP categories, both BASILISK-nostop and WMEB-nostop significantly outperform MEB-nostop over all precision measurements. WMEB-nostop performs similarly to BASILISK-nostop over the first 100 terms, however WMEB-nostop achieves significantly higher precision in the later measurements. Furthermore, WMEB-nostop outperforms both BASILISK and MEB.

Algorithm	1-100	101-200	201-300	301-400	401-500	1-500
BASILISK-nostop	81.2	73.4	67.7	64.3	66.7	70.6
BASILISK	81.0	72.9	68.0	67.8	69.3	71.7
MEB-nostop	74.4	61.8	50.5	45.0	38.0	53.7
MEB	77.6	65.9	56.5	50.0	49.6	59.6
WMEB-nostop	81.3	78.2	79.1	74.7	71.6	77.1
WMEB	**88.1**	**92.2**	**89.0**	**91.0**	**83.4**	**88.7**

Table 6.3: Performance of bootstrappers on MEDLINE with and without stop categories

When the STOP categories are introduced into BASILISK, the average precision decreases slightly over the first 200 terms, and improvement gains are then noticed in the later iterations. This late increase in performance was also noted by Thelen and Riloff (2002), and this is likely to be a result of two properties of BASILISK. Firstly, the terms extracted in the earlier iterations occur infrequently in the dataset and so are unlikely to cause rapid semantic drift. It is not until later that BASILISK extracts more frequent terms, which leads to more category collisions including with STOP categories. Secondly, the incorrect terms that are extracted by a category are often those of the other categories of interest which are semantically close. This is because the competing categories' patterns are not mutually exclusive in BASILISK. Therefore, it is not until the later iterations, that the categories drift into the STOP categories. As a result the precision improves in the later iterations with the inclusion of the STOP categories (e.g. +2.6% at 401-500). The calculated p-value for the first 500 terms is 0.0003, and this shows that the 1.1% precision increase from incorporating the STOP categories into BASILISK is statistically significant.

Unlike BASILISK, when the STOP categories are introduced into MEB and WMEB, the average precision for each method improves significantly across all ranges, especially in the later iterations (e.g. MEB +11.6% and WMEB +11.8% at 401-500). This is because, these algorithms prefer more frequent terms, and the semantic categories are mutually exclusive, and thus the categories can immediately compete with the STOP categories to prevent semantic drift.

6.3.4 Individual Categories

The results presented so far only show each algorithm's precision averaged over each of the ten semantic categories. A more detailed performance overview is provided in Tables 6.4 and 6.5. These tables show each algorithm's precision, with and without STOP categories, for the first and last 100 terms extracted into each individual category, as well as the average over all categories and the averaged inverse rank. For the remainder of this section, these results will be discussed in detail.

It is clear that some categories are much easier than others to extract, e.g. CELL LINE and PROTEIN, while others are quite difficult, e.g. FUNCTION and TUMOUR. In particular, PROTEIN is almost perfect with and without the STOP categories across the different methods. For many categories there is a wide variation across the algorithms' performance. For example, WMEB performs much better on ANTIBODY, and BASILISK performs much better on DISEASE. Interestingly, the STOP categories had varied and some unexpected influences on the precisions of the semantic lexicons extracted by each method.

In BASILISK-nostop, categories rarely drift into unspecified categories. However, the categories do drift into semantically similar categories of interest. In particular, ANTIBODY, which was the most poorly extracted category by BASILISK (53% at 1-100 and 2% at 401-500), drifted almost completely into the PROTEIN category. In BASILISK, the TUMOUR category also drifted into the CELL LINE category. This is possible as the extracting patterns in BASILISK are not mutually exclusive. Unfortunately, as these categories drifted into other categories of interest, no additional STOP categories could be devised to prevent this. In comparison, in MEB and WMEB the ANTIBODY and PROTEIN, and CELL LINE and TUMOUR categories naturally compete against each other and are thus forced apart. However, this increases the likelihood of the categories drifting into additional unknown categories, which is the case for DISEASE and TUMOUR in both MEB and WMEB.

Over the first 100 terms, the CELL category was extracted by each algorithm with over 90% precision. However, severe semantic drift occurred in the later iterations of MEB and

Category	BASILISK		MEB		WMEB	
	NOSTOP	STOP	NOSTOP	STOP	NOSTOP	STOP
ANTIBODY	53	53	95	93	98	96
CELL	93	93	95	98	98	100
CELL LINE	91	82	93	100	96	100
DISEASE	94	89	43	42	60	84
DRUG	67	63	79	94	90	99
FUNCTION	80	80	61	61	73	82
MUTATION	90	90	88	70	88	84
PROTEIN	100	100	99	100	100	100
SYMPTOM	98	99	56	96	67	99
TUMOUR	56	56	35	22	43	37
Average	81.2	80.5	74.4	77.6	81.3	88.1
Inv Rank	4.5	4.5	4.2	4.3	4.7	4.8

Table 6.4: MEDLINE individual category results (1-100 terms)

WMEB. As expected, with the introduction of the STOP categories, the precision of both MEB and WMEB on CELL improved significantly. In particular, MEB increased from 2 to 42% at 401-500 terms, but this is still much worse than BASILISK. Many of the terms extracted into CELL by each algorithm are abbreviations of multi-term cell names, which predominantly end in C for cell/cells. In MEB and WMEB, the extracted abbreviations are also abbreviations for chromatography methods, which resulted in the CELL lexicons drifting into these and other experimental techniques in later iterations. Whereas, in BASILISK, CELL slightly drifted between extracting CELL and CELL LINE.

A large performance difference between the three methods occurs with the DISEASE category. Over the first 100 terms, BASILISK-nostop performs superior to both MEB and WMEB. Unexpectedly, the STOP categories did not improve the precision of DISEASE in MEB, whereas WMEB gained 24%. Over the last 100 terms, both WMEB-nostop and WMEB performed very well on DISEASE and thus there was little room for improvement. In MEB, only one correct term was extracted over the 401-500 terms. On the other hand, DISEASE gained the most with BASILISK over the 401-500 terms, increasing from 69 to 93%. This was the largest performance gain for BASILISK using the STOP categories.

Category	BASILISK		MEB		WMEB	
	NOSTOP	STOP	NOSTOP	STOP	NOSTOP	STOP
ANTIBODY	2	2	27	59	93	94
CELL	73	74	2	42	16	29
CELL LINE	96	97	82	89	87	100
DISEASE	69	93	0	1	97	97
DRUG	80	81	98	76	94	97
FUNCTION	54	51	64	66	73	78
MUTATION	98	98	0	1	62	71
PROTEIN	100	99	91	99	99	100
SYMPTOM	71	77	11	60	7	83
TUMOUR	24	21	5	3	88	85
Average	66.7	69.3	38	49.6	71.6	83.4
Inv Rank	3.3	3.5	2	2.3	3.8	4.4

Table 6.5: MEDLINE individual category results (401-500 terms)

Another unexpected result was obtained with MUTATION. Over the first 100 terms extracted into MUTATION by MEB, an 18% drop in precision was seen when the STOP categories were introduced, and only one correct term was extracted in the last 100 terms. This was unexpected because the AMINO ACID STOP category was created to prevent drift specifically within MUTATION. To investigate this further, I manually inspected the AMINO ACID lexicon extracted by MEB. To my surprise, this lexicon quickly drifted into chemical compounds. Considering this topic change, the AMINO ACID lexicon may be responsible for the increase in precision of the DRUG category (MEB +15% at 1-100). Likewise, DRUG, which initially had a high precision with WMEB-nostop (90%), increased by 9% with WMEB.

Over the 401-500 terms, WMEB-nostop performed poorly on SYMPTOM, with only 7% precision. This is significantly less than BASILISK-nostop (71%). With the incorporation of the STOP categories, WMEB boosted precision up to 83%, which also outperforms both BASILISK (77%) and MEB (60%).

The last semantic category of interest, TUMOUR, had a varied performance across each algorithm. Firstly, for BASILISK and MEB, TUMOUR was the second most poorly extracted lexicon, following ANTIBODY and DISEASE, respectively. Inspection of BASILISK's TUMOUR lexicon, showed that the terms drifted into the CELL and CELL LINE categories, whereas in MEB

Algorithm	1-100	101- 200	201-300	301-400	401-500	1-500
BASILISK-nostop	61.4	50.3	43.0	48.5	46.4	49.0
BASILISK	61.8	51.7	42.5	46.0	46.9	49.8
MEB-nostop	56.9	51.7	38.4	30.6	32.1	41.9
MEB	56.4	44.1	36.1	32.6	35.2	40.9
WMEB-nostop	65.8	52.5	56.2	52.0	45.6	54.0
WMEB	**69.4**	**63.9**	**61.6**	**60.1**	**58.9**	**62.8**

Table 6.6: Performance of bootstrappers on TREC with and without stop categories

the lexicon drifted into the CELL and BODY PART categories. Many of the incorrect terms are considered to be TUMOUR related as they are used in the long names of tumours to specify the cell type or body part involved (e.g. *myeloblastic* and *lymphoblastic* occur in some leukaemia names). In WMEB, the TUMOUR category behaves unlike the other semantic categories — the precision in the later stages is significantly greater than the first 100 extracted terms (37% at 1-100 and 85% at 401-500, $p \leq 0.0001$).

6.3.5 TREC Genomics

Each algorithm's performance for extracting lexicons from the TREC Genomics full-text collection is also compared. In the TREC experiments, the same parameters, categories and STOP categories, and seed sets, that were used in the MEDLINE experiments are used (except the MUTATION seeds — see Table 5.1). The results of these experiments are shown in Table 6.6, 6.7 and 6.8.

On average, WMEB performs consistently better than BASILISK and MEB, however each algorithm is significantly less precise on TREC than on MEDLINE, e.g. 62.8% versus 88.7% for WMEB (Table 6.6). The poor performance on TREC may be attributed to both the variation in language use and topic coverage. In the MEDLINE abstracts, the content is denser and more precise, and thus contexts are likely to be less noisy than those in full text. An example of this noise is evident in the CELL lexicon extracted by MEB — many of the terms refer to the cells of tables, which then drifts into both figure and table references, such as *cell1D* and *4C*. The second property of the TREC documents leading to poor performance may be the lack of adequate coverage of the semantic categories of interest, and/or the introduction of new semantic

Category	BASILISK		MEB		WMEB	
	NOSTOP	STOP	NOSTOP	STOP	NOSTOP	STOP
ANTIBODY	31	32	58	56	46	31
CELL	79	85	76	75	65	94
CELL LINE	94	96	58	91	96	94
DISEASE	75	80	68	61	64	79
DRUG	52	53	68	66	74	74
FUNCTION	64	65	66	75	82	80
MUTATION	63	61	37	7	68	75
PROTEIN	53	42	96	100	100	99
SYMPTOM	61	63	20	20	38	48
TUMOUR	42	41	22	13	25	20
Average	61.4	61.8	56.9	56.4	65.8	69.4
Inv Rank	4.0	4.0	3.4	3.5	4.1	4.2

Table 6.7: TREC individual category results (1-100 terms)

categories rarely present in abstracts, which are semantically related to the categories of interest. For example, the ANTIBODY lexicons drifted into the names of biotechnology companies that provide antibodies to research laboratories. These company names occur frequently within the methods sections, to describe the individual antibodies used in detail. Many individual ANTIBODIES can be extracted by the same patterns as the companies, and as a company's name may be mentioned more frequently than an individual antibody, it is more likely to be retrieved, especially by MEB and WMEB. This new semantic drift in ANTIBODY from TREC, results in WMEB's largest performance difference between MEDLINE and TREC (94% on MEDLINE and 14% on TREC at 401-500).

The introduction of the STOP categories had little effect on the performance of both BASILISK and MEB on the TREC data, while WMEB improved significantly (+8.8% at 1-500 terms). Many of the categories that drifted into these STOP categories in MEDLINE, drifted into other categories. For example, on MEDLINE the DISEASE lexicon extracted by WMEB-nostop often drifted into BODY PARTS, which was prevented by the addition of this STOP category. However, on TREC the DISEASE lexicon drifted from viruses that cause disease to experimental viruses and then plasmids and vectors, which also occur more frequently in the methods sections of the articles. As a result, the DISEASE lexicon does not improve when the STOP category BODY PART is added, and its precision actually decreases (WMEB-nostop 75% vs WMEB 64% at

Category	BASILISK		MEB		WMEB	
	NOSTOP	STOP	NOSTOP	STOP	NOSTOP	STOP
ANTIBODY	6	7	2	6	49	14
CELL	53	72	12	0	27	61
CELL LINE	92	96	77	75	83	90
DISEASE	65	76	40	56	75	64
DRUG	62	26	22	53	39	73
FUNCTION	21	19	59	55	64	63
MUTATION	36	12	2	0	1	82
PROTEIN	65	85	96	97	100	98
SYMPTOM	54	62	6	9	11	36
TUMOUR	10	14	1	1	7	8
Average	46.4	46.9	31.7	35.2	45.6	58.9
Inv Rank	2.4	2.8	1.5	1.8	1.8	3.4

Table 6.8: TREC individual category results (401-500 terms)

401-500, Table 6.8). On TREC, the STOP categories also unexpectedly decreased the precision of the ANTIBODY lexicon extracted by WMEB (Table 6.7 and 6.8). On MEDLINE, ANTIBODY was not affected by the STOP categories (Table 6.4 and 6.5). Clearly, STOP categories are domain and even corpus specific.

6.3.6 Pattern Evaluation

In this section, the quality of the extraction patterns selected by each algorithm is considered. Previous evaluations of bootstrapping algorithms have only focused on the lexicons or relationships identified. The pattern evaluation compares the first 100 patterns selected by each algorithm for each semantic category on MEDLINE, and involves manually judging each pattern as either a *true match* (TM), *multi-match* (MM) or *non-match* (NM). This evaluation methodology is discussed in Section 5.7. Table 6.9 shows the distribution of these judgments to the patterns extracted by BASILISK, MEB and WMEB.

Overall, the patterns selected by the algorithms are predominantly true and multi-matches. WMEB selects more true matches and fewer non-matches than BASILISK and MEB, and MEB selects the most non-matching patterns. BASILISK, which performed poorly when extracting ANTIBODY terms, also identified many non-matching (29) patterns (ANTIBODY 53% at 1-100

Category	BASILISK			MEB			WMEB		
	TM	MM	NM	TM	MM	NM	TM	MM	NM
ANTIBODY	63	8	29	97	0	3	100	0	0
CELL	2	98	0	1	68	31	2	84	14
CELL LINE	1	99	0	79	21	0	78	22	0
DISEASE	80	15	5	5	81	14	95	5	0
DRUG	80	17	3	82	16	2	78	17	5
FUNCTION	62	33	5	10	42	48	49	50	1
MUTATION	3	27	70	3	26	71	9	91	0
PROTEIN	98	1	1	54	0	46	99	0	1
SYMPTOM	93	6	1	90	5	5	12	54	34
TUMOUR	4	94	2	0	81	19	2	67	31
Average	48.6	39.8	11.6	42.1	34.0	19.1	52.4	39.0	8.6

Table 6.9: Judgements of the first 100 patterns extracted from MEDLINE

in Table 6.4). On the other hand, WMEB which extracted ANTIBODY terms with almost perfect precision had no non-matching patterns (ANTIBODY 96% at 1-100).

Each algorithm extracted a large number multi-match patterns for CELL. These patterns also semantically identify BODY PART, however they are rarely capable of extracting BODY PART terms due to the competition with the BODY PART category. As a result, BASILISK performs strongly on CELL. On the other hand, both MEB and WMEB are less precise than BASILISK, and this is due to the large number of non-matching patterns, which do lead to semantic drift. For example, most of the non-matching CELL patterns identify abbreviations of chromatography methods.

Although each algorithm performs well on CELL LINE, many of the extracted patterns are multi-matches, especially for BASILISK. In BASILISK, the multi-match patterns typically also match the TUMOUR and CELL categories, however little drift into these categories is evident. The reverse occurs for the TUMOUR multi-match patterns, which match CELL LINE terms, and these multi-matches lead to semantic drift within the TUMOUR lexicon in BASILISK.

Although BASILISK extracts MUTATION terms with high precision (90% at 1-100, Table 6.4), the MUTATION patterns extracted by BASILISK were predominantly non-matching (70/100) and most likely to extract PROTEIN terms. Further, the large proportion of MUTATION multi-match patterns (27/100) were also considered to identify PROTEINS. Therefore, for

MUTATION the term selection process in BASILISK, which prefers infrequent terms, is working sufficiently well. This bias towards infrequent terms is effective for MUTATION, as they occur much less frequently than terms from other categories. In contrast, the large number of MUTATION non-matching patterns extracted by MEB (71/100) resulted in poor lexicon precision (70% at 1-100, Table 6.4). This is because MEB's term selection phase prefers terms that match the most patterns. By forcing the categories' patterns to be mutually exclusive in MEB and WMEB, the potential drift caused by the large number of multi-match patterns is reduced.

6.4 Future Work

This thesis investigates the extraction of single-word entities only. Multi-word expressions and entities are used frequently within newswire, web text and the biomedical literature. Extracting these will increase the vocabulary within semantic resources, and in turn their coverage of the domain. Therefore, an important next step in bootstrapping semantic lexicons is to expand the set of candidate terms to include multi-word expressions. Multi-word expressions are less ambiguous than single-word entities, and thus the mutual exclusion property between the semantic categories is more likely to hold. Based on the multi-word experiments performed on MEB by Murphy and Curran (2007), this should be a relatively simple extension to carry out on WMEB, which will significantly improve the coverage of the extracted lexicons.

The patterns used in the experiments by BASILISK, MEB, and WMEB have been restricted to those extracted from 5-grams and are composed of the two words either side of a possible candidate term. It is plausible that different semantic categories are extracted more effectively using different pattern context geometries. Therefore, it would be interesting to explore various pattern context geometries and utilise combinations of these.

One main advantage of using window-based patterns is that the bootstrappers do not rely on additional tools, such as parsers or chunkers, making them particularly useful for domains without these tools. More sophisticated patterns incorporating syntactic information, have been exploited by other systems (e.g. Thelen (2001), Pantel et al. (2004) and Paşca et al. (2006)). It would also be interesting to experiment with these types of patterns with WMEB.

The bootstrapping algorithms developed for identifying tuples of related terms, such as Snowball (Agichtein and Gravano, 2000), are currently based on the single-category bootstrapping framework — that is, they aim to extract one relation type between two semantic categories. The patterns used in these methods tend to be very restrictive. In Paşca et al.'s (2006) bootstrapper, patterns containing terms indicative of the relation extracted are ranked higher. For example, patterns containing the term *capital* when extracting *City-CapitalOf-Country* tuples are preferred. This may be to reduce the two categories of interest drifting, and in turn extracting incorrect relation pairs, such as *rivers* and their locations. It remains to be explored whether extracting multiple types of relation tuples simultaneously in a multi-relation framework can be used to prevent this and allow less restrictive patterns to be used. For WMEB to be most effective, the semantic categories within the different relation types will need to be opposing. For example, the *City-CapitalOf-Country* tuples may be extracted simultaneously with the *River-LocatedIn-Country* tuples.

Other interesting areas that may also benefit from WMEB, or the multi-category framework in general, are that of automatic attribute extraction of semantic categories (Durme et al., 2008) and concept and instance extraction (Hovy et al., 2009). The semantic resources extracted by these methods are also very valuable.

6.5 Summary

This chapter presents a new bootstrapping algorithm, *Weighted Mutual Exclusion Bootstrapping* (WMEB), for extracting high precision biomedical lexicons, and the patterns that identify them, from raw text. WMEB extracts the terms and patterns of multiple semantic categories simultaneously, based on the assumption of mutual exclusion and the framework of MEB. To reduce semantic drift in WMEB, a new term and pattern relevance weighting scheme was introduced that promotes terms and patterns that are most strongly associated with each other. I also proposed the cumulative pattern pool to keep all selected patterns active throughout bootstrapping.

This chapter presented a detailed evaluation of three different algorithms for extracting biomedical semantic lexicons — WMEB, MEB and BASILISK. My initial analysis has shown that WMEB's candidate relevance weighting and cumulative pattern pool combine effectively to reduce the semantic drift that still dominates in MEB. The analysis of the different relevance weighting metrics, showed that the χ^2 measure for independence is more effective than the Pointwise Mutual Information measures tested, and the naïve productivity measure used in MEB. As a result, WMEB outperforms MEB by a significant margin. This suggests that my intuition regarding MEB's naïve scoring is correct.

I have also demonstrated that WMEB significantly outperforms BASILISK, which is also more effective than MEB. The individual category analysis identified particular sources of semantic drift within each algorithm. The lexicons identified by both WMEB and MEB are prone to drifting into unspecified semantically-related categories, as the lexicons and patterns of the categories of interest are mutually exclusive. On the other hand, BASILISK typically drifts between the categories of interest and identifies infrequent terms. As a result, BASILISK improves less than WMEB and MEB with the STOP categories.

Although, WMEB significantly outperforms the previous approaches, semantic drift still occurs in the later iterations. The suggested approach to overcome this, by Curran et al. (2007) and Lin et al. (2003b) to introduce more STOP categories, is inappropriate. Firstly, it is difficult to identify new STOP categories that will work effectively for each algorithm. Secondly, this method only prevents drift into the specified categories, and thus drift is then likely to occur into unspecified ones. This issue has lead to my proposal of utilising *bagging* to correct semantic drift in the extracted lexicons. This approach is the focus of the following chapter, and is shown to significantly improve the quality of the lexicons extracted. In Chapter 8, I present another technique that reduces semantic drift significantly during the bootstrapping process by exploiting *distributional similarity* measures.

Algorithm 1 Weighted Mutual Exclusion Bootstrapping

Input : Seed word lists S_c $\forall$ categories c
Input : Raw patterns $\mathcal{P}$ and terms $\mathcal{T}$
Input : # terms N_T and patterns N_P added per iteration
Output : Lexicon L_c and patterns Pool_c lists $\forall$ category c
$L_c \longleftarrow S_c \forall$ categories c
for each iteration **do**
  Phase I - Select Candidate Patterns
  for each $category\, c \in \mathcal{C}$ **do**
    for each $pattern\, p \in \mathcal{P}$ **do**
      reliability = number of times p matches *term* $t \in L_c$
      if *reliability* > 0 **then**
        compute *relevance weight* of p
        $candidate_P \longleftarrow p$
  for each $category\, c \in \mathcal{C}$ **do**
    for each $pattern\, p \in candidate_P$ **do**
      if p matches terms from multiple categories **then**
        discard p
    sort $candidate_P$ in descending order of their *reliability* and *relevance weight*
    $x = 0$
    while $x <= N_P$ **do**
      pattern p = $candidate_P[y]$
      $\text{Pool}_c \longleftarrow p$
      $x+ = 1$
    if no new pattern was added to Pool_c **then**
      $n =$ next top pattern not $\in \text{Pool}_c$
      $\text{Pool}_c \longleftarrow n$
  Phase II - Select Lexicon Terms
  for each $category\, c \in \mathcal{C}$ **do**
    for each $term\, t \in \mathcal{T}$ **do**
      reliability = number of times t matches *pattern* $p \in \text{Pool}_c$
      if *reliability* > 0 **then**
        compute *relevance weight* of t
        $candidate_T \longleftarrow t$
  for each $category\, c \in \mathcal{C}$ **do**
    for each $term\, t \in candidate_T$ **do**
      if t matches patterns from multiple categories **then**
        discard t
    sort $candidate_T$ in descending order of their *reliability* and *relevance weight*
    $x = 0$
    while $x <= N_T$ **do**
      $termt = candidate_T[y]$
      $L_c \longleftarrow t$
      $x+ = 1$

Chapter 7

Random Seeds and Bagging

This chapter demonstrates a fundamental flaw in the evaluation paradigm used to compare and analyse bootstrapping algorithms. This standard approach, which was used in the previous chapter, evaluates an algorithm with only one set of seed terms. However, I will demonstrate that the algorithms vary greatly with different seed sets and so it is not possible to reliably compare the performance or behavior of these algorithms using this method.

Section 7.2 presents a superior evaluation methodology which utilises multiple sets of random seeds. This approach allows us to examine the impact of different seeds and compare the algorithms more robustly and reliably. The experiments in Section 7.2.1 use this approach to show that BASILISK is significantly more sensitive to the seed sets than WMEB and MEB. In this evaluation, WMEB is still shown to outperform MEB and BASILISK. However, semantic drift is still prominent for many categories, preventing the extraction of large yet precise lexicons.

My random seed experiments identify previously unreported variations in the terms extracted by BASILISK, MEB and WMEB using different seeds. This insight lead to my hypothesis that semantic drift within the lexicons could be reduced by using an ensemble of randomly initialised bootstrappers, commonly known as bagging.

Section 7.3 introduces the framework for bagging a bootstrapper. I present two types of bootstrapper bagging: the first follows the traditional approach for *supervised* ensembles, and uses gold seed terms to initiate the bootstrappers; and a novel *unsupervised bagging* approach, which requires no more gold seeds than the initial hand-picked seeds. These techniques effectively correct semantic drift within the lexicons, significantly improving performance. Most of the work in this chapter was described in McIntosh and Curran (2009a).

7.1 Unreliable Evaluation

The first step in bootstrapping lexicons is to select a set of unambiguous seeds by hand for each category of interest. These hand-picked seeds are typically selected by a domain expert who, based on their knowledge of the field, tries to find a set that will be an unambiguous representative sample to initialise a bootstrapper. In each of the bootstrapping experiments so far, the algorithms have been initialised with five hand-picked seeds for each category. These seeds, shown in Table 5.1, were carefully selected based on my background and intuition in biomedicine.

The literature offers very little advice on the selection of good seeds, and it is unlikely that domain experts will agree on representative seeds or be able to tailor seeds for specific bootstrapping algorithms. Indeed, the hand-picked seeds are a very small subset of the possible unambiguous terms for each category. To improve the seeds selected, the frequency of the potential seeds in the corpora is often considered, on the assumption that highly frequent seeds will be more effective (Thelen and Riloff, 2002; Ando, 2004). For example, in Thelen and Riloff (2002), the seeds for BASILISK were selected by first sorting the words in the corpus by their frequency, and then by manually identifying the 10 most frequent nouns belonging to each semantic category. And Meij and Katrenko (2007) selected the most frequent terms identified using hyponym patterns introduced in Hearst (1992) like:

<term> *is a* [NAMED ENTITY],

where the wildcard <term> stands for instances of the named entity type. Unfortunately, highly frequent seeds may be too general and extract many non-specific patterns.

Another approach, known as *strapping* was devised by Eisner and Karakos (2005) to identify useful seeds for word sense disambiguation. In strapping, multiple semi-supervised bootstrapping instances are used to train a meta-classifier, which given a bootstrapping instance can predict the usefulness (*fertility*) of its seeds. The most fertile seeds can then be used in place of hand-picked seeds. The design of a strapping algorithm is more complex than that of a supervised learner (Eisner and Karakos, 2005), and it is unclear how well strapping will generalise to other bootstrapping tasks.

Bootstrapping algorithms are typically evaluated by measuring the quality of the lexicons extracted using a single set of seeds for each category of interest. For example, almost all of the algorithms discussed in the previous chapters were evaluated using only one set of seeds (see Thelen and Riloff, 2002; Curran et al., 2007; Paşca, 2007).

This approach is unreliable and does not provide a fair comparison. Even evaluating on multiple categories does not ensure the robustness of the evaluation. Firstly, it is possible that the strategy for selecting seeds can perform well on one algorithm and poorly on another, and thus using a single seed set for comparison is unreliable. Secondly, it provides no insight into how sensitive an algorithm is to different seed sets. In the following section, I will describe my own evaluation methodology to address these issues, which is based on seeding each bootstrapping algorithm multiple times with random gold seeds.

7.2 Random Gold Seeds

Rather than comparing algorithms using the standard evaluation paradigm, with only one set of hand-picked seeds for each category, this evaluation involves initialising each algorithm 10 times with different sets of random gold seeds for each category. Two different random sets of gold seeds were sampled from two separate sets of correct terms extracted from the evaluation cache (see Section 5.4). 1) UNION: the correct terms extracted by any of BASILISK,

WMEB and MEB; and 2) UNIQUE: the correct terms uniquely identified by only one algorithm. The UNIQUE seeds are different for each algorithm and the motivation for using these is to determine if an algorithm performs well with seed terms that it uniquely extracts.

Sampling seeds from the evaluation cache eliminates the need for a user to choose many different sets of new seeds and introduces the seed diversity required. The degree of ambiguity of each gold seed is unknown and term frequency is not considered during the random selection process. Note that the 10 sets of UNION seeds are identical for each algorithm, and the UNIQUE seeds are different for each algorithm. Evaluating multiple sets of results makes it possible to compare algorithms more reliably. However, this approach also requires 10 times more manual evaluations to be performed, which is a very time consuming task.

Using this evaluation methodology, we can analyse each bootstrapping algorithms' sensitivity to the initial seeds and the degree of variability of the lexicons generated. The average precision over the 10 bootstrapping runs (average-10) and the corresponding standard deviation can be calculated, as well as the degree of overlap between the different lexicons extracted for each category using the different seeds. An algorithm with a higher standard deviation and/or a small degree of lexicon overlap than another algorithm is considered to be more sensitive to the seed terms. We will also use the average-10 precision of the algorithms to compare their performance more reliably.

7.2.1 Results

Firstly, I investigated the term variability of the lexicons extracted by each algorithm using UNION seeds. For each algorithm, the degree of overlap between the top 100 and top 500 terms in each of the 10 lexicon sets was calculated. These figures are shown in Table 7.1. BASILISK had the smallest degree of overlap across the lexicons, with less than 20% overlap over the first 100 terms. In comparison, both MEB and WMEB produced less varied lexicons. Clearly BASILISK is far more sensitive to the choice of seeds. Unfortunately, this also makes the evaluation cache a lot less valuable for the manual evaluation of BASILISK.

Algorithm	1-100	1-500
BASILISK	18.2	38.5
MEB	33.9	41.7
WMEB	44.3	46.8

Table 7.1: Degree of overlap between the lexicons extracted using UNION random seeds

Algorithm	BASILISK	MEB	WMEB
BASILISK	100.0		
MEB	2.6	100.0	
WMEB	3.9	44.6	100.0

Table 7.2: Degree of overlap between the lexicons extracted by each algorithm using UNION random seeds (1-100 terms)

In Table 7.2, the degree of overlap between each of the algorithms over the first 100 terms is shown. BASILISK rarely extracts terms identified by either MEB or WMEB and, as expected, WMEB and MEB are most similar with 44.6% overlap. Both of these overlap results match the annotators' intuitions that BASILISK retrieved far more of the esoteric, rare and misspelt terms, than MEB and WMEB.

Our next analysis evaluates the performance variation of each algorithm initialised with different gold seed sets. The plots in Figure 7.1 show the variation in precision between WMEB and BASILISK, and WMEB and MEB, with the 10 gold seed sets from UNION. Precision is measured on the first 100 terms and is averaged over the 10 categories. Even with only the top 100 terms, variations in precisions are obvious. The performance achieved using the hand-picked seeds is marked with a square, and each algorithm's average-10 precision, calculated over the 10 random seed sets, with 1 standard deviation (s.d.) error bars are shown. Note that the axes start at 50% precision. Visually, the scatter is quite obvious and the deviations are quite large. Note that on the hand-picked seed evaluation, BASILISK outperformed each of the 10 random seed sets.

A linear regression analysis was performed to identify any correlation between the algorithm's performances. For highly correlated experiments the Pearson's coefficient of regression (R^2) approaches 1, and for un-correlated data 0. The resulting regression lines are shown in Figure 7.1a and 7.1b. The regression analysis identified no correlation between WMEB and BASILISK ($R^2 = 0.13$), or WMEB and MEB ($R^2 = 0.11$). Therefore, it is almost impossible to

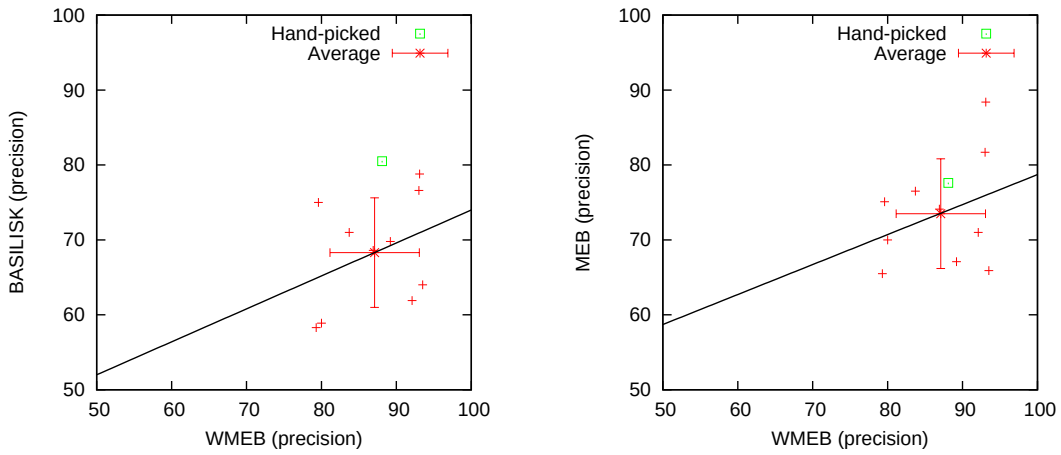


Figure 7.1: Performance relationship between WMEB and BASILISK (**a**), and WMEB and MEB (**b**), on UNION and hand-picked seeds (1-100 terms)

predict the performance of one bootstrapping algorithm with a given set of seeds from another's performance, and thus comparisons using only one seed set are unreliable.

Table 7.3 summarises the results on the 10 different seed sets sampled from UNION and UNIQUE. The average-10 precision is shown, as well as the minimum and maximum average precision. The hand-picked seeds on BASILISK performed much better than average, whereas WMEB performed close to its average-10 precision. That is, BASILISK has been advantaged in the previous experiments. MEB performed better on average than BASILISK, even though it extracted less precise lexicons with the hand-picked seeds, and WMEB significantly outperformed BASILISK on average. Note that the maximum precision value for BASILISK (78.8%) is less than the minimum precision value for WMEB (79.3%). This evaluation demonstrates the difficulty of picking the best seeds for an algorithm, and that comparing algorithms with only one set has the potential to favor or penalise an algorithm substantially.

In the UNIQUE gold seed experiments, I hypothesized that each algorithm would perform well on its own seed set. This was the case for WMEB and MEB, both of which performed better on average than with the UNION seeds. The s.d. for these algorithms was also much smaller. Interestingly, BASILISK performed significantly worse than WMEB and MEB, with a high s.d. of 9.75. As these seeds are selected directly from the lexicons extracted by BASILISK

Algorithm	Hand-picked	Average-10	Min.	Max.	s.d.
	GOLD UNION SEEDS				
BASILISK	80.5	68.3	58.3	78.8	7.31
MEB	77.6	73.5	65.5	88.4	7.32
WMEB	88.1	87.1	79.3	93.5	5.97
	GOLD UNIQUE SEEDS				
BASILISK	80.5	67.1	56.7	83.5	9.75
MEB	77.6	76.0	69.6	82.6	4.59
WMEB	88.1	91.6	82.4	95.4	3.71

Table 7.3: Variation in precision with random gold seed sets (1-100 terms)

with the hand-picked seeds, and that BASILISK prefers low frequency terms, BASILISK's poor performance may be a direct result of these terms not forming good seed sets.

One of the random BASILISK UNIQUE samples performed noticeably poorly with an average precision of 56.7%. This seed sample was evaluated on WMEB and MEB, and both methods resulted in significantly higher average precisions than BASILISK (+71.5% for WMEB, and +70.9% for MEB, $p \leq 0.001$). These seeds are obviously not ideal for WMEB or MEB either. However, this indicates that the performance of WMEB and MEB is more robust with infrequent seeds than BASILISK, and most importantly, seeds that perform well on one algorithm are not necessarily the best for another.

These experiments have demonstrated that the standard evaluation paradigm, using one set of seeds over a few categories, does not provide a robust and informative basis for comparing bootstrapping algorithms. I have shown that randomly selected gold seeds can outperform carefully picked seeds, and that a seed set can perform above or below average on different algorithms. The random seed evaluation has shown that BASILISK is very sensitive to the initial seed set, and that WMEB is more reliable and effective than both BASILISK and MEB for extracting biomedical semantic lexicons.

7.3 Ensembles and Bagging

The concepts of *ensemble learning* and *bagging* were originally developed for the purpose of improving upon stand-alone supervised classification models. A supervised classifier generates a model of the classification task using training data, which can be used to make predictions for new examples. *Ensemble* methods are combinations of different base classifier models produced by one or more supervised classification algorithms, whose individual predictions are combined to form a final classification (Dietterich, 2000). There are lots of strategies for assigning the final classification. For example, the final classification may be the majority predicted class among the models, or if the base models provide class probability estimates, the class with the largest average probability estimate.

Within the NLP community, ensembles of classifiers have become a common architecture in problems that are naturally classification problems, such as named entity recognition (GuoDong et al., 2004) and word sense disambiguation (WSD) (Florian et al., 2002). The main ensemble methodology employed involves utilising multiple classification algorithms to generate *different* classification models based on the same training sets and features. This is based on the intuition that individual classification algorithms have different and complementary strengths and weaknesses (learning biases) and can perform well on different instances, and thus an ensemble can aggregate their strengths and improve performance. For example, Yarowsky et al. (2001) combined four diverse supervised algorithms (bag-of-word naïve Bayes, feature-enhanced naïve Bayes, Cosine-vector-based model, and Interpolated Decision Lists (Yarowsky, 2000)) for WSD. Further, Brody et al. (2006) demonstrated the potential of using unsupervised algorithms within ensembles. Their unsupervised ensembles outperformed the best individual unsupervised components on WSD challenges.

Another popular ensemble approach is known as *bagging* (Breiman, 1996). In the bagging algorithm, multiple samples of the complete training set are used to generate multiple base classifiers of the same type. Each training sample is randomly selected uniformly from the original training set. The ensemble of base models then classifies a new example by returning the class that the majority of base classifiers predict for the example.

Breiman (1996) demonstrates that the bagging ensemble will likely improve overall performance if the base models perform well on new examples, and if the different bagging training sets generate base models that produce different predictions. On the other hand, if bagging is applied to base models, which almost always predict identical classes, then the bagged prediction will predominantly be the same as the base models, and in turn lead to almost no improvement over individual models. Therefore, bagging is more useful when an unstable learning algorithm (one that undergoes major changes in response to small changes in the training set) is used as the base model.

While the wide variation reported in the previous section is an impediment to reliable evaluation, it presents an opportunity to improve the precision of the lexicons extracted by bootstrapping. It has been demonstrated that the bootstrapping algorithms are sufficiently unstable — different seed sets lead to the extraction of diverse lexicons — which would suggest that bagging could be used to reduce semantic drift. This is the first application of bagging to the bootstrapping framework.

7.4 Framework for Bootstrapper Bagging

In my bagging framework, a bootstrapping algorithm is initialised $n = 50$ times with random seed sets for each semantic category. Each of these n bootstrappers generates a set of lexicons for each category, resulting in n new lexicons $L_1, L_2, \ldots, L_n$ for each category. The next phase involves aggregating a category's extracted terms in L_{1-n} to form the category's final lexicon, using a voting function. This process is depicted in Figure 7.2, and is similar to the approach introduced by Breiman (1996).

The voting function is based on two related hypotheses of terms in highly accurate lexicons:

1. the more category lexicons in L_{1-n} a term appears in, the more likely the term is a member of the category;

2. terms ranked higher in lexicons are more reliable category members.

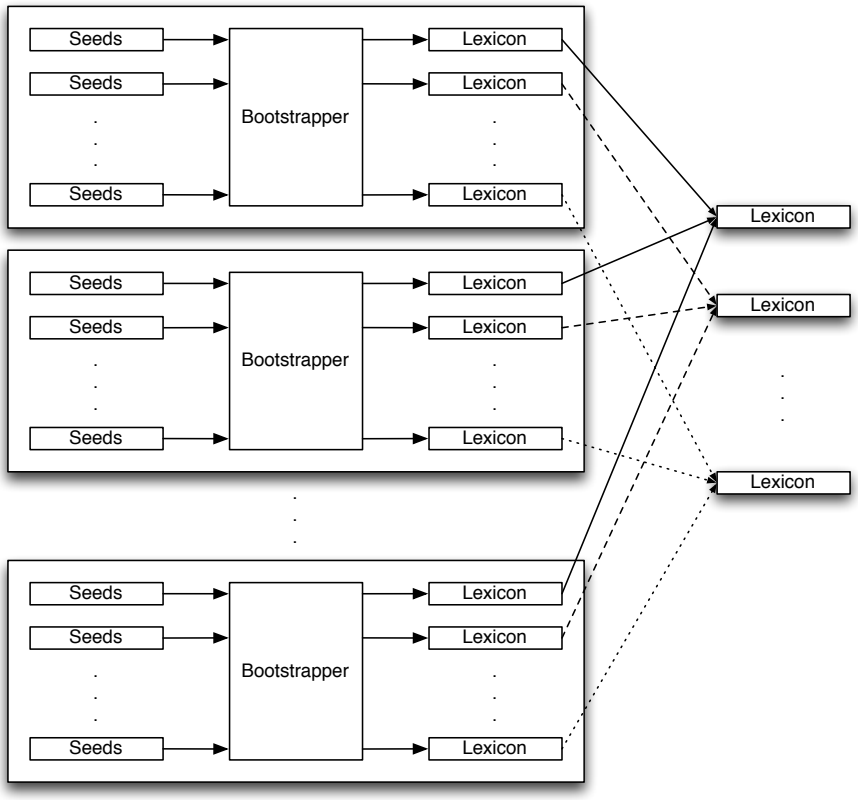


Figure 7.2: Framework for bootstrapper bagging

Based on these hypotheses, the aggregated terms in each category's lexicon are ranked by the number of different lexicons they appear in, and to break ties, the term that has the largest inverse rank sum across the lexicons it appears in is taken. Terms not appearing in a lexicon are assigned an inverse rank of 0. This prefers terms that are extracted more frequently in the earlier iterations.

7.4.1 Supervised Bagging

The first step in bootstrapper bagging is to decide how the sets of random seeds will be selected. In the traditional *supervised* approach, the seed sets for a category must only contain terms which are known to be category members. These seed terms are referred to as correct or gold seeds. The set of possible gold terms for sampling corresponds to the UNION set of the correct terms extracted by BASILISK, MEB and WMEB, using the hand-picked seeds.

For each of the 50 individual bootstrapping instances, a set of 5 gold terms for each category are randomly selected uniformly without replacement. Therefore, multiple bootstrapping instances can be assigned the same seeds. The frequency of the terms within MEDLINE were not considered during sampling, nor the possibility of the seed terms belonging to multiple semantic categories of interest.

7.4.2 Unsupervised Bagging

A significant problem for supervised bagging approaches is that they require a larger set of gold-standard seed terms to sample from — either an existing gazetteer or a large hand-picked set. In my experiments, I used the evaluation cache which took considerable time to accumulate. This negates one of the main advantages of bootstrapping, the quick construction of accurate semantic lexicons, with a chicken-and-egg problem. To overcome this, I propose a novel solution — unsupervised bagging.

In the unsupervised bagging approach, a bootstrapping algorithm is initialised 50 times with unsupervised random seeds. As both BASILISK and WMEB have very high average precision over the first 500 terms (and MEB also over the first 100 terms) extracted using the hand-picked seeds, I hypothesise that the extracted lexicons will provide an acceptable source of seed terms and introduce diversity with little noise. The unsupervised seeds are sampled directly, without correctness checks, from each lexicon extracted by the specific bootstrapper using the original set of hand-picked seeds.

The unsupervised bagging process now uses two rounds of bootstrapping. First, each category's lexicon is generated by a bootstrapper using the original hand-picked seeds ($\mathrm{L_{hand}}$). The next round's 50 set of seeds are randomly sampled from $\mathrm{L_{hand}}$. These unsupervised seeds are then used to instantiate the bagging process, generating 50 lexicons for each category. The second round is followed by aggregating the 50 resulting lexicons for each category. This unsupervised approach now only requires the original 50 hand-picked seed terms across the 10 semantic categories, rather than the maximum 2500 (50×50) gold terms sampled from a lexicon (in the supervised bagging experiments, 2100 different gold terms were randomly selected).

I explore two different approaches for sampling the unsupervised seeds from L_{hand}. The first method performs uniform random sampling from restricted sections of L_{hand}. Random sampling is performed for each category from the top 100, 200 and 500 terms of L_{hand}. The unsupervised seed sets from the smaller samples will have higher precision, as these terms are extracted in the early bootstrapping iterations before semantic drift dominates. These seeds are thus likely to generate precise lexicons. On the other hand, the unsupervised seeds from the smaller samples will also be less diverse as individual terms are more likely to be sampled multiple times. As a result, their generated lexicons are also likely to be less diverse.

In a truly unsupervised approach, it is impossible to know if and when semantic drift occurs. The second sampling method, aims to increase seed diversity while reducing the possibility of sampling incorrect terms. This method involves sampling from the top 500 terms according to a varying probability density function (PDF) using rejection sampling (Ripley, 1987).

In rejection sampling, each term is assigned a probability according to a PDF. Then two random selections are performed. The first selects a term from the lexicon with *uniform* probability, that may be *accepted* or *rejected* as a seed. The second selects a random number from a uniform distribution on [0,1). If the second number is less than the assigned probability of the selected term, than the term is accepted. Otherwise, the term is rejected.

The PDF considered is related to the inverse rank of a term in the lexicon (see Equation 5.3) and is shown in Equation 7.1, where r is the rank of the term. This PDF assigns higher probabilities to terms which are ranked higher in the lexicon.

$$\text{PDF}(r) = \frac{\sum_{i=r}^{n} i^{-1}}{\sum_{i=1}^{n}\sum_{j=i}^{n} j^{-1}} \tag{7.1}$$

7.5 Results

This section presents an evaluation of the performance of the supervised and unsupervised bagging approaches on BASILISK, MEB and WMEB. In each experiment, a bootstrapper is initialised 50 times with different seed sets for the 10 biomedical semantic categories of interest. Each bootstrapper is run for 100 iterations, and the maximum number of terms and patterns that can be added in each iteration is 5.[1]

The seeds for the STOP categories remain the same for each run (see Table 5.2). The supervised seeds are sampled from the UNION sets, and have an average degree of overlap of 2.7%. The same sets of supervised seeds are then used when bagging each algorithm. In the unsupervised experiments, the seed sets are sampled from each algorithm's lexicons extracted using the hand-picked seeds, and therefore each bootstrapper is likely to be initialised with different seed sets.

7.5.1 Supervised Bagging

Table 7.4 compares the average precision of BASILISK, MEB and WMEB for extracting large lexicons using the original hand-picked seeds and supervised bagging. Supervised bagging significantly improved the precision of the lexicons extracted by each algorithm over the first 200 terms ($p \leq 0.0001$). In particular, MEB's precision improved by 13.5%. The improvements continued for both MEB and BASILISK, with the supervised bagging significantly outperforming the hand-picked seeds over 1000 terms (MEB +19%; BASILISK +12.2%, $p \leq 0.0001$). Whereas for WMEB, the supervised bagging resulted in fewer precise terms in the later segments (201-1000). Overall, supervised bagging of WMEB was slightly less effective than a single instance of WMEB with only the hand-picked seeds. Despite this, WMEB, with and without supervised bagging, significantly outperformed BASILISK and MEB with bagging.

[1] The number of patterns is incremented by 1 in each iteration of BASILISK.

Algorithm	1-200	201-400	401-600	601-800	801-1000	1-1000
			Hand-picked seeds			
BASILISK	76.3	67.9	67.8	63.1	58.3	66.7
MEB	71.6	52.3	46.3	38.4	40.4	49.8
WMEB	90.3	89.8	82.3	68.7	62.0	78.6
			Supervised bagging			
BASILISK	83.9	80.1	79.1	†74.7	†70.0	†77.5
MEB	85.1	71.3	64.1	54.9	48.0	64.7
WMEB	94.9	87.0	80.7	68.1	61.3	78.4

Table 7.4: Supervised bagging with UNION gold seed sets. †Bagging BASILISK resulted in only 670 MUTATION terms.

It is likely that both MEB and BASILISK benefited significantly from supervised bagging as they generated more diverse lexicons with different seeds. The average overlap between the 50 lexicon sets for MEB and BASILISK are 38.0% and 32.9%, respectively. In comparison, WMEB's lexicons had a high overlap of 50.5%. The overlap difference is even larger when measured over the first 100 extracted terms: BASILISK 11.2%, MEB 33.9% and WMEB 44.2%. Therefore, when aggregating the lexicons extracted by MEB and BASILISK, there is more variability to exploit, and the chance of a random co-occurrence is lower and so when terms co-occur they are more significant, than with WMEB. In fact, 69.0% of the terms identified by supervised bagging of BASILISK were not identified with the hand-picked seeds. For WMEB, only 39.1% of the bagged lexicons' terms are new. These observations are related to Breiman's (1996), who noted that performance improves when ensembles of different base models are used.

For BASILISK, the supervised bagging of the 50 MUTATION lexicons only resulted in a combined lexicon with 670 different terms. That is, *all* terms within the 50 lexicons are present in the final MUTATION lexicon. This results from a very large overlap between the 50 MUTATION lexicons (97.1%). This overlap is unexpected, as BASILISK typically generates more diverse lexicons and the MUTATION seed sets had an overlap of 2.7%. In comparison, WMEB and MEB with the same seeds generated MUTATION lexicons with 57.6% and 27.9% overlap, respectively.

Algorithm	1-200	201-400	401-600	601-800	801-1000	1-1000
			Top-100 Unsupervised bagging			
BASILISK	71.3	67.6	63.4	65.4	61.6	65.9
MEB	73.3	62.3	56.4	49.9	47.4	57.9
WMEB	89.8	84.8	78.8	74.0	68.2	79.1
			Top-200 Unsupervised bagging			
BASILISK	69.6	68.7	62.7	48.6	46.9	59.3
MEB	69.9	57.6	52.8	50.0	44.8	55.0
WMEB	90.7	84.6	78.4	69.1	61.5	76.8
			Top-500 Unsupervised bagging			
BASILISK	64.1	60.2	60.0	†51.0	†50.7	†57.2
MEB	65.9	56.8	51.6	46.5	42.0	52.5
WMEB	92.7	85.3	81.7	69.2	61.1	78.0
			PDF-500 Unsupervised bagging			
BASILISK	69.4	68.3	66.5	†60.5	†50.6	†63.0
MEB	66.5	58.5	50.1	47.2	45.9	53.6
WMEB	92.5	87.6	80.3	74.3	70.2	81.0

Table 7.5: Unsupervised bagging. †Bagging BASILISK resulted in only 652 (Top-500) and 702 (PDF-500) MUTATION terms.

7.5.2 Unsupervised Bagging

Table 7.5 shows the average precision of the aggregated lexicons for each algorithm with unsupervised bagging. The unsupervised seeds for each algorithm are sampled from the top 100, 200 and 500 terms that were extracted using the hand-picked seeds.

Unsupervised bagging of BASILISK was far less effective than the supervised bagging and BASILISK with only the hand-picked seeds. BASILISK's best performance occurs when the seeds are sampled from the top 100 lexicon terms. BASILISK was hypothesised to perform well on these seeds, as they are very precise and are extracted before semantic drift is prominent. However, unsupervised bagging of BASILISK is not effective, which in part may be attributed to the very infrequent terms extracted by BASILISK, that in turn make very unproductive seeds. This phenomena was also observed in the UNIQUE random gold experiments on BASILISK, where the seeds were sampled from L_{hand} (Table 7.3). When the sampling sizes are increased (200, 500, PDF-500), the aggregated precisions decreased. This further indicates that L_{hand} must be accurate enough in the first place for it to be a reasonable approximation of gold seeds.

Both MEB and WMEB improve significantly with unsupervised bagging using each of the seed sampling sizes. With each seed sample, unsupervised bagging of MEB significantly outperforms MEB with the hand-picked seeds. MEB is more precise with seeds selected from the top 100 and 200 terms. This is not surprising, as the larger samples will contain significantly more incorrect terms due to semantic drift, leading to less precise lexicons for bagging.

Unsupervised bagging of WMEB is effective when seeds are sampled from the top 500 terms. Sampling with PDF-500 results in more accurate lexicons over the first 1000 terms than the other sampling methods for WMEB. In particular, unsupervised bagging of WMEB is more accurate than supervised bagging and the hand-picked seeds (81.0% vs 78.4% and 78.6%). In the critical later stages (801-1000), WMEB PDF-500 improves over supervised bagging and the original hand-picked seeds by more than 8%. WMEB benefits from the larger variability introduced by the more diverse sets of seeds, which were sufficient for generating diverse lexicons with less overlap than in the supervised experiments (44.2%). This greater variability available also out-weighs the potential noise from incorrect seeds.

7.5.3 Individual Categories

For the final bagging evaluation, the performance gains of the best bagging system are investigated by analysing the individual semantic lexicons. Table 7.6 shows the precision of each lexicon extracted by WMEB with the hand-picked seeds, and the final lexicons formed following unsupervised bagging of WMEB with PDF-500 seeds.

The semantic categories that are extracted very precisely (above 95.0 across each measurement; PROTEIN, CELL LINE, and DRUG) by WMEB with the hand-picked seeds, improve slightly with bagging where there is room for improvement. Although the precision increases are slight, over 24% of the terms in their aggregated lexicons were not identified with the hand-picked seeds (new terms: PROTEIN 27.7%, CELL LINE 31.7% and DRUG 24.2%).

The CELL category, which had the highest degree of semantic drift, dropping from 99.0% (1-200) to 0% (801-1000), gained the most from unsupervised bagging (+29.5% at 401-600

Category	Hand-picked seeds			Unsupervised bagging PDF-500		
	1-200	401-600	801-1000	1-200	401-600	801-1000
ANTIBODY	97.0	94.0	40.0	97.5	92.5	79.5
CELL	99.0	37.0	0.0	98.0	66.5	44.0
CELL LINE	99.5	94.5	95.5	100.0	99.0	97.5
DISEASE	91.5	96.0	71.5	99.0	94.5	78.5
DRUG	98.5	96.0	95.0	100.0	99.0	94.0
FUNCTION	83.0	71.0	67.5	83.5	55.0	46.0
MUTATION	86.5	76.5	34.0	92.0	78.0	71.0
PROTEIN	100.0	100.0	99.0	100.0	99.5	99.0
SYMPTOM	95.5	80.0	57.0	90.0	72.0	64.0
TUMOUR	52.5	78.0	61.5	65.0	47.0	28.0
Average	90.3	82.3	62.1	92.5	80.3	70.2
Inv Rank	5.8	4.9	3.7	5.5	4.7	4.2

Table 7.6: Unsupervised bagging of WMEB: MEDLINE individual category results

and +44.0% at 801-1000, $p \leq 0.0001$). Unsupervised bagging introduced approximately 500 new terms to the CELL lexicon. These new terms outweighed the original drifted terms, with respect to their extraction frequency and inverse rank, and in turn prevented the selection of the drifted terms. Similar performance gains are achieved for the ANTIBODY, DISEASE and MUTATION categories, which have a high degree of semantic drift in the later iterations with WMEB (e.g. at 801-1000 terms: ANTIBODY +39.5% and MUTATION +38.0%). Early errors in the DISEASE and MUTATION lexicons were also corrected.

So far the analysis has demonstrated that categories which exhibit little semantic drift initially (1-200: >85.0%), can be significantly improved with unsupervised bagging. This lack of initial drift influences the bagging performance in two ways. First, the unsupervised seeds will be more precise. Second, the base lexicons generated will be more precise. As a result there will be ample precise terms extracted, which can be aggregated.

Not all drifting categories improved with bagging. The precisions of the FUNCTION, SYMPTOM and TUMOUR lexicons were significantly reduced after bagging. This was unexpected for the SYMPTOM category, as the other categories with little initial semantic drift improved significantly with bagging. The unsupervised seeds were expected to be predominantly correct and generate precise lexicons for aggregating. The SYMPTOM seeds were correct, however they

were also polysemous. These unfortunately encouraged semantic drift earlier in the base lexicons, and consequently increased the likelihood of selecting drifted terms into the aggregated lexicon. This was similarly observed with the FUNCTION category.

The TUMOUR category shows that unsupervised bagging is not appropriate for categories that suffer from semantic drift early (1-200: 52.5%). The unsupervised bagging significantly improved the precision over the first 200 terms (+12.5%, $p \leq 0.0001$). However, the subsequent measurements were significantly less precise than the initial lexicon. This is also observed following supervised bagging (TUMOUR 1-200: 86.5%, 401-600: 48.0%, 801-1000: 29.5%). Together, these results show that the seeds are only partially responsible for the precision of the aggregated lexicons. The TUMOUR category is also affected by its tendency to quickly drift into BODY PART, and then during aggregation the common drifted terms are selected.

7.6 Summary

This chapter began with a discussion on the unreliability of the traditional methodology for evaluating and comparing bootstrapping algorithms — using one set of seeds. I consequently identified the need to evaluate the lexicons extracted after seeding a bootstrapper multiple times with different random gold seeds. Using this new evaluation methodology, I have also presented the first in-depth performance analysis of BASILISK, MEB and WMEB. Each algorithm's sensitivity to the initial seeds, the variability of the lexicons they extract, and their average performance over multiple seed sets, has been investigated.

The presented analyses have demonstrated that each algorithm is sensitive to the seeds, and in turn generates diverse lexicons. BASILISK is most sensitive to different seeds; extracting very diverse lexicons, and is also the least stable bootstrapper, followed by MEB. On average BASILISK performed significantly worse with the hand-picked seeds, and has a large performance standard deviation. These observed performance variations support my claim that the standard evaluation paradigm, using only one set of seeds is inadequate for comparing bootstrapping algorithms. Robust evaluation requires results averaged across randomised seed sets.

And through this evaluation, it has been shown again that WMEB significantly outperforms MEB and BASILISK.

This chapter also presented new supervised and unsupervised bagging approaches, which successfully exploit the wide variation of the multi-category bootstrappers. While the combination of models have been studied previously for other NLP tasks, their use in the bootstrapping setting is novel. My supervised bagging approach was shown to significantly outperform the standard approach with hand-picked seeds, by aggregating multiple lexicons extracted by a bootstrapper seeded with gold terms. Supervised bagging was most effective for algorithms that are more susceptible to semantic drift, and more sensitivity to the input seeds, such as BASILISK and MEB, which allow for more possible corrections during aggregation.

Unfortunately, supervised bagging reduces a major advantage of bootstrapping — requiring only a few gold seeds to rapidly extract new semantic lexicons. To maintain this property and also benefit from bagging I proposed a novel unsupervised bagging method. My unsupervised approach requires no more gold seeds than the original bootstrapping algorithm, as the bagging seeds are sampled directly from the lexicons extracted with the initial hand-picked seeds. I have demonstrated that unsupervised bagging is best suited to algorithms which generate more precise lexicons, with frequent terms, so to allow diverse seed selection whilst reducing noise. Semantic drift within the lexicons extracted by WMEB and MEB, was significantly reduced following unsupervised bagging. In particular, unsupervised bagging of WMEB significantly outperformed the supervised approach, and the hand-picked seeds.

Although bagging can reduce semantic drift within the extracted lexicons after bootstrapping, it does not address the underlying cause of drift. In the following chapter, I present a novel approach for detecting and preventing semantic drift during the bootstrapping process.

Chapter 8

Detecting Semantic Drift

Unsupervised bagging effectively corrects much of the semantic drift introduced during bootstrapping by selecting terms extracted by multiple bootstrappers in the earlier iterations. However, as semantic drift still dominates in the later iterations, drifted terms are eventually selected into the bagged lexicons. This is because bagging does not directly correct semantic drift. This chapter presents a novel approach for detecting and preventing semantic drift during the bootstrapping process, and corresponds to part of the work presented in McIntosh and Curran (2009a). The approach is based on my hypothesis that semantic drift occurs when a candidate term is more similar to the group of recently added terms than to the seeds and high precision terms extracted in the earlier iterations.

This chapter begins by exploring the distributional similarity approach that is often used to extract semantic lexicons. Initial results demonstrate that distributional similarity is less effective than WMEB. Section 8.4 discusses the intuition behind my hypothesis, and introduces the novel drift metric that utilises distributional similarity measurements over the expanding lexicon, to identify drifting candidate terms. The drift metric measures the variation of a term's similarity between the initial terms and the recently extracted terms, and is incorporated directly into the WMEB algorithm to detect and prevent the extraction of drifting terms during bootstrapping. This reduces semantic drift as the prevented terms cannot contribute more drifting patterns. In Section 8.5, the drift metric is shown to significantly reduce drift in WMEB and outperform previously proposed distributional similarity filters and unsupervised bagging.

8.1 Distributional Similarity

Another common approach for automatically extracting semantically related terms is *distributional similarity* (Grefenstette, 1994). This approach is based on Harris's (1954) *distributional hypothesis* that *similar terms appear in similar contexts*. In a computational framework, each term is represented by a vector of relations describing the contexts surrounding the term in text. Each vector records the observed distributional patterns of a term with each context relation in a corpus. Comparing the similarity of two terms involves directly comparing their *context vectors*, and terms are considered similar if their context vectors are similar. Existing approaches vary primarily in the way they define *context* and the distance metric used for measuring the similarity between the contexts vectors.

8.1.1 Context

The context vectors formed by most similarity systems utilise co-occurrence and/or syntactic information from the terms surrounding the head term. They cover a wide range of linguistic information ranging from co-occurring neighbouring terms, through shallow to more sophisticated deep syntactic analysis.

The simplest form of context vectors define a term with respect to its co-occurring neighbouring words within a limited distance. This defines a context relation instance as a tuple (w, w'), where w is a term that has the word w' as one of its neighbours, within a fixed *window* of surrounding terms. The attributes are often extended to record the position of w', and can have unbalanced window boundaries for neighbours before and after w. Window-based extractors are very easy to implement and efficient due to their low complexity. After tokenisation, these extractors are almost language independent.

More sophisticated approaches incorporate grammatical relations. These approaches will produce more informative context vectors. Curran (2004) defines these relation instances as tuples (w, r, w'), where w is a term which occurs in a syntactic relation r with another word w' in the same sentence. The tuple (r, w') is referred to as an attribute of w. Grefenstette's (1994)

thesaurus extraction system, SEXTANT, uses shallow grammatical relations between a term and its surrounding words to provide context. The grammatical relations used associate nouns with verbs, modifiers and prepropositional phrases. In comparison, Lee and Pereira (1999) and Lee (1999) only used the direct-object relations between a noun and a verb, extracted by the CASS partial parser (Abney, 1996), while Hindle (1990) used subject-verb relations.

Grammatical relations extracted by full parsers, such as MINIPAR (Lin, 1998c) and RASP (Briscoe and Carroll, 2002), have also been exploited to construct more informative context vectors. Lin's (1998a) distributional similarity framework utilises the grammatical relations extracted by the broad-coverage parser, MINIPAR, from newspaper text. MINIPAR (Lin, 1998c) is based on a manually constructed grammar with 59 dependency relationships. Lin (1998a) showed that using multiple relations is beneficial, and his system has been applied in many other NLP tasks, including Paşca et al.'s (2006) large-scale fact extraction bootstrapper. Padó and Lapata (2003) extended the context relations based on grammatical relations formed from dependency paths. Their approach uses MINIPAR dependency relationships to form context relations from chains of MINIPAR dependencies. The resulting context vectors enable semantic relations between terms, such as hyponymy and synonymy, to be distinguished. Others have used relation data extracted by the RASP parser to form context relations for vector-space semantic similarity (McCarthy et al., 2003; Weeds and Weir, 2003). Curran (2004) provides a detailed comparison of context representations.

Lin and Pantel (2001) extended Harris's (1954) distributional hypothesis, to apply to dependency paths, where *if two paths tend to occur in similar contexts, the meanings of the paths tend to be similar*. Their approach identifies similar *inference rules*, which are essentially patterns for identifying semantic relations between terms, such as:

Y *is solved by* X

X *resolves* Y

As Lin and Pantel (2001) are identifying similar patterns, the context vectors represent the context relations the patterns appear in, that is the context vectors are composed of the terms which fill the slots of the patterns.

8.1.2 Similarity

The second component of a distributional similarity system performs nearest-neighbour or cluster analysis to determine term similarity. Once the set of context attributes is defined, the frequency counts of the co-occurrences of the head terms (or patterns in Lin and Pantel (2001)) with each of the attributes are collected. This produces a vector of attributes and their frequencies in the corpus. The degree of similarity between two target words is then determined by a vector comparison function. Both Weeds (2003) and Curran (2004) evaluated many common measure and weight metrics proposed in the literature.

The measure function calculates the similarity between two weighted context vectors. Several different types have been considered, including geometric distance metrics (Manhattan and Euclidean distance), IR inspired metrics (DICE, JACCARD and COSINE), as well as distributional methods (Pereira et al., 1993). Amongst the many proposed measures, Lin's (1998b) measure based on his information-theoretic similarity theorem (Lin, 1997, 1998b) and cosine are the most widely used within NLP.

Hindle (1990) proposed a mutual information based vector similarity measure. His measure factors in the strength of association of the attributes (co-occurring subject-verb relations) with each head term, using pointwise mutual information (PMI, see Section 6.2.3). Semantically similar head terms, will have similar PMI scores of the co-occurring attributes.

The notion of substitutability has also been considered for the basis of distributional similarity measures, whereby the more suitable it is to substitute a word with another, the more semantically similar they are, and that the substitutability of one word by another is not necessarily symmetrical (Lee, 1999; Weeds, 2003). The *Kullback-Leibler divergence* (KL-divergence) measure is asymmetrical, and Lee (1997) motivates its use in distributional similarity. Lee (1999) proposed the α-*skew divergence*, by modifying KL-divergence so that it does not require smoothed probabilities.

Weeds (2003) proposed distributional measures, called the *co-occurrence retrieval models* (CRM), that are based on substitutability. The degree of which word w_1 can be substituted

by word w_2, is based on how well w_2 retrieves the co-occurrence attributes identified by w_1. Weeds's (2003) distributional measure is defined in terms of w_2's recall and precision of w_1's attributes, and the importance of both precision and recall are specified with empirically set weights. Equal weights for recall and precision produces a symmetrical distributional similarity score for w_1 with w_2 and w_2 with w_1.

To improve performance of the similarity measures, a weighting may be applied to assign higher value to context attributes that are more informative indicators of semantic distance than others. The attributes with higher weight will contribute more to the distance calculation between two terms. The weighting is usually a measure of the association strength of the co-occurring attribute with the target words. For example, Dagan et al. (1993, 1995) used PMI to weight the context vectors compared using the JACCARD measure, while Curran (2004) showed that the TTEST is the most effective weighting with JACCARD. In some cases, the weighting applied results in an existing distance measure, such as Hindle's (1990) and Lin's (1998b), which incorporate PMI directly.

8.2 Extracting Semantic Lexicons using Distributional Similarity

This section describes and evaluates the distributional similarity approach for extracting biomedical semantic lexicons, based on the seed terms for each semantic category of interest. I introduce the language-independent window-based contextual relations and the weighted similarity measure used in these experiments.

Like the bootstrapping algorithms, this approach takes as input the set of candidate terms that may be extracted into a semantic lexicon depending on their degree of similarity to the seeds terms. Each of the candidate terms and seeds are represented by context vectors. For a given seed term, the system computes the similarity of the seed with all other candidate terms and returns a list of terms similar to the seed ranked in descending order of semantic similarity.

To compose a category's semantic lexicon, the individual lists obtained by each seed term are merged. The merged terms are then ranked and the top-n terms are selected to form the

final semantic lexicon. This process is similar to the aggregation phase in bagging and follows the same hypothesis (see Section 7.4). I experiment with five aggregation methods:

1. SET: terms are ranked by the number of different lists they appear in;
2. SCORE: terms are ranked by the sum of their distributional similarity scores across the lists they appear in;
3. RANK: terms are ranked by the sum of their inverse ranks across the lists they appear in;
4. SET+SCORE: terms are ranked by SET, and to break ties, the term with the highest SCORE is selected;
5. SET+RANK: terms are ranked by SET, and to break ties, the term with the highest RANK is selected.

8.2.1 Contextual Information for Candidate Terms

The contextual relations for each candidate and seed term are extracted from the set of MEDLINE filtered 5-grams (t_1, t_2, t_3, t_4, t_5). The candidate terms correspond to the set of middle tokens (t_3). Using a window-based approach, four different relation attributes that consider the position of the two co-occurring left and right surrounding terms are extracted. For example, the relation instance (t, L1, t'), is assigned to the term t which occurs with another term t' in the first left position of a 5-gram. Each candidate term is represented by a vector of frequency counts of its co-occurrences with each relation instance. Relation instances occurring less than 5 times across the vectors are filtered.

The context attributes considered for distributional similarity are a generalisation of the patterns used for bootstrapping. In bootstrapping, generalising patterns into unigrams would be detrimental and thus strict context specific patterns are used to reduce the search space and in turn semantic drift. The coverage of bootstrapping is then extended gradually when more patterns are iteratively introduced. Distributional similarity on the other hand, requires a broader

view of each term's co-occurrences in one representation to capture any pair-wise similarities between terms. By splitting the bootstrapping patterns into four individual attributes, the frequency of the initial pattern co-occurrences are smoothed over all of the attributes.

8.2.2 Calculating Similarity

Once the context vectors are created for each candidate word and seed term, vector similarity is calculated using the weighted JACCARD measure (Equation 8.1) with the standard TTEST (Equation 8.3) as its context relation weight function. The TTEST weight assigns a higher value to contexts that are more indicative of the meaning of that word. Curran (2004) showed that the TTEST weighting and both the DICE and JACCARD measures, producing identical results, significantly outperformed all other weights and measures including Lin's (1998b) on semantic lexicon extraction.

The notation used here is from Curran (2004), and extends that used by Lin's (1998b). The tuple (w, r, w') represents the relationship r between a word w and another word w'. An asterisk $(*)$ indicates the set of all existing values of that component in the relationship tuple. Subscripted asterisks are used to indicate the binding of variables.

Grefenstette (1994) defines the weighted JACCARD measure for context vector similarity of two words as:

$$\text{JACCARD}(w_1, w_2) = \frac{\sum min(wgt(w_1, *_r, *_{w'}), wgt(w_2, *_r, *_{w'}))}{\sum max(wgt(w_1, *_r, *_{w'}), wgt(w_2, *_r, *_{w'}))} \tag{8.1}$$

where $wgt(w, r, w')$ is a placeholder for a weight function, such as the TTEST.

The t-test is a standard hypothesis testing technique used to reject a *null hypothesis* with some level of confidence. The t-test compares a value x against a normal distribution defined by its mean μ, sample variance s^2 and sample size N:

$$\tau = \frac{x - \mu}{s}\sqrt{N} \tag{8.2}$$

Category	SET	SCORE	RANK	SET+SCORE	SET+RANK	WMEB
ANTIBODY	18	26	45	30	21	96
CELL	71	93	93	92	69	100
CELL LINE	57	72	73	72	57	100
DISEASE	80	93	87	90	82	84
DRUG	75	81	86	81	75	99
FUNCTION	89	88	75	87	88	82
MUTATION	96	55	33	82	95	84
PROTEIN	83	85	77	84	84	100
SYMPTOM	63	80	80	80	59	99
TUMOUR	35	52	61	51	35	37
Average	66.7	72.5	71.0	74.9	66.5	88.1
Inv Rank	3.5	4.5	4.4	4.4	3.5	4.8

Table 8.1: Distributional similarity: MEDLINE individual category results (1-100 terms)

To determine the association strength within relations, that is between headwords and attributes, this is translated into:

$$\text{TTEST}(w_1, r, w_2) = \frac{p(w_1, r, w_2) - p(*, r, w_2)p(w_1, *, *)}{\sqrt{p(*, r, w_2)p(w_1, *, *)}} \tag{8.3}$$

8.2.3 Results

This section investigates the performance of Curran's (2004) distributional similarity system for extracting biomedical semantic lexicons. In these experiments, the top 1000 distributionally similar terms of each seed term used in the bootstrapping experiments are extracted. A semantic lexicon is then constructed for each biomedical category of interest by aggregating the five individual lists identified by each of its seed terms. Each category's final aggregated lexicon can contain up to 5000 terms (assuming no overlap). Each lexicon's top-500 ranked terms are evaluated following the same manual procedure described in Section 5.4.

Table 8.1 and 8.2 show the precision of the first 100 and last 100 terms that form each individual category's lexicon after aggregating the corresponding seeds' lists. The results for the best performing bootstrapping algorithm, WMEB, are also included.

SET+SCORE P: 51.0	glioblastoma 0.137, glioma 0.136, hepatoma 0.119, myeloma 0.118, choriocarcinoma 0.105, fibrosarcoma 0.093, astrocytoma 0.088, *lymphoblastoid* 0.087, leukemic 0.080, mesothelioma 0.079, insulinoma 0.079, teratocarcinoma 0.079, rhabdomyosarcoma 0.077, medulloblastoma 0.076, hybridoma 0.076, *osteoblastic* 0.073, tumour 0.071, hepatoblastoma 0.070, adenocarcinoma 0.070, thymoma 0.067, erythroleukemia 0.067, sarcomas 0.067, *monocytic* 0.067, chondrosarcoma 0.066 ...
WMEB P: 37.0	glioblastoma, hepatoma, leukemia, hepatoblastoma, myeloma, adenocarcinoma, *lymphoblastoid*, leukemic, carcinoma, *monocytic*, RCC, *myeloid*, *keratinocyte*, rhabdomyosarcoma, astrocytoma, *lymphoblast*, *lymphoblastic*, MM, HCC, choriocarcinoma, erythroleukemia, *trophoblast*, *stem*, seminoma, *urothelial* ...

Figure 8.1: TUMOUR semantic lexicons extracted by distributional similarity (SET+SCORE) and WMEB (1–20 terms)

Over the first 100 terms, the SET+SCORE aggregation approach outperformed the other distributional based lexicons, however it is not competitive on average with WMEB. When the individual categories were considered, two unexpected characteristics were noticed. Firstly, many of the categories that are extracted precisely by WMEB — ANTIBODY, CELL LINE, DRUG, PROTEIN, and SYMPTOM— are extracted poorly using the distributional similarity approach. In particular, ANTIBODY precision dropped by 51% with RANK, and by 66% with the best system, SET+SCORE. Secondly, the categories that are extracted less precisely by WMEB are extracted more precisely using the distributional similarity approach. For example, the TUMOUR lexicon extracted by RANK is more precise than that extracted by WMEB (RANK +24% at 1-100; $p \leq 0.0001$).

Figure 8.1 shows the top-20 terms in the TUMOUR semantic lexicons extracted by SET+SCORE and WMEB. The terms in the SET+SCORE lexicon are associated with their total distributional similarity score. Terms that are incorrectly extracted into the TUMOUR lexicon are shown in red and italics, and the first incorrect term for both methods is the same (*lymphoblastoid*). Each of the incorrect terms are either types of CELL (e.g. *lymphoblast*, *keratocyte* and *stem*) or adjectives pertaining to CELL and BODY PART (e.g. *lymphoblastic*, *lymphoblastoid* and *urothelial*). WMEB suffers from semantic drift very early when extracting the TUMOUR lexicon even though it is aware of, and competing with, other semantic categories, such as CELL and BODY PART.

Category	SET	SCORE	RANK	SET+SCORE	SET+RANK	WMEB
ANTIBODY	15	2	11	5	13	94
CELL	24	26	26	11	23	29
CELL LINE	43	43	48	41	53	100
DISEASE	74	70	58	60	73	97
DRUG	55	54	60	49	56	97
FUNCTION	43	51	41	49	45	78
MUTATION	19	12	25	23	19	71
PROTEIN	50	56	65	59	51	100
SYMPTOM	48	40	42	37	47	83
TUMOUR	8	16	15	9	7	85
Average	37.9	37.0	39.1	34.3	38.7	83.4
Inv Rank	2.0	2.2	2.0	1.8	2.2	4.4
Average 1-500	52.3	51.5	52.3	55.0	53.5	88.7

Table 8.2: Distributional similarity: MEDLINE individual category results (401-500 terms)

SET+SCORE P: 9.0	*T47D* 0.022, *3T6* 0.022, *AR42J* 0.022, *synovial* 0.022, *nonneuronal* 0.022, *foreskin* 0.022, hemangioendothelioma 0.022, *CV-1* 0.021, mesenchymoma 0.021, *phenylketonuria* 0.021, *OVCAR-3* 0.021, *Rat-1* 0.021, *INS-1* 0.021, histiocytosis 0.021, *uninfected* 0.021, *cumulus* 0.021, *Hürthle* 0.021, *HCT116* 0.021, *T84* 0.021, *C6* 0.021
WMEB P: 85.0	IMSCT, IPT, JGCTs, SSTs, IITs, PRET, RPTs, SLCTs, SMTs, SPPT, carcinosarcomas, TCTs, TMPNST, UTROSCTs, UUTT, cystadenocarcinomas, thymomas, fibroadenomas, pheochromocytomas, HBT

Figure 8.2: TUMOUR semantic lexicons extracted by distributional similarity (SET+SCORE) and WMEB (480–500 terms)

On the other hand, SET+SCORE initially extracts significantly more precise TUMOUR terms based on only the five seed terms.

This evaluation has indicated that the distributional similarity approaches can outperform WMEB for some semantic categories. However, the analysis so far has only considered the first 100 terms extracted, which does not allow us to compare the usability of these methods for extracting large precise biomedical semantic lexicons. Table 8.2 shows the precision of the 401-500 terms in each semantic category, and the average precision over the first 500 terms, identified by the distributional similarity approaches and WMEB.

The SET+SCORE method is the most effective of the distributional similarity approaches. However, it is much less precise on average than WMEB ($p \leq 0.0001$). SET+SCORE suffers

significantly from semantic drift, and each of the lexicons extracted by SET+SCORE, including TUMOUR, are significantly less precise than those of WMEB. Therefore, distributional similarity alone is not suitable for extracting large semantic lexicons.

Figure 8.2 shows the terms ranked 480-500 in the TUMOUR semantic lexicons extracted by SET+SCORE and WMEB. The incorrect terms extracted by SET+SCORE were predominantly CELL LINE (e.g.*3T6* and *Hürthle*), and those referring to other DISEASES (*phenylketonuria*). Drifting between related categories is common with distributional similarity approaches, as unlike WMEB, they are not aware of competing categories. Over the same range of ranked terms, WMEB did not extract any incorrect TUMOUR terms and extracted many rare abbreviations of TUMOUR (e.g.*UTROSCTs*: *Uterine tumors resembling ovarian sex cord tumors*).

The degree of overlap between the lexicons of WMEB and SET+SCORE (1-100: 30.3%), further suggests that these methods identify different types of terms, and are thus complementary. In particular, the categories that are more precise with SET+SCORE have little overlap with WMEB (1-100: DISEASE 13%, FUNCTION 33%, TUMOUR 39%), and are thus extracting correct terms not identified by WMEB. Over the 401-500 range the overlap is only 3.1%. In comparison, SET+SCORE and RANK identify significantly more common terms.

8.3 Distributional Similarity with Pattern-based Approaches

Recently, systems have been proposed that exploit the complementary nature of both distributional similarity and pattern-based approaches by combining them in a pipeline. Lin et al. (2003a) utilised lexical patterns to identify pairs of synonyms and antonyms among a head term's distributionally similar nouns. To distinguish between the types of pairs, they exploited two *patterns of incompatibility*, which suggest that pairs of terms matching the patterns are semantically incompatible:

from X *to* Y

either X *or* Y

Lin et al. (2003a) considered a pair of distributionally similar terms X and Y to be synonymous if the ratio between the number of occurrences of the terms appearing *near* each other and the number of occurrences of the terms matching the two patterns, is greater than a specified threshold. In their experiments, the threshold was set to 2000. Their approach was evaluated on known pairs of synonyms and antonyms among the top-50 distributionally similar terms, and reported 95% recall and 86.4% precision in identifying synonymous terms.

Pantel and Ravichandran (2004) proposed to label clusters of distributionally similar terms using lexico-syntactic patterns by identifying hyponym relations between each member of a cluster and the cluster label. Terms within the output of Lin's (1998a) distributional similarity system are first clustered using the algorithm *Clustering by Committee* (CBC, Pantel and Lin, 2002) in order to separate polysemous terms. In CBC each cluster is assigned a set of terms, known as a committee, which unambiguously represents the cluster. Each committee member is represented by a feature vector of its co-occurring syntactic relations. These feature vectors are averaged to form a syntactic signature of the cluster. To label a cluster, a set of four patterns, which identify hyponym relations are searched for within the cluster's signature. The term that most frequently occurs with the matching patterns is considered a hyponym of each of the cluster's members, and is assigned the cluster's label.

Mirkin et al. (2006) highlighted the mutual complementary nature of pattern-based and distributional evidence by improving the performance of lexical entailment relation extraction using a system that integrates the approaches. Candidate terms, which lexically entail a given head term, are identified by either using 11 manually constructed lexical patterns in which terms co-occur with the target in web data, or by using Geffet and Dagan's (2004) distributional similarity system.[1] Candidate entailment pairs are formed by pairing each head term with each candidate term. In the next stage, both approaches are exploited again to represent each candidate pair by a vector of both pattern-based and distributional features. These feature sets are also used to build a SVM classifier based on a set of training entailment pairs, for distinguishing between correct and incorrect pairs. Incorrect candidate pairs are then filtered

[1] Geffet and Dagan's (2004) similarity measure extends Lin's (1998b) by assigning attribute weights through a bootstrapped weight function.

using the SVM classifier. Their combined approach outperformed each individual component, on a set of 20 head terms, by greater than 10% F-score.

Paşca et al. (2006) also noted the utility of integrating these approaches. Their large-scale bootstrapping fact extraction algorithm, described in Section 4.4.3, incorporates distributional similarity directly within the bootstrapping process to validate and rank candidate facts based on their similarity to the initial set of seed facts. In the two bootstrapping iterations, each candidate fact that is not distributionally similar to any of the seed facts is filtered from the candidate set, and thus cannot be extracted. A candidate fact is assigned a similarity score of zero if its neighbouring terms do not co-occur with those of a seed fact.

Paşca et al. (2006) state that this candidate fact validation is essential when collecting large candidate sets of facts from noisy web text in two iterations. However, within a less aggressive bootstrapping framework where a small number of terms are added in each iteration, such as WMEB, it seems less appropriate. Firstly, it applies an unreasonably strict restriction that the candidate terms must be similar to a seed term. This conflicts with the behavior of iterative bootstrapping — new terms help identify additional candidate terms, and in turn candidates can rank highly based on patterns that do not match any seeds. Therefore, these candidates may not be similar to a seed. Secondly, any candidate term that shares at least one co-occurring term with a seed term, will receive a non-zero similarity score and thus the non-zero score cannot confirm a term's correctness. Further, a zero threshold does not consider the various degrees of similarity between a term and the seeds — a term that is very similar to each of the seeds is considered just as correct as a term that has little similarity to just one seed.

8.4 Semantic Drift Detection in WMEB

In this section, I describe my novel approach to detect and prevent semantic drift during the bootstrapping process. This approach is motivated by the intuition that a lexicon's meaning drifts rapidly when ambiguous patterns extract either incorrect or ambiguous terms that are similar to each other. These new terms can introduce incorrect patterns, which in turn identify

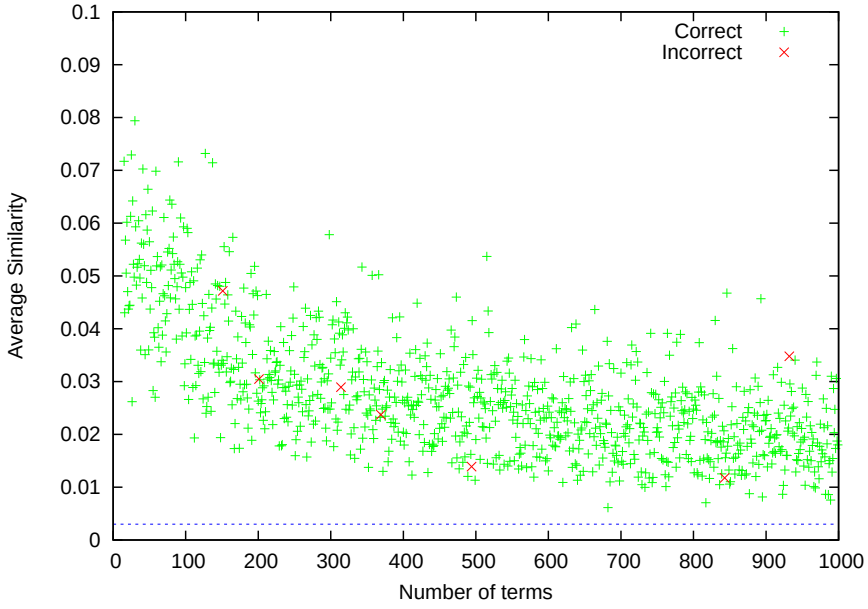
additional incorrect terms. I introduce a novel metric that is based on distributional similarity measurements over the extracted lexicon, to identify this behavior and drifting candidate terms.

8.4.1 Motivation

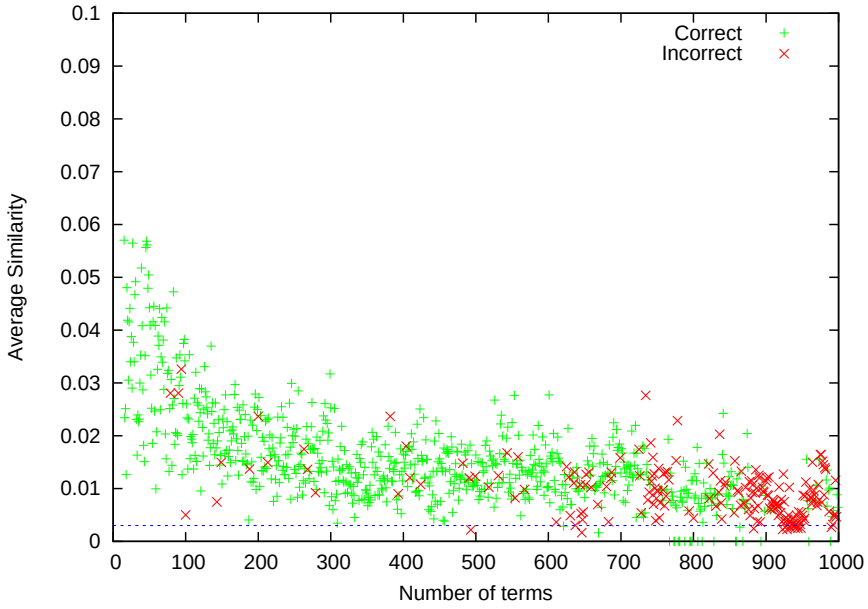
It is expected that as bootstrapping progresses a lexicon's newly added terms will be less similar to the seeds and the terms added in the earlier iterations. These less similar terms are not necessarily incorrect. To verify this, I plotted the distribution of correct and incorrect terms extracted by WMEB. Each term's rank within the lexicon is plotted against its average distributional similarity score to the first 20 terms (5 seeds + first 15 extracted terms).

Figure 8.3 shows the behavior of two biomedical categories, PROTEIN (8.3a) and ANTIBODY (8.3b), which are extracted very precisely (ANTIBODY less so in the later iterations), and Figure 8.4 shows the behavior the CELL (8.4a) and MUTATION (8.4b) categories that suffer greatly from semantic drift. Correct and incorrect terms are indicated by a green and red cross, respectively. A dashed line is shown at an average score value of 0.003, which provides a balance between precision and recall of terms with low average similarity scores. Visual inspection of Figure 8.3 clearly shows a trend indicating that as bootstrapping progresses the correct and incorrect terms are less similar to the first 20 terms. This trend is also observed in the categories with more semantic drift (Figure 8.4).

Many incorrect terms have similar average similarity scores to correct terms. For example, there are numerous incorrect terms scattered among the correct terms with similar scores in the MUTATION lexicon (Figure 8.4a). These graphs also support my intuition that Paşca et al.'s (2006) zero similarity filter will have little influence on preventing the initial stages of semantic drift. This filter will only remove some of the incorrect terms extracted as a result of the drift. For example, in the CELL lexicon, drift begins at approximately 400 terms and the topic shifts completely after 700 terms, resulting in the extraction of terms with little or no similarity to the first 20 terms.

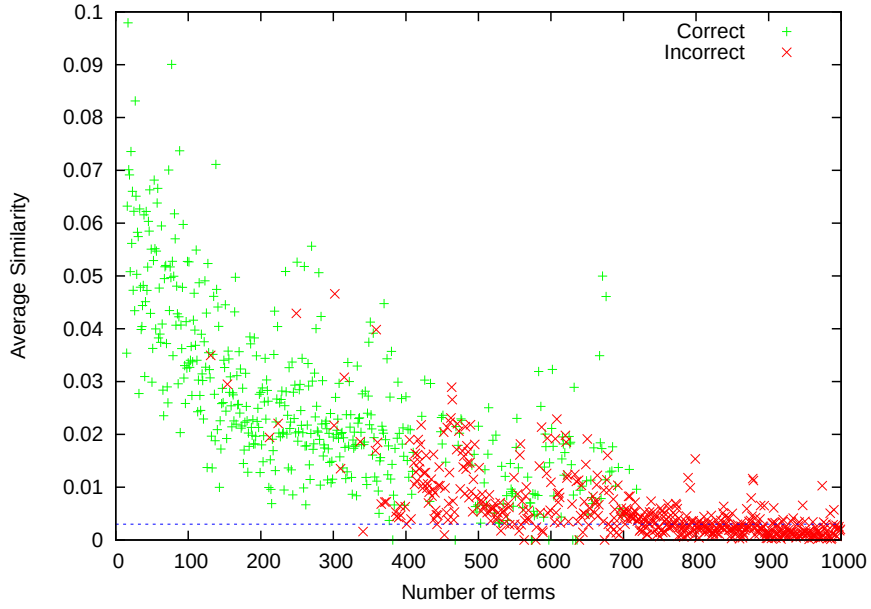


(a) PROTEIN

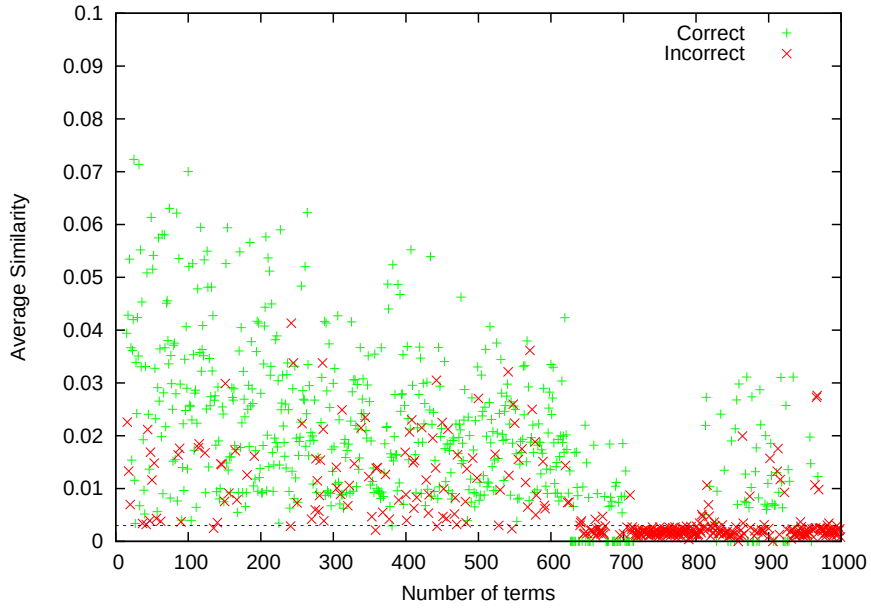


(b) ANTIBODY

Figure 8.3: Average distributional similarity of lexicon terms to the first 20 terms



(a) CELL



(b) MUTATION

Figure 8.4: Average distributional similarity of lexicon terms to the first 20 terms

A simple solution for detecting the initial drift, is to filter terms with an average similarity value below 0.02 or 0.01. However, this is completely inappropriate for the ANTIBODY and MUTATION lexicons. Further, stricter filters, like 0.003, will simply remove the incorrect terms in the later stages, and again not address the underlying cause of semantic drift. These observations indicate that it is impossible to identify a filter threshold suitable for all semantic categories of interest, and most importantly that semantic drift cannot be identified or prevented using only similarity comparisons to the initial terms, without eliminating many correct terms in the process.

8.4.2 Drift Metric

For a lexicon's associated meaning to shift, it is necessary for more than a few scattered polysemous or incorrect terms to be extracted across numerous iterations. These scattered terms are typically spurious and appear in similar patterns and contexts to many of the correct terms extracted around them, and thus have very little influence on identifying further incorrect terms in the subsequent iterations. However, as WMEB progresses, the earlier precise patterns match far fewer of the candidate terms, and the more recently added patterns that match these scattered terms begin to dominate. This results in the extraction of clusters of incorrect terms with similar meanings, which in turn identify new drifting patterns and then the lexicon's associated meaning shifts.

For example, during WMEB's extraction of ANTIBODY, the terms *DFA* (rank 95; incorrect), *DIFA* (rank 492; incorrect), *ICMA* (rank 530; polysemous), *FFA* (rank 544; incorrect), and *MACRIA* (rank 555; incorrect), are identified. Each of these terms match at least one of the following patterns that correctly identify many ANTIBODY terms.

antibody (<term> *) was/is*

antibodies (<term> *) are*

antibody (<term> *) assay*

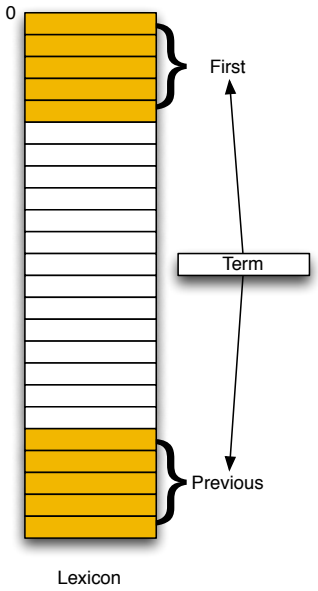


Figure 8.5: Diagram of drift detection during bootstrapping

However, only *ICMA* refers to a specific type of ANTIBODY.[2] Each of the terms refer to an experimental procedure (*assay*) that utilises antibodies, and match other patterns more frequently, such as:

assay (<term> *) based*

assay (<term> *) tests*

These patterns begin to dominate in the later iterations and the lexicon's associated meaning drifts from ANTIBODY to procedures. As a result, terms such as *ELISAs* (rank 742; incorrect), *EMSAs* (rank 743; incorrect), *cELISA* (rank 762; incorrect), *TaqMan* (rank 763; incorrect), *SPERA* (rank 877; incorrect) and *DELFIA*[3] (rank 906; incorrect) are extracted. This behavior is similarly observed in the CELL lexicon, which extracts abbreviations referring to both CELL and chromatography techniques.

Based on my observations in the manual evaluation process, I hypothesise that semantic drift within a lexicon occurs when a candidate term to be extracted is more similar to a cluster of recently added terms than to the seed and/or high precision terms extracted in the

[2] *ICMA* is an abbreviation for *islet cell membrane antibodies* and *Immunochemiluminometric assays*.
[3] DELFIA (dissociation enhancement lanthanide fluoroimmunoassay)

earlier iterations. I propose a new selection measurement, called *drift*, which reflects a term's degree of similarity to the initial terms versus the previously extracted terms (see Figure 8.5 and Equation 8.4).

Existing approaches have only considered the similarity of a candidate term to the set of seeds. In my approach, I effectively exploit distributional similarity evidence to identify whether a term is more similar to the recently extracted terms than to those extracted in the earlier iterations. A ratio is used to combine the two pieces of similarity evidence. Given a growing lexicon of size N, L_N, let $L_{1...n}$ correspond to the first n terms extracted into L, and $L_{(N-m+1)...N}$ correspond to the last m terms added to L_N. In a bootstrapping iteration, let t be the next candidate term to be added to the lexicon. We calculate the average distributional similarity (sim) of t with all terms in $L_{1...n}$ and those in $L_{(N-m+1)...N}$ and call the ratio the drift for term t:

$$\mathrm{drift}(t, n, m) \quad = \quad \frac{\mathrm{sim}(L_{1...n}, t)}{\mathrm{sim}(L_{(N-m+1)...N}, t)} \tag{8.4}$$

Smaller values of $\mathrm{drift}(t, n, m)$ correspond to the current term moving further away from the first set of terms, and closer to the previous set of terms added before it, and are therefore indicative of semantic drift occurring. A $\mathrm{drift}(t, n, m)$ of 0.2 corresponds to a 20% difference in average similarity between $L_{1...n}$ and $L_{(N-m+1)...N}$ for term t, and t is more similar to the previous m terms than the first n terms. Drift values greater than 1.0 are assigned to terms that are more similar to the initial terms than the previous terms.

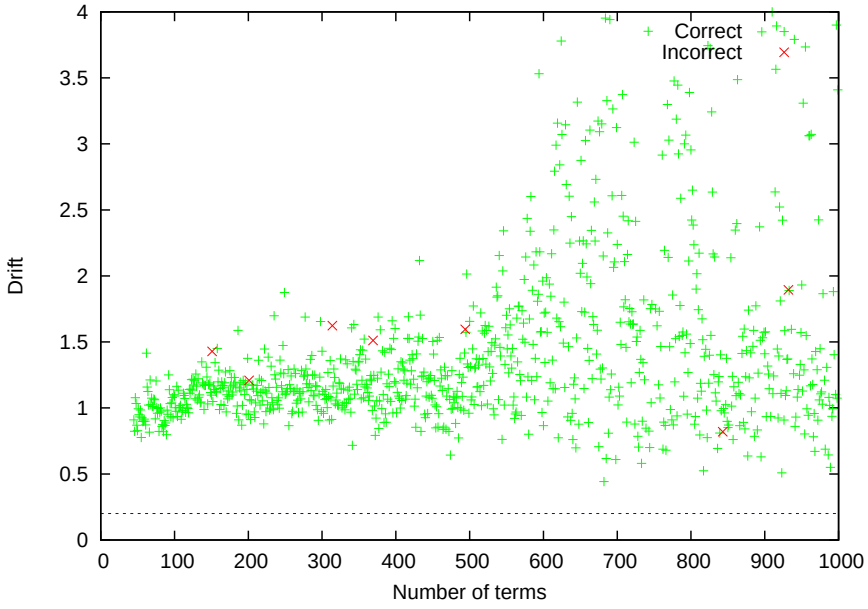
By using a ratio to identify differences between the average similarities, the drift metric is not affected by the relative magnitude of the average scores for the different categories, allowing consistent variations between the two scores to be detected. Therefore, one filter threshold can be used across all categories, which is not the case with a metric based on the difference between the average scores.

Figure 8.6 and 8.7 show the distribution of correct and incorrect terms extracted by WMEB with the term ranks plotted against their drift score, calculated over the first 100 and the previous 20 terms. Figure 8.6 shows the behavior of the PROTEIN (8.6a) and ANTIBODY (8.6b) lexicons, and Figure 8.7 shows the behavior the CELL (8.7a) and MUTATION (8.7b) lexicons. A drift score of 0.2 is indicated by the blue dashed line. These graphs show that there is a trend where low values of drift correspond to incorrect terms being added, and that the drift metric effectively separates the non-drifting terms from the majority of terms causing the initial semantic drift and those that are a consequence of drift. For example, the drifting CELL terms in the 400-600 range are separated more from the correct terms and have similar drift values to the incorrect terms in the later sections, than those based only on similarity comparisons to the initial 20 terms in Figure 8.4a.

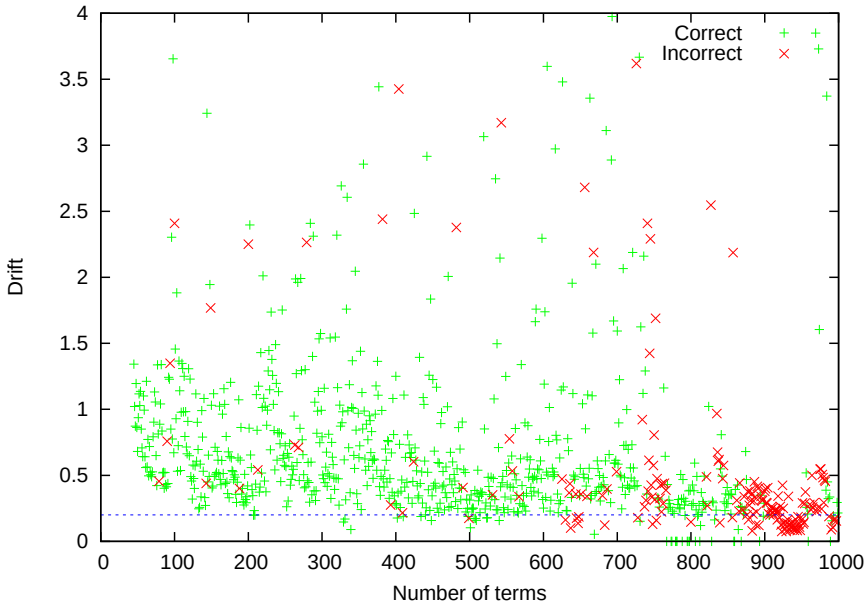
Drift can be used as a post-processing step to filter terms that are a possible consequence of semantic drift. However, my main proposal is to incorporate the drift measure directly within the WMEB algorithm, to detect and then prevent drift occurring.

In each iteration of WMEB, the set of candidate terms to be added to the lexicon are scored and ranked for their suitability as normal. Additionally, the drift of each candidate term is calculated before it is added to the lexicon. If a term's drift is below a specified threshold, θ, it is not selected and prevented from being selected in the subsequent iterations. If the term has zero similarity with the last m terms, but is similar to at least one of the first n terms, the term is selected. Preventing the term from entering the lexicon during the bootstrapping process has a flow on effect, as it will not be able to extract additional divergent patterns, which would lead to accelerated drift.

For calculating drift the distributional similarity system described in Section 8.2 is utilised. Instead of extracting lists of semantically similar terms, the system produces a similarity score for the candidate term with each of the first n terms and m previously added terms.

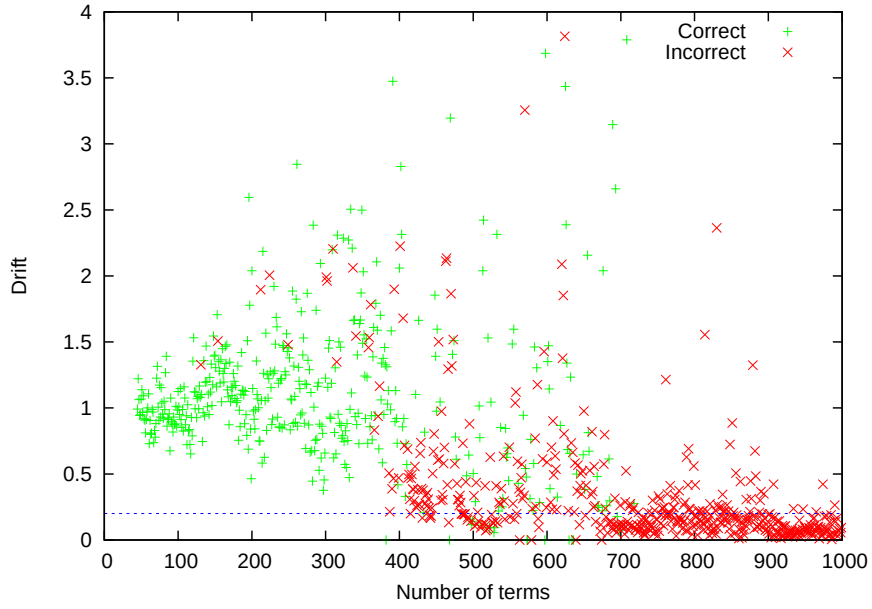


(a) PROTEIN

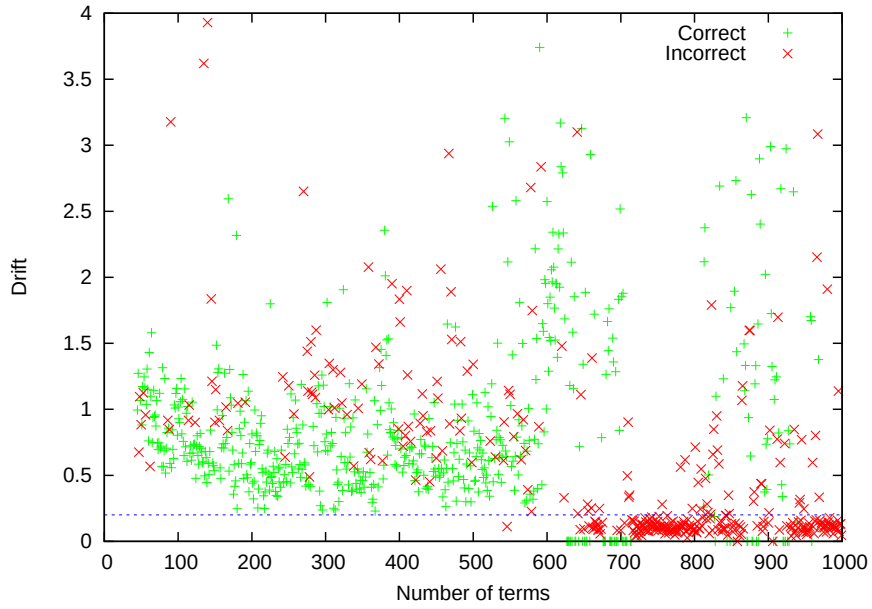


(b) ANTIBODY

Figure 8.6: Drift of terms to the first 100 and previous 20 terms



(a) CELL



(b) MUTATION

Figure 8.7: Drift of terms to the first 100 and previous 20 terms

8.5 Results

In this section, the effectiveness of the drift metric in combination with WMEB for detecting and preventing semantic drift is evaluated. The first set of results evaluates the simple similarity filters, which only consider the seeds and/or the initially extracted terms, such as Paşca et al.'s (2006). Section 8.5.2 presents the results of the drift metric as a post-processing filter over the lexicons extracted by WMEB and with the filter incorporated directly into WMEB (WMEB-DRIFT). This is followed by a discussion on the individual lexicons extracted by WMEB-DRIFT. This analysis concludes with an evaluation of WMEB-DRIFT using the randomised methodology introduced in Section 7.2.

In the experiments, various values of the drift metric's parameters — $L_{1...n}$ (n = 5 – 100) and $L_{(N-m+1)...N}$ (m = 5 – 100), are compared. A drift threshold of θ=0.2 is used, which was selected empirically, in order to achieve a balance between precision and recall. A higher value substantially reduced the lexicons' size, while a lower value resulted in little improvements. WMEB is initialised with the original five hand-picked seeds for the 10 biomedical semantic categories and the STOP categories, and is run for 300 iterations.

8.5.1 Simple Filters

Table 8.3 shows the effectiveness of applying simple similarity post-processing filters over the lexicons extracted by WMEB after 400 iterations (each lexicon initially contains 2000 terms). The first row corresponds to WMEB prior to filtering. POST-PASCA corresponds to Paşca et al.'s (2006) filter, which removes terms that are not similar to any of the seed terms. POST-SEED corresponds to removing terms that have an average distributional similarity score of less than 0.002 – 0.005 to the seed terms. Likewise, the remaining rows (POST-FIRST20) corresponds to the filter that considers a term's average similarity score to the first 20 terms (5 seeds and first 15 terms). The filter thresholds, 0.002 – 0.005, were selected in order to balance between precision and recall, after inspecting graphs such as those in Figure 8.3 and 8.4.

Algorithm	1-500	501-1000	1-1000
WMEB	88.7	68.6	78.6
POST-PASCA	88.6	67.4	78.0
POST-SEED			
0.002	88.4	67.4	77.9
0.003	87.5	(68.8) 64.5	(79.0) 76.1
0.004	87.0	(70.1) 61.8	(79.8) 74.4
0.005	(86.9) 86.8	(62.0) 55.2	(80.4) 71.1
POST-FIRST20			
0.002	88.3	67.9	78.1
0.003	88.0	(68.4) 68.2	(78.4) 78.1
0.004	87.8	(68.3) 67.1	(78.8) 77.4
0.005	87.4	(68.8) 64.0	(79.5) 75.7

Table 8.3: Post-processing WMEB lexicons with distributional similarity filters

As each extracted lexicon contains 2000 terms, a filtered lexicon may contain less than 1000 terms if more than 1000 terms are removed. These smaller lexicons will be penalised when they are evaluated past their stopping points. To take this into account, adjusted average precision scores are provided in parentheses.

The post-processing filters are ineffective in improving the average precision of the initial lexicons. After the initial terms are filtered, many of which were correctly extracted by WMEB, more incorrect terms that pass the filter are simply introduced into the top-1000 range. For example, POST-PASCA removed 169 terms from the lexicons of which only 16 terms were incorrectly extracted by WMEB. This result verifies that correct terms are not necessary similar to the seed terms. Further, the difference in performance between POST-SEED and POST-FIRST20, also indicates that the seed terms provide insufficient evidence for post-filtering.

The trade-off between precision and recall using different similarity thresholds is evident. As the thresholds are increased, far fewer terms are kept, and the precisions increase (adjusted precisions). Unfortunately, increasing the thresholds above 0.003 significantly reduces the sizes of many lexicons. For example, with POST-SEED 0.005 only four category lexicon's contained 1000 terms (e.g. DISEASE 492 terms, CELL 642 terms). With 0.003, only DISEASE and CELL have less than 1000 terms. These results show that these simple filters are insufficient for improving the initial lexicons without significantly reducing recall.

Algorithm	1-500	501-1000	1-1000
WMEB	88.7	68.6	78.6
WMEB-PASCA	87.8	72.5	80.2
WMEB-SEED			
0.002	86.8	(71.3) 71.2	(79.1) 79.0
0.003	86.0	(75.3) 72.2	(80.5) 80.2
0.004	85.4	(71.1) 66.8	(78.6) 76.1
0.005	85.4	(72.6) 58.0	(80.2) 71.7
WMEB-FIRST20			
0.002	87.8	(72.7) 72.5	(80.3) 80.2
0.003	87.7	(74.0) 73.3	(80.8) 80.5
0.004	86.8	(74.3) 72.1	(80.5) 79.7
0.005	86.9	(73.6) 70.3	(80.3) 78.5

Table 8.4: Inline distributional similarity filtering with WMEB

Table 8.4 presents the results of incorporating the simple filters directly into WMEB, whereby a candidate term will only be selected if it satisfies the filter. When the filters are incorporated, the extracted lexicons are significantly more precise than those after post-processing, while the average precision over the first 500 terms is worse. In the later stages where semantic drift is prominent, the precision is significantly higher (501-1000: $p \leq 0.0001$). These precision increases are a direct result of extracting more similar terms earlier, which in turn can influence the subsequent iterations.

The inline filters are most effective when more evidence is incorporated (SEED vs FIRST20), and when the filter thresholds are set to 0.003 or less. As in the post-processing approach, the higher thresholds significantly reduce the number of terms that can possibly be extracted. The best performing systems for inline and post filtering occur using FIRST20 at 0.003, and the inline approach significantly outperforms standard WMEB (+1.9% at 1-1000, $p \leq 0.0001$). However, as with post-processing identifying a threshold that is suitable in terms of precision and recall for each category is difficult. For instance, each of the thresholds completely exhaust the MUTATION candidate terms that WMEB considers, despite running WMEB for 300 iterations (e.g. only 879 and 678 MUTATION terms are extracted with WMEB-FIRST20 0.003 and 0.005, respectively and so approximately 50% of terms are removed).

Algorithm		1-200	201-400	401-600	601-800	801-1000	1-1000
WMEB		90.3	89.8	82.3	68.7	62.0	78.6
POST-DRIFT							
n	m						
5	5	89.0	86.5	72.8	65.0	63.0	75.3
5	20	88.2	72.2	65.0	65.0	64.7	74.9
5	50	88.0	84.3	71.9	65.5	65.0	75.0
5	100	89.3	84.7	71.9	65.6	65.2	75.3
20	5	90.3	88.0	80.7	66.8	60.9	77.3
20	20	90.3	87.5	78.4	67.8	60.9	77.0
20	50	90.3	87.5	77.8	68.3	60.8	77.0
20	100	90.3	87.8	77.7	68.8	60.8	77.1
100	5	90.3	89.0	81.4	68.3	61.5	78.1
100	20	90.3	89.0	81.5	68.8	61.5	78.2
100	50	90.3	89.2	81.2	69.2	61.5	78.3
100	100	90.3	89.2	81.0	69.3	61.6	78.3

Table 8.5: Post-processing WMEB lexicons with semantic drift detection (θ=0.2)

8.5.2 Semantic Drift Detection

Table 8.5 presents the results of applying the drift metric as a post-processing filter over the lexicons extracted by WMEB (POST-DRIFT). To pinpoint the sections of the lexicons which have changed, average precision measurements for every 200 terms across the first 1000 terms are shown. The first row corresponds to WMEB prior to filtering. The remaining figures correspond to POST-DRIFT with different sets of drift parameters n and m, and a drift threshold of θ=0.2.

As observed with the other post-processing filters, the average precision of the lexicons following POST-DRIFT is less than that achieved by standard WMEB. Interestingly, the POST-DRIFT filters which utilise evidence from the first 5 and 20 terms (n=5 and 20) are less effective than POST-SEED and POST-FIRST20, respectively. However, when more evidence is incorporated (n=100), POST-DRIFT outperforms both POST-SEED and POST-FIRST20. Increasing the number of recently added terms (m) also improves performance (when n is also high). As was observed with the other forms of post-processing, many incorrect terms are effectively removed. However, the underlying cause of semantic drift cannot be corrected.

Table 8.6 shows the results of incorporating the drift detection approach directly into WMEB. For each value of n and m tested, WMEB-DRIFT outperforms POST-DRIFT. As expected, values

Algorithm		1-200	201-400	401-600	601-800	801-1000	1-1000
WMEB		90.3	89.8	82.3	68.7	62.0	78.6
WMEB-DRIFT							
n	m						
5	5	89.5	82.8	74.6	71.4	63.2	76.3
5	20	89.6	83.7	74.4	70.8	62.3	76.9
5	50	89.5	84.2	74.0	71.4	67.3	77.3
5	100	89.3	83.0	73.4	72.1	66.6	76.9
20	5	90.8	85.8	79.4	73.3	78.0	81.5
20	20	90.6	86.4	79.9	74.0	76.4	81.4
20	50	90.5	86.2	79.3	75.4	74.7	81.2
20	100	90.6	86.1	79.0	76.3	74.5	81.3
100	5	90.5	87.3	82.0	74.6	79.8	82.8
100	20	90.5	87.3	81.3	73.2	77.1	81.9
100	50	90.5	87.1	81.2	72.9	77.1	81.7
100	100	90.6	87.0	81.1	72.6	76.4	81.5

Table 8.6: Inline semantic drift detection with WMEB (θ=0.2)

of n below 20 are inadequate. Even as more evidence is included by increasing the value of m, the overall performance remains below that of standard WMEB. These settings removed too many correct terms from the earlier iterations, which were not similar to the seeds. This significantly reduced the precision over the first 400 terms. Despite this, with n=5 the last 400 terms extracted are more precise than WMEB, indicating that the drift metric is successfully detecting drift in the later stages where it is most prominent.

When n is increased, the overall performance of WMEB-DRIFT significantly outperforms that of WMEB ($p \leq 0.0001$). Further, WMEB-DRIFT with n=100 is more effective than with n=20. This further supports my intuition that as bootstrapping progresses, candidate terms will not always be similar to the seeds or the terms extracted in the initial iterations.

With respect to the drift parameter m, it was expected that larger values would be more effective as they would introduce more similarity evidence. However, when drift is used inline, smaller values are most effective when $n \geq$20. When m=5, the drift metric considers a candidate term's similarity to those added in the previous iteration (5 new terms added per iteration). This implies that the drift metric can effectively detect semantic drift that has begun as early as in the last iteration.

The drift metric effectively combines the distributional similarity evidence for a given term with the first 20 or 100 terms and the most recently added terms. Each parameter combination significantly outperforms not only WMEB, but also WMEB-PASCA and WMEB-FIRST20 ($p \leq 0.0001$). The best performance gain was obtained with WMEB-DRIFT using similarity evidence from the first n=100 terms and the previous m=5 terms (+17.7% at 801-1000 and +4.2% at 1-1000). The best system also significantly outperforms the unsupervised bagging of WMEB with PDF500 (+1.8% at 1-1000; $p \leq 0.0001$, see Table 7.5).

8.5.3 Individual Categories

In this section, the performance gained with the best WMEB-DRIFT system is analysed further by investigating the individual categories. Table 8.7 presents the precision of the lexicons, in the later sections, extracted with WMEB and WMEB-DRIFT (θ=0.2, n=100, m=5).

The semantic categories, with little semantic drift in WMEB (PROTEIN, CELL LINE, and DRUG) were also extracted precisely with WMEB-DRIFT. Only one term, correctly identified as a PROTEIN, was filtered from the PROTEIN candidates, whereas 151 (142 correct) and 30 (29 correct) terms were removed from the CELL LINE and DRUG candidates, respectively. Despite these correct terms being filtered, the categories' precisions were not adversely affected.

The three categories that suffer substantially from semantic drift with WMEB (CELL, MUTATION and ANTIBODY) are all improved significantly with WMEB-DRIFT. As in the unsupervised bagging experiments, the CELL category improved the most with drift detection (+27.9% at 601-800 and +65.0% at 801-1000). Over the last 200 terms, WMEB-DRIFT improved upon WMEB+PDF500 with an additional 22.0% precision. WMEB-DRIFT filtered 456 potential CELL terms in total, of which 345 were incorrect.

The MUTATION category, which had a large precision drop, (-35.0% from 401-600 to 601-800), improved significantly with WMEB-DRIFT (+40.5% at 601-800 and +58.8% at 801-1000). The MUTATION lexicon extracted by WMEB had drifted initially into *amino acids*, but from 601-1000 the terms are predominantly verbs and adjectives. The inline drift filter

Category	WMEB		WMEB-DRIFT	
	601-800	801-1000	601-800	801-1000
ANTIBODY	75.0	40.0	80.5	87.5
CELL	21.5	0.0	49.4	65.0
CELL LINE	95.0	95.5	93.0	99.0
DISEASE	63.0	71.5	70.5	80.0
DRUG	95.5	95.0	95.5	94.0
FUNCTION	67.5	67.5	68.0	69.0
MUTATION	41.5	34.0	82.0	94.0
PROTEIN	100.0	99.0	100.0	99.0
SYMPTOM	68.0	57.0	63.0	58.5
TUMOUR	60.0	61.5	40.0	51.5
Average	68.7	62.1	74.6	79.8
Inv Rank	4.7	3.7	4.6	4.9

Table 8.7: MEDLINE individual category results for WMEB and WMEB-DRIFT (θ=0.2, n=100, m=5)

prevented 736 candidate terms from being selected. However, 675 of these were correct MUTATION terms. From visually inspecting the incorrect filtered terms, three specific semantic clusters were identified – *restriction enzymes*[4], *amino acids*, and *DNA modifying procedures*, with no verbs identified. Therefore, the drift filter effectively identified the initial drifting terms and in turn prevented the lexicon drifting completely into verbs and adjectives, despite filtering many MUTATION terms in the process.

The semantic drift occurring in the later stages of the ANTIBODY lexicon, was successfully detected and prevented using WMEB-DRIFT (+47.5% at 801-1000). In Section 8.4.2, the example of drift occurring in the ANTIBODY lexicon helped motivate my intuition for how to detect semantic drift within WMEB. Following on from the example, the incorrect terms *DFA*, *DIFA*, *FFA* and *MACRIA*, and the polysemous term *ICMA* are still extracted with WMEB-DRIFT, but in later stages.[5]

The 119 incorrect ANTIBODY terms extracted by WMEB within the last 200 terms are significantly prevented with WMEB-DRIFT — 108 of these terms are no longer extracted, and WMEB-DRIFT only extracted 25 incorrect terms in this range. Note also that none of the six

[4]Restriction enzymes cleave DNA at specific recognition nucleotide sequences known as restriction sites.
[5]*DFA* (rank: 95→102), *DIFA* (492→601), *FFA* (544→671), *MACRIA* (555→561) and *ICMA* (530→901).

Algorithm	Hand-picked	Average-10	Min.	Max.	s.d.
1-200					
WMEB	90.3	82.2	73.3	91.5	6.43
WMEB-DRIFT	90.5	84.5	76.8	90.9	4.94
401-600					
WMEB	82.3	66.7	61.2	74.4	4.60
WMEB-DRIFT	82.0	71.2	63.0	77.8	4.41
801-1000					
WMEB	61.5	57.8	52.2	66.1	5.09
WMEB-DRIFT	79.8	81.1	72.6	87.4	4.40

Table 8.8: Variation in precision of WMEB-DRIFT with random gold seeds (θ=0.2, n=100, m=5)

incorrect terms listed in Section 8.4.2, which are a result of drift, are extracted. Therefore, WMEB-DRIFT successfully identified semantic drift occurring within the ANTIBODY lexicon.

The TUMOUR category demonstrates that semantic drift is difficult to detect when it begins in the early iterations (1-200: 52.5%). Using WMEB-DRIFT, the TUMOUR lexicon is significantly less precise than standard WMEB. However, WMEB-DRIFT does significantly outperform unsupervised bagging (+23.5% at 801-1000; $p \leq 0.0001$). On the other hand, the SYMPTOM category was more precise with unsupervised bagging than WMEB-DRIFT (+5.5% at 801-1000; $p \leq 0.0001$).

8.5.4 Random Seed Evaluation

In the final experiments, the performance of the best performing WMEB-DRIFT system (θ=0.2, n=100, m=5) using the 10 random UNION gold seed sets from Section 7.2 are reported. These results are shown in Table 8.8. In the initial stages the hand-picked seeds extracted more precise lexicons than the random seeds on average with both WMEB and WMEB-DRIFT. With the hand-picked seeds, there is also little difference between WMEB and WMEB-DRIFT over the 1-200 and 401-600 term ranges. However, the differences between the average-10 scores are significantly greater with WMEB-DRIFT outperforming WMEB (+4.5% at 401-600, $p \leq 0.0001$). Further, over the last 200 terms where semantic drift is prominent, WMEB-DRIFT on average is significantly more precise than WMEB (+23.3% at 801-1000, $p \leq 0.0001$).

These experiments demonstrate that the filter threshold chosen has not been an over-fit to the initial lexicons extracted with the hand-picked seeds. Although the threshold was set empirically based on WMEB's initial lexicons, it effectively identified drifting terms regardless of the initial seeds. These results further confirm that WMEB-DRIFT effectively identifies and prevents the semantic drift that occurs with WMEB.

8.5.5 Future Work

The experiments in this chapter suggest a number of avenues to be explored. Firstly, WMEB-DRIFT can be extended to remove the extracted terms that started the semantic drift. WMEB-DRIFT currently prevents the extraction of candidate terms that are more similar to a group of recently extracted terms than to the terms extracted in the earlier iterations. Based on my observations and the effectiveness of the drift metric, I hypothesise that the recently extracted terms which are similar to a drifting candidate term are largely responsible for the semantic drift, and are thus polysemous or incorrect. This suggests that semantic drift can be further reduced by backtracking WMEB-DRIFT to remove those terms from the lexicon and any new patterns they identified. This will prevent the terms and patterns from contributing further to the bootstrapping process, and thus to semantic drift.

Currently, the drift parameters are identical for each semantic category. I would like to explore the impact of varying these parameters for different categories. In particular, I would like to investigate the TUMOUR category further to see if more suitable parameters can prevent it drifting. Finally, based on the significant improvements achieved by unsupervised bagging in Chapter 7, I would also like to apply an ensemble of WMEB-DRIFT bootstrappers to correct the semantic drift further.

8.6 Summary

This chapter introduced the distributional similarity approach for extracting semantic lexicons and comparing it to WMEB. The results showed that distributional similarity approaches can outperform WMEB in the initial stages for a limited number of categories. Overall WMEB is more precise, especially when extracting lexicons of 500 terms or more. Further analysis of the different lexicons indicated that these approaches are complementary, which suggests that WMEB may be able to exploit distributional similarity techniques to reduce semantic drift.

The next set of experiments utilised distributional similarity to filter terms extracted by WMEB that are dissimilar to the seeds or the terms extracted in the first few iterations. These simple filters were tested as a post-processing step and as an inline filter within WMEB. The post-processing filters were ineffective, while the inline filters significantly outperformed WMEB. Despite this, these filters are sensitive to the thresholds, with only slight changes generating significantly different results. It is also difficult to select a threshold that is suitable for all categories. Further, these filters remove many correct terms as they are based on the assumption that terms with little or no similarity to the initial terms are incorrect.

These observations helped formulate the hypothesis of this chapter, which states that semantic drift occurs when a new candidate term to be added is more similar to a group of recently extracted terms than to the seeds and/or high precision terms extracted in the early iterations. To test this hypothesis, I devised a novel drift metric that exploits distributional similarity measurements to detect this behavior and drifting terms. And my preliminary analyses showed that this metric can effectively separate the correct terms from the terms that are responsible for, or a consequence of semantic drift.

The drift metric was first considered as a post-processing filter over the lexicons extracted by WMEB (POST-DRIFT). Like the other post-processing filters, the POST-DRIFT lexicons were less precise than that of WMEB. In POST-DRIFT, many incorrect terms are filtered. However, the principal cause of semantic drift is not prevented. On the other hand, when drift detection is incorporated directly into WMEB (WMEB-DRIFT), semantic drift is significantly reduced.

The experiments demonstrate that WMEB-DRIFT effectively detects and prevents drifting terms from being extracted and influencing the bootstrapping process. In particular, WMEB-DRIFT improves the precision of the CELL (+65.0%), MUTATION (+58.8%) and ANTIBODY (+47.5%) lexicons greatly over the last 200 terms. Further, WMEB-DRIFT significantly outperforms standard WMEB by 17.7% and unsupervised bagging of WMEB by 9.6% on average over the last 200 terms.

Chapter 9

Conclusion

This thesis demonstrated the need for biomedical lexical-semantic resources, and presented novel domain-independent bootstrapping solutions for automatically extracting them.

Minimally supervised bootstrapping algorithms are attractive for this task. However, they are prone to semantic drift, which prevents the extraction of large yet precise lexicons. In this thesis, I have developed new methods for reducing semantic drift — from improved candidate term and pattern selection methods (WMEB), to meta approaches for correcting drift within extracted lexicons (bagging) and detecting and preventing drift during the bootstrapping process using distributional similarity. The effectiveness of these techniques is demonstrated within the biomedical domain. Each method extracts significantly more precise semantic lexicons than the existing state-of-the-art systems.

None of these techniques are specific to the biomedical domain. They do not rely on domain specific knowledge, apart from the initial seed terms, and can be applied to raw text. Therefore, these methods are domain independent and will thus be beneficial to wider NLP.

This thesis began by exploring the various linguistic phenomena within biomedical abstracts and full texts that may impair an IE system and investigating the advantages of full-text processing. In Chapter 2, I presented the *Molecular Interaction Map* (MIM) corpus created primarily for this purpose. The MIM corpus is a unique resource that maps interaction facts to passages within full-text articles. These passages are annotated for coreference and negated expressions, as well as synonym and extra fact dependencies, which must be resolved for the interaction

fact to be extracted. The MIM corpus complements existing biomedical corpora that consist of sentences or abstracts annotated with named-entities, the relationships between them, and syntactic analyses, and provides invaluable insight into the challenges of automatically extracting biomedical knowledge from full-text.

Chapter 3 presented a detailed evaluation of the MIM corpus. This analysis demonstrated that full-text processing is necessary to extract interaction facts — interactions are most commonly stated in the results section of articles, and often depend on information, such as synonyms, defined in other sections. The evaluation also gave the first insight into the degree of fact redundancy within biomedical articles. These results indicate that IE systems will benefit from processing full-text rather than only abstracts.

The analysis also showed the importance of tasks for biomedical IE systems that are often neglected — resolving negated and coreference expressions, and identifying synonym and extra fact dependencies. The importance of resolving anaphoric expressions and extra fact dependencies was further demonstrated with oracle sentence retrieval experiments using the MIM corpus as a gold-standard test set. For example, more than 35% of false negative instances are due to extra fact dependencies. Each of these tasks can be improved with lexical-semantic resources. Unfortunately, the currently available biomedical resources are limited in size and biased towards specific topics within biomedicine. These results provided the motivation for developing sophisticated bootstrapping methods for extracting large precise biomedical semantic lexicons from raw-text, to help overcome this knowledge bottleneck.

Chapter 4 surveyed the existing minimally supervised approaches for extracting semantic lexicons, which extend the influential pattern-based approaches. Multiple bootstrapping algorithms were described, including the single-category and multi-category frameworks. The multi-category bootstrappers, MEB, BASILISK and NOMEN, were specifically developed to reduce semantic drift of the lexicons by utilising information about other competing semantic categories. These algorithms differ primarily by the types of patterns used, how they score candidate terms and patterns, and incorporate knowledge from multiple semantic categories. Although these algorithms significantly outperform the single-category bootstrappers, they still

suffer from semantic drift, preventing them from extracting large yet precise biomedical semantic lexicons.

In Chapter 5, I detailed the evaluation methodology used throughout this thesis to measure the quality of the biomedical lexicons and their patterns extracted from raw biomedical text. The biomedical semantic categories of interest and the document sets their lexicons are extracted from are described. Due to the lack of coverage of the available biomedical gold-standard corpora and resources, the extracted lexicons are evaluated manually to ensure accurate evaluation. I presented the guidelines for this evaluation process and an analysis of the reliability of this approach. The inter-evaluator agreement between two domain experts demonstrated near perfect agreement across the categories. Chapter 5 also presented the guidelines for judging the quality of the patterns used in bootstrapping.

In Chapter 6, I proposed a new minimally supervised multi-category bootstrapping algorithm, *Weighted Mutual Exclusion Bootstrapping* (WMEB), for extracting larger biomedical lexicons. The chapter began by identifying characteristics of the previous approaches that lead to semantic drift in the early stages, preventing the extraction of precise lexicons. The development of WMEB focused on these issues, with an emphasise on reducing semantic drift.

WMEB extends the mutual exclusion framework of MEB, by incorporating a cumulative pattern pool and a new term and pattern weighting scheme. The pattern pool iteratively accumulates the top patterns from previous iterations to ensure they can contribute in later iterations. The term weighting, which is symmetrical for patterns, assumes that terms which are highly correlated with the extracting patterns are more precise. I investigated numerous measures of association strength, and demonstrated that the χ^2 statistic is most effective.

The experiments in Chapter 6, exhaustively compared the performance of WMEB with MEB and BASILISK, and demonstrated that WMEB significantly outperforms MEB and BASILISK. Over the first 500 terms extracted, WMEB achieves an average precision of 88.7%, and is significantly more precise than MEB (59.6%) and BASILISK (70.6%). These results indicated that the cumulative pattern pool and new term and pattern weighting within WMEB are effective at reducing semantic drift.

Although WMEB is less susceptible to semantic drift than the existing approaches, drift still has a major significant impact in the later iterations. To address this further, I focused on investigating methods for correcting and preventing semantic drift that can be applied to any bootstrapping algorithm.

Chapter 7 argued that the standard approach for evaluating bootstrapping algorithms using only one set of seeds is unreliable. This motivated a new evaluation methodology which compares the sensitivity of algorithms to random seed sets. This approach also ensures that bootstrappers are not tuned to the seeds. Based on this evaluation approach, it is confirmed that WMEB is indeed significantly less susceptible to semantic drift than BASILISK and MEB. Further, BASILISK was shown to be far more sensitive to the input seeds than both MEB and WMEB, generating very diverse lexicons.

The observed performance variations and sensitivity of each of the bootstrappers motivated using an ensemble of bootstrappers seeded with random seeds to correct semantic drift within the extracted lexicons. The process of aggregating the lexicons for a category is based on the hypothesis that terms extracted in the earlier iterations by multiple bootstrappers are more likely to be correct lexicon members.

Supervised bagging was shown to be most effective for algorithms that are very sensitive to the seeds and more susceptible to semantic drift, such as MEB and BASILISK. For these algorithms, supervised bagging effectively corrected many of the errors within the lexicons.

Unfortunately, supervised bagging negates one of the main advantages of bootstrapping by requiring thousands of gold seeds. This motivated the development of a novel unsupervised bagging approach that requires only one set of seeds. In unsupervised bagging, the random seeds are sampled from the initial lexicons extracted using the original hand-picked seeds. My results demonstrated that unsupervised bagging of WMEB and MEB can significantly outperform the standard bootstrapping algorithms, and even supervised bagging of WMEB, when suitable unsupervised seeds are available.

Chapter 8 hypothesised that semantic drift occurs when a candidate term is more similar to the group of recently added terms than to the seed and/or high precision terms extracted

in the earlier iterations. The chapter first introduced distributional similarity, which is also often used to extract semantic lexicons. Initial experiments showed that although distributional similarity is less precise than WMEB, they produce complementary results. This suggested that distributional similarity may be incorporated within WMEB to detect semantic drift.

I proposed a novel drift metric which reflects the variability of a candidate term's degree of similarity to the initial terms within the lexicon and those recently extracted. A candidate term's degree of drift is defined as the ratio of its average distributional similarity to the set of first n terms extracted into the lexicon, and its average similarity to the last m terms added in the previous iterations. Smaller values of drift correspond to the term moving further away from the first set of terms. To evaluate the effectiveness of my metric, it was introduced directly into WMEB's term selection phase to detect drifting terms (WMEB-DRIFT). Candidate terms with a drift score below a specified threshold are not extracted into the lexicon. This prevents the drifting terms from influencing the subsequent bootstrapping iterations, and in turn reduces semantic drift.

The results demonstrated that the semantic drift introduced by WMEB is significantly reduced with this additional criteria. WMEB-DRIFT significantly outperforms both supervised and unsupervised bagging of WMEB. To confirm these observations, I also performed a final random seed evaluation. These results further demonstrated the effectiveness of the drift metric and that the threshold selected was not an over-fit to the hand-picked seeds. These results also confirmed that the initial hypothesis is valid.

Automatically extracting semantic lexicons from text is critical for improving many NLP applications. This thesis presented three novel approaches for reducing semantic drift in semantic lexicon bootstrapping, allowing larger more precise lexicons to be automatically extracted. I demonstrated the effectiveness of these techniques within the biomedical domain and showed that they significantly outperform the existing approaches. My algorithms extract new terms from raw-text using window-based contextual patterns, without any bio-specific information, and are thus domain-independent. Therefore, the results of this work can be exploited to extract semantic lexicons for other domains and contribute to advancing many NLP tasks.

References

Abney, Steven. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344, December 1996.

Agichtein, Eugene and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94, San Antonio, TX, USA, June 2000.

Ando, Rie Kubota. Semantic lexicon construction: Learning from unlabeled data via spectral analysis. In *Proceedings of the 8th Conference on Computational Natural Language Learning*, pages 9–16, Boston, MA, USA, May 2004.

Ando, Rie Kubota. BioCreative II gene mention tagging system at IBM Watson. In *Proceedings of the Second BioCreative Challenge Evaluation*, pages 101–103, Madrid Spain, April 2007.

Ao, Hiroko and Toshihisa Takagi. ALICE: An algorithm to extract abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 12(5):576–586, 2005.

Ashburner, Michael and Rachel Drysdale. FlyBase — the Drosophila genetic database. *Development*, 120:2077–2079, 1994.

Bader, Gary D., Ian Donaldson, Cheryl Wolting, B. F. Francis Ouellette, Tony Pawson, and Christopher W. V. Hogue. BIND - The Biomolecular Interaction Network Database. *Nucleic Acids*, 29(1):242–245, 2001.

Barzilay, Regina and Michael Elhadad. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS-97)*, pages 10–17, Madrid, Spain, July 1997.

Batschè, Eric, Christian Muchardt, Jürgen Behrens, Helen C. Hurst, and Chantal Crémisi. RB and c-Myc activate expression of the E-cadherin gene in epithelial cells through interaction with transcription factor AP-2. *Molecular and Cellular Biology*, 18(7):3647–58, July 1998.

BD Biosciences. *Introduction to Flow Cytometry: A Learning Guide*. BD Biosciences, San Jose, CA, USA, April 2002. Manual Part Number: 11-11032-01.

Berland, Matthew and Eugene Charniak. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 57–64, College Park, MD, USA, June 1999.

Björne, Jari, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on BioNLP: Shared Task*, pages 10–18, Boulder, CO, USA, June 2009.

Brants, Thorsten and Alex Franz. Web 1T 5-gram version 1. Technical Report LDC2006T13, Linguistics Data Consortium, 2006.

Breiman, Leo. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

Brill, Eric, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. Data-intensive question answering. In *Proceedings of the 10th Text REtrieval Conference (TREC 2001)*, Gaithersburg, MD, USA, 2001.

Brin, Sergey. Extracting patterns and relations from the world wide web. In *Proceedings of the WebDB Workshop at Extending Database Technology (EDBT)*, November 1998.

Brin, Sergey, Rajeev Motwani, Lawrence Page, and Terry Winograd. What can you do with a web in your pocket? *Data Engineering Bulletin*, 21(2):37–47, 1998.

Briscoe, Ted and John Carroll. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, Las Palmas de Gran Canaria, May 2002.

Brody, Samuel, Roberto Navigli, and Mirella Lapata. Ensemble methods for unsupervised WSD. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 97–104, Sydney, Australia, July 2006.

Bunescu, Razvan, Raymond Mooney, Arun Ramani, and Edward Marcotte. Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from medline. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06*, pages 49–56, New York City, NY, USA, June 2006.

Caporaso, J. Gregory, William A. Baumgartner Jr, David A. Randolph, K. Bretonnel Cohen, and Lawrence Hunter. Rapid pattern development for concept recognition systems: Application to point mutations. *Bioinformatics and Computational Biology*, 5(6):1233–1259, 2007.

Carletta, Jean. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254, 1996.

Castaño, José, Jason Zhang, and James Pustejovsky. Anaphora resolution in biomedical literature. In *Proceedings of the International Symposium on Reference Resolution for Natural Language Processing*, Alicante, Spain, June 2002.

Clarke, Charles L. A., Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 358–365, New Orleans, LA, USA, September 2001.

Cohen, Paul R. *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1995.

Corbett, Peter, Colin Batchelor, and Simone Teufel. Annotation of chemical named entities. In *Proceedings of BioNLP 2007: Biological, translational, and clinical language processing*, pages 57–64, Prague, Czech Republic, June 2007.

Curran, James R. *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh, 2004.

Curran, James R., Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 172–180, Melbourne, Australia, September 2007.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 164–171, Columbus, OH, USA, June 1993.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. *Computer, Speech and Language*, 9:123–152, 1995.

Daraselia, Nikolai, Sergei Egorov, Andrey Yazhuk, Svetlana Novichkova, Anton Yuryev, , and Ilya Mazo. Extracting protein function information from MEDLINE using a full-sentence parser. In *Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, pages 15–21, Pisa, Italy, September 2004.

Day, David, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. Mixed-initiative development of language processing systems. In *Proceedings of the Fifth ACL Conference on Applied Natural Language Processing*, pages 348–356, Washington, DC, USA, April 1997.

Dietterich, Thomas G. Ensemble methods in machine learning. *Lecture Notes In Computer Science, Proceedings of the First International Workshop on Multiple Classifier Systems*, 1857:1–15, 2000.

Donaldson, Ian, Joel Martin, Berry de Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D. Bader, Katerina Michalickova, Tony Pawson, and Christopher W. V. Hogue. PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4(11), 2003.

Durme, Benjamin Van, Ting Qian, and Lenhart Schubert. Class-driven attribute extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 921–928, Manchester, UK, August 2008.

Eisner, Jason and Damianos Karakos. Bootstrapping without the boot. In *Proceedings of the Conference on Human Language Technology and Conference on Empirical Methods in Natural Language Processing*, pages 395–402, Vancouver, British Columbia, Canada, October 2005.

Ellman, Jeremy. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Boston, MA, USA, 1998.

Erdogmus, Muge and Osman Ugur Sezerman. Application of automatic mutation-gene pair extraction to diseases. *Bioinformatics and Computational Biology*, 5(6):1261–1275, 2007.

Etzioni, Oren, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, June 2005.

Fano, Robert M. *Transmission of Information: A Statistical Theory of Communications*. MIT Press, Cambridge, MA, USA, 2nd printing with corrections edition, 1963.

Fellbaum, Christiane. WordNet: An electronic lexical database. MIT Press, Cambridge, MA, USA, 1998.

Florian, Radu, Silvio Cucerzan, Charles Schaffer, and David Yarowsky. Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4):327–341, 2002.

Florian, Radu, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, Edmonton, Canada, May/June 2003.

Friedman, Carol, Pauline Kra, Hong Yu, Michael Krauthammer, and Andrey Rzhetsky. GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(suppl. 1):74–82, 2001.

Gale, William A., Kenneth W. Church, and David Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language at the Human Language Technology Conference*, pages 233–237, Harriman, NY, USA, 23-26 February 1992.

Garcí-Varea, Ismael, Franz J. Och, Hermann Ney, and Francisco Casacuberta. Refined lexicon models for statistical machine translation using a maximum entropy approach. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 204–211, Toulouse, France, July 2001.

Geffet, Maayan and Ido Dagan. Feature vector quality and distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, August 2004.

Gonzalo, Julio, Felisa Verdejo, and Irina Chugu. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING/ACL 98 Workshop on Usage of WordNet for NLP*, pages 38–44, Montreal, Quebec, Canada, August 1998.

Grefenstette, Gregory. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston, MA, USA, 1994.

Grishman, Ralph and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational Linguistics (COLING)*, pages 466–471, Copenhagen, Denmark, 1996.

Grolier. *Academic American Encyclopedia*. Grolier Electronic Publishing, Danbury, CT, USA, 1990.

Grover, Claire, Michael Matthews, and Richard Tobin. Tools to address the interdependence between tokenisation and standoff annotation. In *Proceedings of the Multi-dimensional Markup in Natural Language Processing Workshop*, Trento, Italy, April 2006.

Grover, Claire, Barry Haddow, Ewan Klein, Michael Matthews, Leif Nielsen, Richard Tobin, and Xinglong Wang. Adapting a relation extraction pipeline for the BioCreAtIvE II tasks. In *Proceedings of the Second BioCreative Challenge Workshop*, pages 273–286, Madrid, Spain, April 2007.

GuoDong, Zhou, Shen Dan, Zhang Jie, Su Jian, Tan Soon Heng, and Tan Chew Lim. Recognition of protein/gene names from text using an ensemble of classifiers and effective abbreviation resolution. In *Proceedings of the EMBO Workshop 2004 on a critical assessment of text mining methods in molecular biology*, Granada, Spain, March 2004.

Harris, Zellig. Distributional structure. *Word*, 10(23):146–162, 1954.

Hatzivassiloglou, Vasileios, Pablo A. Dubouè, and Andrey Rzhetsky. Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics*, 17(1):S97–S106, 2001.

Hearst, Marti A. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545, Nantes, France, August 1992.

Hersh, William, Aaron M. Cohen, Lynn Ruslen, and Phoebe M. Roberts. TREC 2007 Genomics track overview. In *Proceedings of the 16th Text Retrieval Conference*, Gaithersburg, MD, USA, November 2007.

Hickl, Andrew, Kirk Roberts, Bryan Rink, Jeremy Bensley, Tobias Jungen, Ying Shi, and John Williams. Question Answering with LCC's CHAUCER-2 at TREC 2007. In *Proceedings of the 16th Text REtrieval Conference*, Gaithersburg, MD, USA, November 2007.

Hindle, Donald. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275, Pittsburgh, PA, USA, 6-9 June 1990.

Hirschman, Lynette, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of BioCreAtIvE: Critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl. 1), 2005.

Holmes, Janet. Doubt and certainty in ESL textbooks. *Applied Linguistics*, 9(1):21–44, 1988.

Hovy, Eduard, Zornitsa Kozareva, and Ellen Riloff. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 948–957, Suntec, Singapore, August 2009.

Humphreys, Kevin, Robert Gaizauskas, and Saliha Azzam. Event coreference for information extraction. In *Proceedings of the ACL/EACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Madrid, Spain, July 1997.

Hunter, Lawrence, Zhiyong Lu, James Firby, William A. Baumgartner, Helen L. Johnson, Philip V. Ogren, and K. Bretonnel Cohen. OpenDMAP: An open-source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-specific gene expression. *BMC Bioinformatics*, 9(10), 2008.

Hyland, Ken. Writing without conviction? Hedging in science research articles. *Applied Linguistics*, 17(4):433–454, 1996.

Jones, Rosie, Andrew McCallum, Kamal Nigam, and Ellen Riloff. Bootstrapping for text learning tasks. In *Proceedings of the IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, Stockholm, Sweden, August 1999.

Kanagasabai, Rajaraman, Khar Heng Choo, Shoba Ranganathan, and Christopher J. O. Baker. A workflow for mutation extraction and structure annotation. *Journal of Bioinformatics and Computational Biology*, 5(6):1319–1337, 2007.

Kanehisa, Minoru and Susumu Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

Karamanis, Nikiforos, Ruth Seal, Ian Lewin, Peter McQuilton, Andreas Vlachos, Caroline Gasperin, Rachel Drysdale, and Ted Briscoe. Natural language processing in aid of FlyBase curators. *BMC Bioinformatics*, 9(193), 2008.

Kim, Jin-Dong, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl. 1):i180–i182, 2003.

Kim, Jin-Dong, Tomoko Ohta, and Jun'ichi Tsujii. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(10), 2008.

Kim, Jin-Dong, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on BioNLP: Shared Task*, pages 1–9, Boulder, CO, USA, June 2009.

Kohn, Kurt W. Molecular interaction map of the mammalian cell cycle and DNA repair systems. *Molecular Biology of the Cell*, 10:2703–2734, 1999.

Landis, J. Richard and Gary G. Koch. The measurement of observer agreement in categorical data. *Biometrics*, 33(1):159–174, 1977.

LCC. CiceroLite. Language Computer Corporation (LCC), 2009.

LDC. North American News Text Corpus. LDC Catalog No: LDC95T21, 1995.

Lee, Lillian. *Similarity-Based Approaches to Natural Language Processing*. published as tr-11-97, Harvard University, Cambridge, MA, USA, 1997.

Lee, Lillian. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 23–32, College Park, MD, USA, June 1999.

Lee, Lillian and Fernando Pereira. Distributional similarity models: Clustering vs. nearest neighbors. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 20–26, College Park, MD, USA, June 1999.

Lee, Seungwoo and Gary G. Lee. Exploring phrasal context and error correction heuristics in bootstrapping for geographic named entity annotation. *Information Systems*, 32:575–592, 2007.

Light, Marc, Xin Ting Qui, and Padmini Srinivasan. The language of bioscience: Facts, speculations, and statements in between. In *Proceedings of BioLink 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*, pages 17–24, Boston, MA, USA, May 2004.

Lin, Dekang. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–71, Madrid, Spain, July 1997.

Lin, Dekang. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, pages 768–774, Montreal, Quebec, Canada, August 1998a.

Lin, Dekang. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, Madison, WI, USA, July 1998b.

Lin, Dekang. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems*, pages 234–241, Granada, Spain, May 1998c.

Lin, Dekang and Patrick Pantel. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360, 2001.

Lin, Dekang, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 2003a.

Lin, Winston, Roman Yangarber, and Ralph Grishman. Bootstrapped learning of semantic classes from positive and negative examples. In *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, pages 103–111, August 2003b.

Liu, Alvin Y., Randy R. Robinson, E. David Murray Jr, Jeffrey A. Ledbetter, Ingegerd Hellström, and Karl Erik Hellström. Production of a mouse-human chimeric monoclonal antibody to CD20 with potent Fc-dependent biologic activity. *Journal of Immunology*, 139(10): 3521–3526, 1987.

Lussier, Yves, Tara Borlawsky, Daniel Rappaport, Yang Liu, and Carol Friedman. PHENOGO: Assigning phenotypic context to gene ontology annotations with natural language processing. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 64–75, Maui, HI, USA, January 2006.

Manning, Christopher D. and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

McCarthy, Diana, Bill Keller, and John Carroll. Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 73–80, Sapporo, Japan, July 2003.

McCrae, John and Nigel Collie. Synonym set extraction from the biomedical literature by lexical pattern discovery. *BMC Bioinformatics*, 9(19), 2008.

McIntosh, Tara and James R. Curran. Challenges for extracting biomedical knowledge from full text. In *Proceedings of BioNLP 2007: Biological, translational, and clinical language processing*, pages 171–178, Prague, Czech Republic, July 2007a.

McIntosh, Tara and James R. Curran. Sentence retrieval for extracting biomedical knowledge. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 342–349, Melbourne, Australia, September 2007b.

McIntosh, Tara and James R. Curran. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 97–105, Hobart, Australia, December 2008.

McIntosh, Tara and James R. Curran. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 396–404, Suntec, Singapore, August 2009a.

McIntosh, Tara and James R. Curran. Challenges for automatically extracting molecular interactions from full-text articles. *BMC Bioinformatics*, 10(311), September 2009b.

Meij, Edgar and Sophia Katrenko. Bootstrapping language associated with biomedical entities. The AID group at TREC Genomics 2007. In *Proceedings of the 16th Text Retrieval Conference*, Gaithersburg, MD, USA, November 2007.

Mercer, Robert E. and Chrysanne Di Marco. A design methodology for a biomedical literature indexing tool using the rhetoric of science. In *Proceedings of BioLink 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*, pages 77–84, Boston, MA, USA, May 2004.

Mikheev, Andrei, Claire Grover, and Marc Moens. Description of the LTG system used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, VA, USA, April 1998.

Mirkin, Shachar, Ido Dagan, and Maayan Geffet. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 579–586, Sydney, Australia, July 2006.

Mitkov, Ruslan. Factors in anaphora resolution: they are not the only things that matter. A case study based on two different approaches. In *Proceedings of the ACL/EACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution*, pages 14–21, Madrid, Spain, July 1997.

Moldovan, Dan, Christine Clark, and Mitchell Bowden. Lymba's PowerAnswer 4 in TREC 2007. In *Proceedings of the 16th Text REtrieval Conference*, Gaithersburg, MD, USA, November 2007.

Murphy, Tara and James R. Curran. Experiments in mutual exclusion bootstrapping. In *Proceedings of the Australasian Language Technology Workshop*, pages 66–74, Melbourne, Australia, December 2007.

Murphy, Tara, Tara McIntosh, and James R. Curran. Named entity recognition for astronomy literature. In *Proceedings of the Australasian Language Technology Workshop*, pages 59–66, Sydney, Australia, November/December 2006.

Ng, Vincent and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, PA, USA, July 2002.

NLM. Unified Medical Language System (UMLS). U.S. National Library of Medicine, 2006.

NLM. Medical Subject Headings (MeSH). U.S. National Library of Medicine., 2008.

Noreen, Eric W. *Computer Intensive Methods for Testing Hypotheses: An Introduction*. John Wiley & Sons, New York, NY, USA, 1989.

Ohta, Tomoko, Yuka Tateisi, and Jin-Dong Kim. GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 73–77, San Diego, CA, USA, March 2002.

Ohta, Tomoko, Yusuke Miyao, Takashi Ninomiya, Yoshimasa Tsuruoka, Akane Yakushiji, Katsuya Masuda, Jumpei Takeuchi, Kazuhiro Yoshida, Tadayoshi Hara, Jin-Dong Kim, Yuka Tateisi, and Jun'ichi Tsujii. An intelligent search engine and GUI-based efficient Medline search tool based on deep syntactic parsing. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 17–20, Sydney, Australia, July 2006.

Okazaki, Naoaki and Sophia Ananiadou. A term recognition approach to acronym recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 643–650, Sydney, Australia, July 2006.

Paşca, Marius. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 683–690, Lisbon, Portugal, November 2007.

Paşca, Marius and Sanda M. Harabagiu. The informative role of wordnet in open-domain question answering. In *Proceedings of the Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pages 138–143, Pittsburgh, PA, USA, June 2001.

Paşca, Marius, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 809–816, Sydney, Australia, July 2006.

Padó, Sebastian and Mirella Lapata. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Sapporo, Japan, July 2003.

Pantel, Patrick and Dekang Lin. Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 613–619, Edmonton, Alberta, Canada, July 2002.

Pantel, Patrick and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia, July 2006.

Pantel, Patrick and Deepak Ravichandran. Automatically labelling semantic classes. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 321–328, Boston, MA, USA, May 2004.

Pantel, Patrick, Deepak Ravichandran, and Eduard Hovy. Towards terascale knowledge acquisition. In *The 20th International Conference on Computational Linguistics*, pages 771–777, Geneva, Switzerland, August 2004.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, OH, USA, June 1993.

ProMED. ProMED-Mail. The global electronic reporting system of emerging infectious diseases and toxins. International Society of Infectious Diseases. www.promedmail.org/pls/promed/promed.home, 2003.

Pyysalo, Sampo. *A Dependency Parsing Approach to Biomedical Text Mining*. PhD thesis, Turku Centre for Computer Science, University of Turku, August 2008.

Pyysalo, Sampo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50), 2007.

Ravichandran, Deepak and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 41–47, Philadelphia, PA, USA, July 2002.

Regev, Yizhar, Michal Finkelstein-Langau, Ronen Feldman, Mayo Gorodetsky, Xin Zheng, Samuel Levy, Rosane Charlab, Charles Lawrence, Ross A. Lippert, Qing Zhang, and Hagit Shatkay. Rule-based extraction of experimental evidence in the biomedical domain - the KDD Cup 2002 (Task 1). *ACM SIGKDD Explorations*, 4(2):90–92, 2002.

Reimlinger, M., R. Hoffmann, C. Pfeil-Putzien, and P. Scheinert. Enrofloxacin, a new drug for ornamental fish. *Tierärztliche Praxis*, 18(6):653–657, 1990.

Reynar, Jeffrey C. and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 16–19, Washington, DC, USA, March/April 1997.

Riloff, Ellen. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049, Portland, OR, USA, August 1996a.

Riloff, Ellen. An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, 85:101–134, 1996b.

Riloff, Ellen and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference*, pages 474–479, Orlando, FL, USA, July 1999.

Riloff, Ellen and Jessica Shepherd. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124, Providence, RI, USA, August 1997.

Ripley, Brian D. *Stochastic Simulation*. John Wiley and Sons, 1987.

Roark, Brian and Eugene Charniak. Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 1110–1116, Montreal, Quebec, Canada, August 1998.

Schuemie, M. J., M. Weeber, B. J. A. Schijvenaars, E. M. van Mulligen, C. C. van der Eijk, R. Jelier, B. Mons, and J. A. Kors. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, 20(16):2597–2604, 2004.

Sehgal, Aditya K., Padmini Srinivasan, and Olivier Bodenreider. Gene terms and English words: An ambiguous mix. In *Proceedings of the SIGIR 2004 Workshop on Search and Discovery for Bioinformatics*, Sheffield, UK, July 2004.

Sekine, Satoshi and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal, May 2004.

Shaffer, Lisa G., Marilyn L. Slovak, and Lynda J. Campbell, editors. *ISCN An International System for Human Cytogenetic Nomenclature: Recommendations of the International Standing Committee on Human Cytogenetic Nomenclature*. S Karger with Cytogenetic and Genome Research, third edition, 2009.

Shah, Parantu K., Carolina Perez-Iratxeta, Peer Bork, and Miguel A. Andrade. Information extraction from full text scientific articles: Where are the keywords? *BMC Bioinformatics*, 4(20), 2003.

Siegel, Sidney and N. John Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York, second edition, 1988.

Sinclair, Gail and Bonnie Webber. Classification from full text: A comparison of canonical sections of scientific papers. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 66–69, Geneva, Switzerland, August 2004.

Smeaton, Alan F., Fergus Kelledy, and Ruari O'Donnell. TREC-4 experiments at Dublin City University: Thresholding posting lists, query expansion with WordNet and POS tagging of spanish. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, Gaithersburg, MD, USA, November 1995.

Soubbotin, Martin M. and Sergei M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 175–182, Gaithersburg, MD, USA, 2001.

Stevenson, Mark and Robert Gaizauskas. Using corpus-derived name lists for named entity recognition. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, pages 290–295, Seattle, WA, USA, May 2000.

Sundheim, Beth M. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Conference on Message Understanding*, pages 3–21, McLean, VA, USA, June 1992.

Tanabe, Lorraine, Natalie Xie, Lynne H. Thom, Wayne Matten, and W. John Wilbur. GENETAG: A tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6 (Suppl 1 (S3)), 2005.

NLM. National Library of Medicine: MEDLINE Fact Sheet. U.S. National Library of Medicine., 2009.

Thelen, Michael. Simultaneous generation of domain-specific lexicons for multiple semantic categories. Master's thesis, University of Utah, 2001.

Thelen, Michael and Ellen Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 214–221, Philadelphia, PA, USA, July 2002.

Toutanova, Kristina, Hisami Suzuki, and Achim Ruopp. Applying morphology generation models to machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 514–522, Columbus, OH, USA, June 2008.

Tweedie, Susan, Michael Ashburner, Kathleen Falls, Paul Leyland, Peter McQuilton, Steven Marygold, Gillian Millburn, David Osumi-Sutherland, Andrew Schroeder, Ruth Seal, Haiyan Zhang, and The FlyBase Consortium. FlyBase: Enhancing Drosophila Gene Ontology annotations. *Nucleic Acids Research*, 37(D555-D559), 2009.

Uetz, Peter, Johannes Goll, and Jakob Hallermann. The TIGR Reptile Database, 2009. URL `http://www.reptile-database.org`. accessed August 10, 2009.

Vincze, Veronika, György Szarvas, Richaŕd Farkas, György Mó, and János Csirik. The BioScope corpus: Biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(Suppl 11), 2008.

Vlachos, Andreas, Caroline Gasperin, Ian Lewin, and Ted Briscoe. Bootstrapping the recognition and anaphoric linking of named entities in drosophila articles. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 11, pages 100–111, Maui, HI, USA, January 2006.

Voorhees, Ellen M. Overview of the question answering track. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 157–165, Gaithersburg, MD, USA, 2001.

Weeds, Julie. *Measures and Applications of Lexical Distributional Similarity*. PhD thesis, Department of Informatics, University of Sussex, 2003.

Weeds, Julie and David Weir. A general framework for distributional similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Sapporo, Japan, July 2003.

Weischedel, Ralph and Ada Brunstein. BBN pronoun coreference and entity type corpus. Technical Report LDC2005T33, Linguistics Data Consortium, 2005.

Wilbur, W. John, Larry Smith, and Lorraine Tanabe. Biocreative 2. gene ention task. In *Proceedings of the Second BioCreative Challenge Evaluation*, pages 7–16, 2007.

Xenarios, Ioannis, Danny W. Rice, Lukasz Salwinski, Marisa K. Baron, Edward M. Marcotte, and David Eisenberg. DIP: The Database of Interacting Proteins. *Nucleic Acids Research*, 28(1):289–291, 2000.

Yang, Zhihao, Hongfei Lin, Yanpeng Li, Baoyan Liu, and Ye Lu. Trec 2005 genomics track experiments at dutai. In *The 14th Text REtrieval Conference Proceedings (TREC 2005)*, Gaithersburg, MD, USA, 2005.

Yangarber, Roman. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 343–350, Sapporo, Japan, July 2003a.

Yangarber, Roman. *Extraction in the Web Era. Lecture Notes in Computer Science*, volume 2700, chapter Acquisition of Domain Knowledge, pages 1–28. Springer-Verlag Heidelberg, Rome, Italy, 2003b.

Yangarber, Roman, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the sixth conference on Applied Natural Language Processing*, pages 282–289, Seattle, WA, USA, April/May 2000.

Yarowsky, David. Hierarchical decision lists for word sense disambiguation. *Computers and Humanities*, 34(2):179–186, 2000.

Yarowsky, David, Silviu Cucerzan, Radu Florian, Charles Schafer, and Richard Wicentowski. The Johns Hopkins SENSEVAL2 system descriptions. In *Proceedings of SENSEVAL2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 163–166, Toulouse, France, July 2001.

Yeh, Alexander, Lynette Hirschman, and Alexander Morgan. Background and overview for KDD Cup 2002 Task 1: Information extraction from biomedical articles. *ACM SIGKDD Explorations*, 4(2):87–89, 2002.

Yu, Hong and Eugene Agichtein. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19(1):i340–i349, 2003.

Yu, Hong, Vasileios Hatzivassiloglou, Carol Friedman, Andrey Rzhetsky, and W. John Wilbur. Automatic extraction of gene and protein synonyms from Medline and journal articles. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 919–923, San Antonio, TX, November 2002.