Automatically Explaining Literature Based Discoveries

Maryhilda Heidi McClure

Advisor: Dr. Mark Stevenson

Department of Computer Science
The University of Sheffield

This thesis is submitted for the degree of
Doctor of Philosophy

July 2018

I would like to dedicate this thesis to my best friend and husband, Jerry.

Declaration

I hereby declare that the work and words presented in this thesis are my own except where explicitly attributed to others. This work has not been submitted anywhere else for any award.

Maryhilda Heidi McClure
July 2018

Acknowledgements

I would like to thank my advisor, Mark Stevenson, for his patience and careful guidance throughout this PhD process. I also thank Robert Gaizauskas for his valuable inputs during various progress and status meetings about my work. And, for this final version of my thesis, I thank my internal examiner, Lucia Specia, and my external examiner, Eric Atwell for supporting my viva and for sharing your many ideas of how to improve, to share and to expand on my work.

I am grateful to my employer, Polaris Alpha/Intelligent Software Solutions, for supporting my continued education. This journey that led me to pursue a PhD started when I had an opportunity to be involved with the DARPA PAL program (the Personalized Assistant that Learns). That work introduced me to natural language processing and fueled my interest in learning more about the field.

It is difficult to remember all those who helped during this long process of completing my studies – for those not mentioned here, please accept my apologies and my thanks.

Lastly, I would like to thank my family. Thanks to my late parents who always encouraged me to pursue learning in all things that interested me and to my late grandmother who always said that education is a light load to bear. Thanks to my brothers, especially John who read through a few drafts of this thesis and provided valuable inputs. Thanks to my children who, while not mine biologically, have provided me the opportunity to be a mom and, now, Oma to Zoey. And, Thank You to Jerry, my husband, my love and my best friend. Your patience and support during these many years of my PhD research and your support over the many more years we have had together are sometimes unimaginable and are always difficult for me to fathom. I am forever grateful to have you in my life. What's next?

Abstract

Literature based discovery (LBD) identifies potentially related pairs of concepts that are not mentioned together in the same documents. The concept pairs may be identified via linking concepts that are mentioned in both sets of documents or via other statistical relatedness measures like latent semantic indexing. Unfortunately, the nature of the relationships are not identified so the importance and relevancy of the LBD pairs are not known.

The primary objectives of this thesis are to identify candidate LBD related concepts and to determine if the natures of the relationship may be automatically explained using supervised machine learning classification. For example, in the benchmark LBD example of Raynaud's phenomenon (A) being related to fish oil (C), candidate linking concepts are blood viscosity, platelet function and vascular reactivity. The linking concepts are referred to as Bs and, thus, create A-B-C LBD triples. The objectives of this work are to identify a training set of data that includes linking B terms, to identify the relationships between the A and B and the B and C pairs, and to apply supervised machine learning classification techniques to suggest relationship between the A to C concepts. In the Raynaud's example, the suggestion would be that fish oil may *treat* Raynaud's phenomenon.

This work explores data representations suitable for applying classification techniques to explain the relationships. This work applies traditional classification evaluation methods on both classifier outcomes and data designs. Classifiers applied to the training data ultimately accurately predicted the A to C relationships over 70% of the time, while the chosen baselines only achieved approximately 30% accurately predicted relationships. The classifiers were then used on real LBD candidate pairs from an older set of MEDLINE abstracts found using statistical LBD. The predicted LBD explanations were validated against more recent literature which is a time-slice validation approach.

To the best of my knowledge and research, relationship prediction techniques have not been applied to statistically related LBD candidate pairs to provide an explanation of how the A and C pairs are related. Additionally, applying time-slicing for validation of explained LBD candidates is also novel.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Literature based discovery (LBD) is the discovery of hidden knowledge in large sets of documents where the discovery is never explicitly mentioned in any single document. Don R. Swanson was the first to define LBD as a means to discover previously unknown knowledge by examining term occurrences across multiple documents (Swanson, 1986a). After finding documents mentioning a concept of interest, referred to as **A**, Swanson would look for linking terms, referred to as **B**, where A and B are mentioned in the same documents.[1] He would then look for documents that do not mention A, but mention the same B term and would identify a new **C** concept. His hypothesis is that even though never mentioned in the same documents, A and C may be related. Once the related concepts, A and C, are identified, further experiments or investigations are usually performed to determine if there is a real and useful relationship between the LBD pair.

Discoveries that find related A and C concepts using an LBD system may be from an open or closed LBD system. The open approach assumes only a starting concept, A, is known and then all possibly related C concepts are identified and explored. Open LBD systems are like an open-ended search for discoveries. In a closed system, both A and C concepts are identified at the start of exploring their relatedness using, for example, linking B terms. A variation on a closed LBD system is one that limits the vocabulary or set of concepts explored for LBD relatedness. While this work uses this variation of a closed LBD system, the approaches used in this work to explain LBD relationships may be applied to any LBD candidate pairs discovered in either closed or open systems. The focus of this work

[1] To describe the LBD process throughout this thesis, the concepts are referred to as A, B and C where A and C are the LBD discovery, never mentioned together in the same document, and B is a candidate linking term. Multiple linking B terms may exist. The linking B term or terms may show how A is related to B and how B is related to C.

is to find LBD related A and C concepts and then automatically explain the nature of their relationship.

1.1 Motivation for Explaining Discoveries

Simply stating that A and C are related because of the linking B terms is not enough. Understanding the nature of the relationship is also necessary. Computer-based systems are able to quickly and automatically identify literature based discoveries. Unfortunately, they produce incredibly large numbers of hidden knowledge candidates. Many of these prove to be false positives that are not interesting for further study (Henry and McInnes, 2017; Sebastian et al., 2017; Ganiz et al., 2005; Hristovski et al., 2006; Kostoff et al., 2007; Bruza and Weeber, 2008; Seki and Uehara, 2013). Additionally, identifying which of the vast numbers of discoveries are important is daunting and, as noted by Ronald Kostoff et al., is like searching for a needle in a haystack (Kostoff et al., 2007).

In the past, discoveries identified using LBD required additional research to prove or disprove the LBD hypothesis. For example, in Swanson's benchmark discovery that Raynaud's disease may be treated using dietary fish oil, additional clinical trials were later performed to validate the LBD relationship (DiGiacomo et al., 1989). The logic that fish oil may be helpful in curing or treating Raynaud's syndrome simply because these concepts appear in documents mentioning common linking terms is not scientifically sound. Further studies by medical experts are required to understand the relationship between fish oil and Raynaud's disease includes examining the relationships between this LBD candidate pair and the linking B terms. Platelet aggregation is a candidate linking B term for this LBD pair, because it appears in literature mentioning fish oil and in separate literature mentioning Raynaud's disease. As an example of the process required to validate LBD, an expert would determine that blood problems, like platelet aggregation, may be treated with fish oil. They would also determine that Raynaud's disease may be caused by circulation problems such as problems with platelet aggregation. Then, after identifying the natures of the A to B and B to C relationships in the fish oil and Raynaud's syndrome LBD candidate pair, they may conclude that fish oil indeed could treat Raynaud's disease. However, each discovery would require further clinical trials to determine if there is scientific proof that the hypothesis the discovery presents is true, or not.

Computer-aided LBD techniques present an arduously large number of candidate LBD pairs that could contain a few important discoveries. Finding the most interesting and promising candidates is important and useful. The research community would benefit greatly if there was a way to whittle these large lists of possible discoveries down to those that are most interesting and most likely to represent real discoveries. The research presented here seeks to find ways to assist analysts and scientists with their search for interesting candidate discoveries by automatically suggesting explanations for the LBD candidate discovery – the A to C relationships. Explaining LBD candidates will provide a necessary refinement that will help to narrow the candidate discoveries to those that may be the most promising, interesting and useful to explore further.

1.2 Research Questions

The purpose of this research is to automatically explain the nature of the relationships of literature based discoveries. By doing this, researchers may focus on those explanations that are of interest to them and not be bogged down by the mass quantities of discoveries that are irrelevant to their goals. For example, they can focus on the causes of or possible treatments for diseases rather than other less interesting LBD candidate discoveries. The primary question of this thesis is:

> *Can the nature of the LBD relationships be automatically explained?*

Additional questions are:

1. Can this problem be modeled as a classification problem?

   In the medical domain used in this work, is there a representation of the text data that can be modeled and sent to the classifiers in a way that the classifier can predict the nature of the LBD relationship?

2. Is there a source of data for classifier training?

3. Once trained, do classifiers produce measurably better results than baselines?

4. Do classifiers produce results that can be validated against known facts?

   Traditional training and evaluation techniques perform the basis of classification solutions, however, in the field of LBD, additional validation of the produced results is also important.

5. Are ensemble learners better than single classifiers in predicting explanations for LBD relationships?

The ultimate goal in explaining LBD relationships is to be as accurate as possible. This will help researchers focus on the inferred LBD relationships that are most likely to be real and true.

1.3 Objectives

Objectives of this thesis are:

- to design, evaluate and improve classifier solutions for explaining the A to C relationships
- to identify and expand candidate LBD pairs from a corpus, including identification of linking B terms and of A-B and B-C relationships
- to use classifiers and ensemble learners to suggest explanations for the actual LBD relationships
- to validate the explanations of suggested relationships for older LBD pairs using newer facts

1.4 Contributions

These are the contributions that this research provides:

- supervised machine learning classifiers that explain how LBD candidate pairs may be related
- a corpus from SemMedDB that provides machine learning training data and data for validating results
- ensemble learners that improve accuracy in predicting the relationship of LBD pairs
- a framework for using classification to explain LBD discoveries
- a novel validation approach inspired by time-slicing of data

1.5 Thesis Structure

Below is an outline of the rest of this dissertation.

Chapter 2 – Literature Review

This chapter will present current information about LBD and other topics related to this research including techniques used to perform LBD including co-occurrence and statistical methods, background on the data used by the systems in this work (MEDLINE and SemMedDB), LBD evaluation methodologies, limitations of current LBD systems, relationship extraction techniques, an overview of classifiers and of an extension of classification called ensemble learning.

Chapter 3 – Approach and Data for Relationship Prediction

This chapter will begin with the approach used in this work to explain LBD relationships. Then this chapter will present details on the classifier data designs and variations. Training data will be generated and refined in support of the relationship prediction using classification presented in Chapter 4. Refinement of the classifier designs will explore removing instances with the most frequently occurring feature values and regularizing the outcomes.

Chapter 4 – Relationship Prediction Experiments

This chapter will present details on the experiments including evaluation of the classifiers in their ability to explain relationships in the medical domain. Training data described in Chapter 3 will be used in Chapter 4 to apply cross-validation and confusion matrix evaluation of the classifier models. Refined data designs will also be evaluated. Additional evaluation will apply ablation of features to understand the interaction of and dependency between features.

Chapter 5 – Identifying and Explaining Hidden Knowledge

This chapter will present the data used for LBD and for validation of the explained discoveries. It will discuss the results of performing LBD on an older corpus and of LBD relationship explanation using classification on these candidate LBD pairs. The results will be validated against a newer corpus. Additional experiments will apply ensemble learning techniques in an effort to improve accuracy of the predicted results.

Chapter 6 – Conclusions

This chapter will summarize this dissertation. The research questions presented in this chapter will be answered and discussed. This chapter will provide additional discussion of limitations presented in Chapter 2 and will present ideas for future work.

1.6 Published Work

The following papers were published during this research period.
Preiss, J., Stevenson, M., and McClure, M. H. (2012). Towards Semantic Literature Based Discovery *AAAI Fall Symposium Poster Session*.

Chapter 2

Literature Review

This chapter discusses relevant background. The first section provides an overview of literature based discovery (Section 2.1), while the next two discuss how LBD is performed (Section 2.2) and how LBD systems may be evaluated (Section 2.3). The next section presents background information on a source of documents from the medical domain that is used in this research (Section 2.4). The next section presents limitations of current approaches to LBD (Section 2.5). The next section presents how relationship extraction may be done (Section 2.6). The last two sections present information about the various classifiers used in this work (Section 2.7) and information about an extension of classification called ensemble learning (Section 2.8). The last section summarizes this chapter (Section 2.9).

2.1 Literature Based Discovery

As briefly discussed in the introduction, LBD is the discovery of inferred relationships between two concepts that previously were never identified as being related. *Literature* in LBD refers to articles or documents (usually scientific publications). *Discovery* in LBD is the identification of possible relationships between two concepts found in a corpus where the concepts never have been previously mentioned together (Henry and McInnes, 2017; Sebastian et al., 2017; Ganiz et al., 2005; Kostoff et al., 2007; Bruza and Weeber, 2008).

Swanson (1986a,b) is considered to be the first to have mentioned the LBD form of discovery (Smalheiser, 2017; Kostoff et al., 2007; Ganiz et al., 2005; Sehgal et al., 2008). He initially referred to the topic as undiscovered public knowledge since the information that formed the basis of the discoveries was available in publicly available literature and the link between the related concepts had not been explicitly identified or discovered, yet (Swanson, 1986b).

Swanson (1986a) presents one of the early LBD discoveries where Raynaud's disease or phenomenon may be related and possible cured by fish oil. Swanson searched databases available at the time and identified over 2,000 articles on topics relating to Raynaud's disease and over 1,000 on topics relating to fish oil with all articles dated between 1975 and 1984. After further study, he selected two much smaller sets of documents each consisting of 25 documents. The first set mentioned fish oil or related synonyms for fish oil. The second mentioned Raynaud's disease or phenomenon or related synonyms. The sets were also isolated in that the first did not mention Raynaud's and the second did not mention fish oil – in this respect, they were disjoint. These documents were selected because of their extensive discussion about various blood issues like problems with blood viscosity, platelet function and vascular reactivity. The blood issues were the link between the two sets of documents and likewise between the two concepts – fish oil and Raynaud's. Swanson studied these sets of documents and made the connection that, perhaps, Raynaud's disease could be treated with fish oil. Medical studies have since been able to validate this discovery (Bruckner et al., 1987; DiGiacomo et al., 1989).

Throughout his career, Swanson continued to identify novel relationships using LBD techniques including his discovery that there is a relationship between magnesium and migraines (Swanson, 1988). The Raynaud's phenomenon to fish oil relationship and the magnesium to migraines relationship are often used as examples to describe the field of LBD.

In the Kostoff et al. (2007) report, literature related discovery (LRD) is considered to be a super set of literature-based discovery (LBD) and literature assisted discovery (LAD). This is a subtlety of whether or not the discovery simply presented as a possible discovery (i.e., LBD) or if the discovery is further investigated for validity with experts in the topic area (i.e., LAD). Henry and McInnes (2017), Ganiz et al. (2005) and Kostoff et al. (2007) further refined the definition of LBD with the following four basic constraints:

1. *Extraction* of concepts is from published literature (usually scientific articles).

2. *Connections* or relationships of unknown nature are made between pairs of arguments (for example, Raynaud's disease and fish oil or magnesium and migraines.)

3. *Links* are identified between each of the related terms in order to support inference (for example, various blood problems help to form the inferred relationship between fish oil and Raynaud's).

4. *Novelty* of the discovery – no current literature mentions the new related pairs in the same publication thus making this a candidate discovery.

LBD work is different from other basic search and concept linking techniques because fundamental LBD requires that documents mentioning two candidate LBD related concepts must contain some form of linking concept (Kostoff, 2008). That is, for example, magnesium and migraines are never mentioned together prior to the LBD discovery by Swanson (1988). LBD generates the hypothesis that the two may be related because of various concepts linking them together: type A personality, vasospastic disorder, vascular tone and reactivity, calcium channel blockers, etc. Once a discovery is made, further validating studies must be performed. For example, clinical trials that show a drug helps to cure or reduce symptoms of a disease.

2.1.1 The ABCs of Swanson's Work

Swanson (1986b,a, 1988) introduces the use of A, B and C in describing LBD related concepts and their linking terms. He suggests as an example that A stand for Raynaud's disease or phenomenon and that A is found in a set of documents not containing C. He suggests that C stand for Fish Oil and that C is found in a separate set of documents not containing mentions of A. He suggests that B stand for the various linking terms found in both corpora. LBD continues to be explained by most researchers using this A, B and C notation where A and C are the LBD candidate discoveries and B is the linking term.

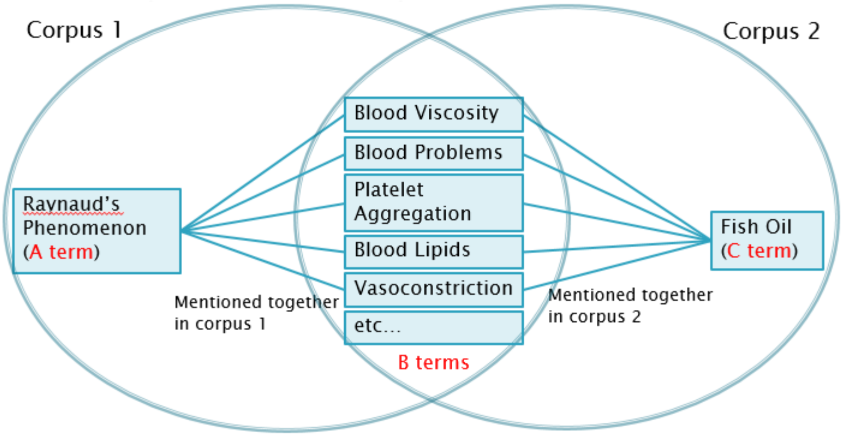


Fig. 2.1 Classic diagram of Literature-Based Discovery using Swanson's example of Raynaud's disease relating to fish oil

There may be and usually are more than one linking B term. Figure 2.1 shows a visualization of two corpora – one containing A on the left, another containing C on the right and the intersection are documents that contain linking B terms. The linking B terms may

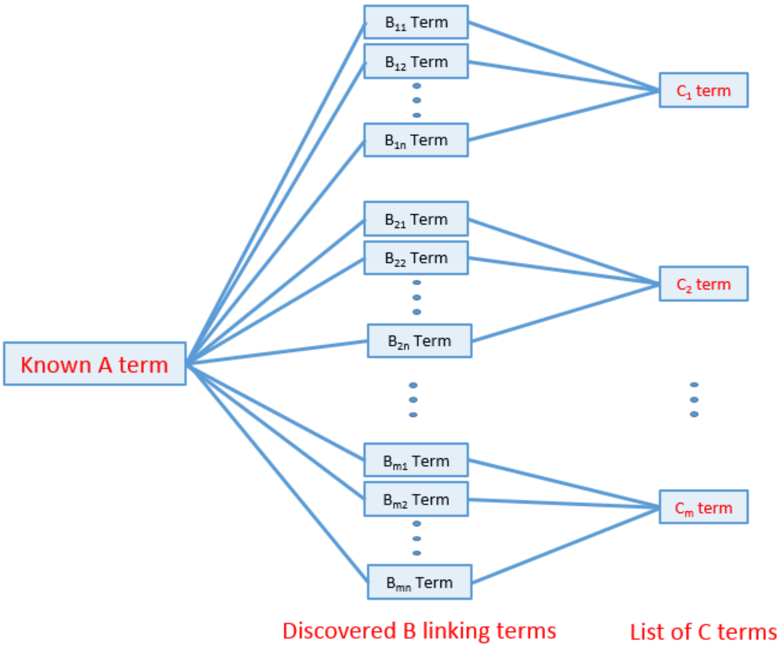


Fig. 2.2 Open LBD System: Uses a starting A term and a set of candidate C terms, C_1 to C_m, and discovers sets of LBD candidate ABC triples. Each blue line represents some relationship between the two terms. This and the next diagram are adapted from various found in LBD literatures including Weeber et al. (2001)

not appear in all of the documents in the intersection and the B linking terms may also appear in documents that mention neither A nor C. This subtlety is not shown in the diagram.

2.1.2 Open and Closed LBD

There are two fundamental types of LBD: open and closed. Open discovery LBD systems begin with a starting term or concept (referred to as A) and then search for intermediate terms or concepts (connecting B terms or concepts) which will lead to target terms or concepts (C). See Figure 2.2. If not previously mentioned together, the LBD system concludes that the opening term, A, and the discovered target term, C, may be related. In a closed system, both the starting (A) and the target (C) terms are known and a linking term or concept (B) is the goal of the discovery. See Figure 2.3. Henry and McInnes (2017) and Ganiz et al. (2005) describe this in more detail.

Examples of open systems or methodologies include BITOLA (Hristovski et al., 2006) and LSI (Gordon and Dumais, 1998; Gordon et al., 2002). Arrowsmith is an example of a closed system (Swanson and Smalheiser, 1997). When researchers try to validate Swanson's

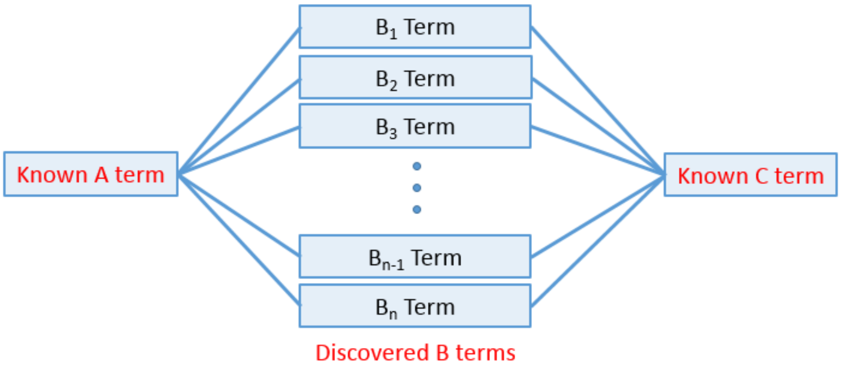


Fig. 2.3 Closed LBD System: Uses a starting A and C term and discovers candidate B terms. As in diagram above, the blue lines between two terms represents a relationship.

fish oil to Raynaud's disease discovery in an LBD system, this is an example of a closed system – both the A and C concepts are known and the goal is to search for logical linking B terms. The work presented in Gordon and Lindsay (1996) is an example of using a known A and C pair (closed LBD). Additionally, Gordon et al. (2002) discuss starting with the middle term – the "inspiration", and by reversing the approach to be C-B-A, they come up with an "extension".

Swanson and Smalheiser (1997) also explored LBD relationships where both concepts have been identified as being related, but where the nature of the relationship is not yet known. For example, the suggested relationships between indomethacin and Alzheimer's disease, between estrogen and Alzheimer's disease and between phospholipases and sleep. These three pairs of concepts represent examples of closed LBD studies because because a relationship had been identified between the concepts, but the linking cause was not understood.

LBD researchers sometimes use lists of words (vocabularies) to restrict which words they considered in each search – when lists are used for both A and C concepts, the LBD system is called a closed system. An open system considers a list of words only for the starting concept, A (Henry and McInnes, 2017; Ganiz et al., 2005). Weeber et al. (2001) consider an open system one that generates hypotheses and a closed system is one that tests hypotheses.

2.1.3 Additional Terminology

This section presents terminology used throughout this thesis. While common in the field of LBD to use A-B-C notation, using classification to explain the LBD A-C relationships requires a few additional expressions (for example, fully and partially qualified triples) and the experimentation presented in this thesis uses two slightly different forms of judging the value of the results (evaluation and validation).

Figures 2.4 and 2.5 provide descriptions using the ABC notation introduced in Section 2.1.1 and they show previously known (solid lines) and candidate LBD relationships (dashed lines). These diagrams are used in some of the definitions that follow.

Candidate LBD pair

Whether LBD is performed manually or using automated techniques, a candidate LBD pair is a a pair of concepts that are found to be related but are never mentioned in the same document. These are the A-C instances shown with dashed line in Figures 2.4 and 2.5.

Linking B Term

Automatically explaining the relationship between an LBD pair requires that one or more linking B term is identified. The linking B terms are found in a set of documents that contain the A concept and another disjoint set of documents that contain the C concept. The concepts in the intersection of the diagram in Figure 2.1 are examples of linking B terms. In Figures 2.4 and 2.5, some A and C candidate LBD pairs are linked together by common mentions of some linking B terms.

Triples and Relationships

Various full and partial triples are shown in Figures 2.4 and 2.5. Basic representations of the ABCs are shown in the first diagram (Figure 2.4) – the solid blue lines represent instances where the concepts are mentioned in the same document; the dashed red lines represent concept pairs not mentioned together but that are considered to be related based on statistical co-occurrence or other LBD methods. Triple A_2-B_1-C_1 is an example of a full triple that has no LBD pair – that is, A_2 is mentioned with B_1 and with C_1 (not necessarily the same documents) and C_1 is also mentioned in documents with B_1. This A_2-B_1-C_1 triple, as shown in Figure 2.4, is not fully qualified, because the nature of the relationships are not shown. Triple A_4-B_2-C_3 is an example of a partial triple where B_2 is a linking term between candidate

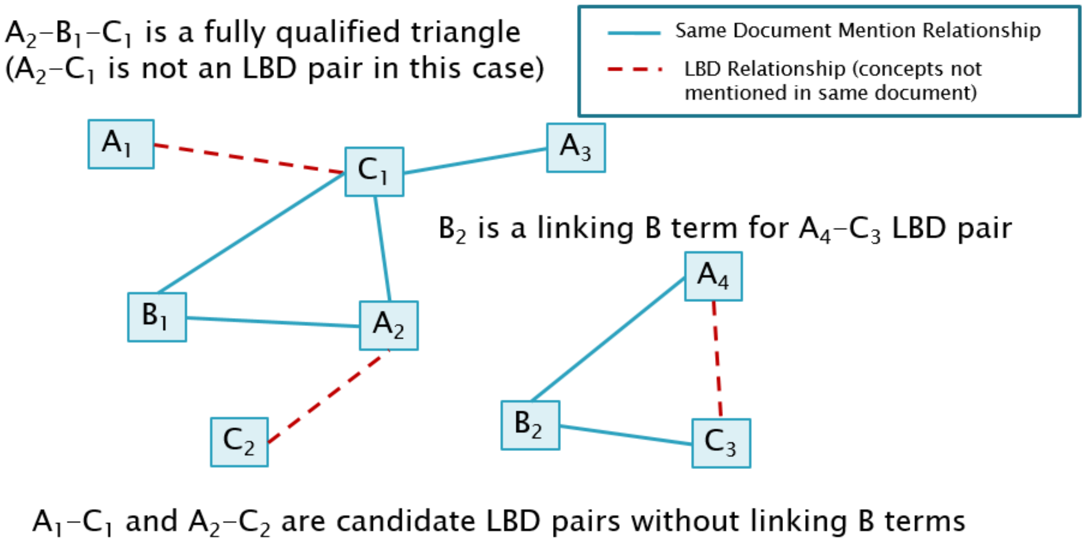

A_2–B_1–C_1 is a fully qualified triangle
(A_2–C_1 is not an LBD pair in this case)
Same Document Mention Relationship
LBD Relationship (concepts not mentioned in same document)
A_1
C_1
A_3
B_2 is a linking B term for A_4–C_3 LBD pair
A_4
B_1
A_2
B_2
C_3
C_2
A_1–C_1 and A_2–C_2 are candidate LBD pairs without linking B terms

Fig. 2.4 What is and is not LBD

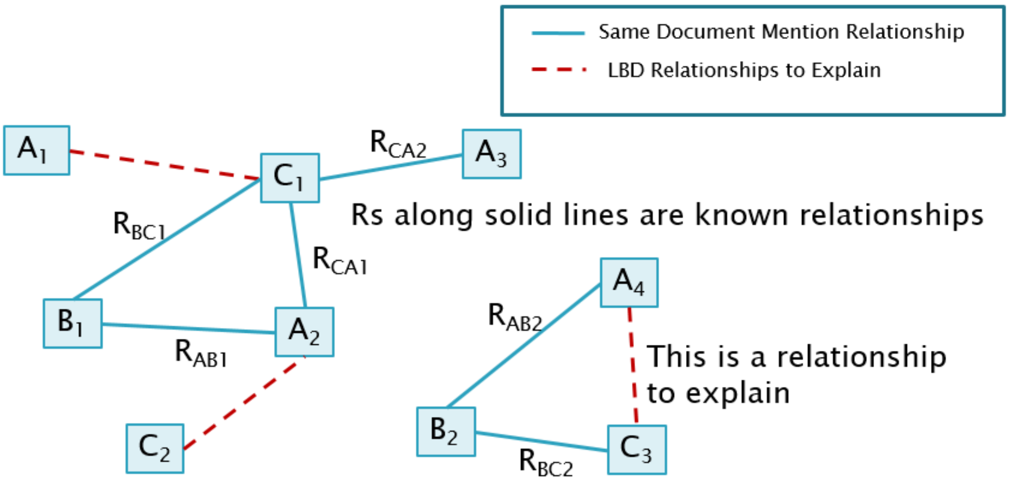

Same Document Mention Relationship
LBD Relationships to Explain
A_1
R_{CA2}
A_3
C_1
Rs along solid lines are known relationships
R_{BC1}
R_{CA1}
A_4
R_{AB2}
B_1
A_2
This is a relationship to explain
R_{AB1}
B_2
C_3
C_2
R_{BC2}

Fig. 2.5 Relationships LBD Triples

LBD pair – A_4 and C_3 (this is not a complete partial triple because the nature of the A-B and B-C relationships are not shown). The other two red dashed lines show instances of candidate LBD pairs where there is no linking B term.

Figure 2.5 shows full and candidate partial triples, but also adds relationship explanations to the solid blue lines. This is the information needed to create fully qualified triples and complete partial triples. A_2-B_1-C_1 along with R_{AB1}, R_{BC1} and R_{CA1} is a fully qualified triple. Each relationship includes not only the nature of the relation, e.g. "causes", but also the direction of that relationship, e.g. A causes B or B causes A. Triple A_4-B_2-C_3 along with R_{AB2} and R_{BC2} is a complete partial triple for a candidate LBD pair with a linking B term. It is a complete partial triple because R_{AB2} and R_{BC2} are defined.

Fully Qualified Triples Fully qualified triples contain A, B and C terms along with A-B, B-C and C-A relationships and the direction of each relationship. As just noted, in Figure 2.5, A_2-B_1-C_1 along with R_{AB1}, R_{BC1} and R_{CA1} represents a fully qualified triple.

Partially Qualified Triples Partial triples contain everything in a fully qualified triple except the C-A relationship and direction. LBD candidate pairs where B linking terms are identified and where the nature of the A-B and B-C relationships are also identified are complete partial triples. In Figure 2.5, A_4-B_2-C_3 along with R_{AB2}, R_{BC2} and an unknown A to C relationship form a partially qualified triple.

Evaluation versus Validation

As with any work that applies classification techniques to solve problems, evaluation may be performed using N-fold cross-validation, studies of confusion matrices and hypothesis testing. This work uses an additional form of evaluation that will be referred to as *validation* in an effort to separate traditional classifier training evaluation from the evaluation of the classifier models applied to the task of explaining the nature of the LBD relationships. Validation presents information on the performance of the classifiers when used on sets of LBD produced data using time-slicing (described in detail in Section 2.3). Briefly, validation takes the results of classifiers trained and used on older data (e.g. data from 1980-1984) and tries to see if the explained discovery is able to be verified in the newer data (e.g. data after 1984).

2.2 Performing LBD

This section first provides a short overview to information retrieval (Section 2.2.1), which is used by most LBD approaches. Then this section discusses how LBD may be performed in automated systems using simple co-occurrence (Section 2.2.2), and using statistical co-occurrence methods (Section 2.2.3).

2.2.1 Information Retrieval

In order to perform LBD, documents must be identified that contain the terms or concepts of interest. Information retrieval (IR) is the process of returning structured or unstructured data that match a specified search criteria (Manning et al., 2008). The data are usually documents including web pages and the search criteria are usually keywords or phrases. Manning et al. (2008) also state that IR is usually performed on a corpus using some form of indexing of the documents to allow for faster and efficient retrieval of them. Indexes may be simple or may be augmented in various ways to enhance speed or focus on specific functionality or outcomes.

Inverted indexes are a basic building block of IR. The simplest form of an inverted index keeps track of each word found in each document. When stored, each word will have a link back to the document or documents from where it came. The words that are stored may be normalized so that, for example, an occurrence of "stores" will be indexed as "store". (Jurafsky and Martin, 2009; Manning et al., 2008).

More complex inverted indexes take into consideration the number of times the word appears in the document and consider the proximity to words when multi-word searches like "apple pie" are performed. When considering these types of improvements, weighting schemes are introduced. An example of this is the term frequency, inverse document frequency (or tf-idf) calculation. The inverse document frequency weight for some term t is computed as follows:

$$idf_t = log\left(\frac{N}{df_t}\right) \tag{2.1}$$

where N is the number of total documents and df_t is the number of documents in which the term t appears in the corpus (Jurafsky and Martin, 2009; Manning et al., 2008). The importance of an idf weighting is that terms that are common across the corpus will have a smaller idf and those that show up less often will have a larger idf. Then when the number of times the term shows up (i.e., the term frequency of term t or the tf_t) is multiplied by the idf_t, less common terms will have higher tf-idf weighted score than common terms. For example, in a corpus that contains documents about businesses and the products they sell, the word "the" will have a smaller tf-idf score than the word "automobile".

Systems based on inverted indexes usually perform additional analysis of the query phrase and of the indexed documents. Query phrases will be normalized and synonyms considered so that the query will return the most accurate and useful set of results. The documents that are indexed will also be normalized and will include information about candidate synonyms so that the index is optimized to provide good search results (Jurafsky and Martin, 2009; Manning et al., 2008).

Vector space models are an important source of algorithms for IR. A vector space model is a vector representation of term information where weighted scores for each term is effectively plotted and compared against other terms of interest. Each term may have a weighted score (like tf-idf discussed above) included in vector model. Using the vector math, a score showing the relatedness of the terms is calculated. In vector space models, cosine similarity is often used to identify similar terms (Salton et al., 1975; Manning et al., 2008).

2.2.2 Co-occurrence

Since his initial study, Swanson and many other researchers have applied automation to the task of LBD (Hristovski et al., 2006; Gordon and Dumais, 1998; Swanson and Smalheiser, 1997; Culotta et al., 2006; Downey et al., 2005). Initially systems would look for candidate B linking terms simply by their co-occurrence in the documents that mention the A term then use these B terms to search for a new set of documents where A is not mentioned. Then they would try to find, again with co-occurrence, candidate C terms linked to B in the new set of documents. The automation of these approaches usually entailed database queries to find linked terms (Swanson and Smalheiser, 1996; Kostoff et al., 2007).

Wren (2008a) suggests that most LBD systems use co-occurrence and that systems need to be improved from here. Although co-occurrence generates interesting results, there is no guarantee that co-occurrence implies relationship. They may not be related. Kostoff et al. (2008a, 2007) stress that just having concepts that co-occur is not enough to conclude that they are related in any meaningful way especially since with LBD-related concepts, there are often few real discoveries. Additionally, simply finding relationships using co-occurrence with single words is not enough. Co-occurrence of multi-word phrases that consist of bigrams and trigrams.[1] are also important in the identification of LBD concept pairs (Lindsay and Gordon, 1999).

An example of how co-occurrence may not mean relatedness is negation (Wren, 2008a). The sentence "watching the news does not cause measles" contains "news" and "measles", but the relationship is a "not cause" relationship. The sentence "there is no convincing evidence that watching the news causes chickenpox" contains "news" and "chickenpox", but further analysis of the sentence is required to see that there is no evidence of the causes relationship (Preiss et al., 2012).

2.2.3 Statistical Methods for LBD

Statistical techniques for performing LBD are technically co-occurrence approaches (Manning and Schütze, 1999). This is because concepts are found to be related using statistics on word occurrences. An interesting result of statistical relatedness is that concepts may or may not reside in the same document. If not, the statistical approach has identified a candidate LBD pair. However, even though the concepts are not in the same document, their co-occurrence with linking terms ultimately make this a co-occurrence LBD methodology (Lindsay and Gordon, 1999).

In statistical approaches to LBD, there may not be a linking term identified when the making the discovery. Instead, related concepts are discovered by the semantic relatedness of the documents, first, and then linking B terms are identified (Gordon and Dumais, 1998). The documents are treated as bags of words and the documents are found to be statistically related based on the occurrences of the same or similar words in the same or similar documents. Again, the distinction between LBD discoveries and non-LBD discoveries in statistically based systems is simply whether or not the related concepts appear together. If the terms do not appear together, a candidate LBD pair has been identified. If the terms appear together, then these terms may end up providing candidate B linking terms.

The rest of this section discusses two general approaches to identify related concepts in documents using statistics methods – latent semantic analysis and random indexing. Both are improvements over simple co-occurrence methods because they reduce the complexity of the m x n storage matrix that relates the m terms to the n documents. Methods that require the full matrix representation, such as the co-occurrence approaches above, do not scale and thus are not suited to large corpora and associated information retrieval tasks including LBD (Deerwester et al., 1990; Cohen et al., 2010; Widdows and Cohen, 2010).

[1] bigrams and trigrams are 2 and 3 word phrases, respectively, and are also referred to as n-grams of length 2 or 3, respectively.

Latent Semantic Analysis and Latent Semantic Indexing

Latent semantic indexing (LSI) is described in Gordon and Dumais (1998). LSI looks not only at index results for words or phrases; it also takes into account the number of times the term or phrase shows up in other documents in deciding if this term or phrase is relevant (i.e., important). LSI is also able to relate terms or phrases based on how many times they appear together in documents.

LSI is an information retrieval technique that uses singular value decomposition (SVD) to reduce the dimensions of extremely large matrices by getting rid of less interesting data. SVD breaks down large problems using dimension reduction. Kalman (1996) gives credit to Strang (1980) for introducing the concepts of SVD and both have explanations of the theory as does Manning et al. (2008). The full mathematics behind SVD will not be presented here, instead a high level conceptual description and overview of how SVD applies to information retrieval is presented.

When applied to information retrieval, SVD involves breaking a complete term by document matrix into the product of three simpler to manipulate matrices. The number of terms may be called *m* and the number of documents, *n*, which would result in the complete *m* x *n* matrix with non-zero or weighted entries in each location where a term (t_i) is related to the document (d_j). Here i is the row and j is the column in the full matrix. The value stored in the (i, j) location in the matrix would usually be a weighted representation of the the relationship between that term (t_i) and the document (d_j) – for example, it could be the term frequency, inverse document frequency weight that is discussed in Section 2.2.1.

The SVD is generally represented as

$$A = U\Sigma V^T \tag{2.2}$$

where the resulting matrix, A, is a low rank approximation of the complete term by document matrix. The matrices, U and V^T are orthogonal. The Σ in this equation is a diagonal matrix which means it has non-zero values along a diagonal and zeros elsewhere. If it is a square matrix where *m* equals *n*, the diagonal goes from the (1, 1) location down the diagonal to the (*m*, *m*) location. When SVD is applied to term by document matrices, *m* is usually not equal to *n* so the Σ matrix is usually not square. In this rectangular case, matrix A is *m* x *n*, U is *m* x *m* and V^T is *n* x *n*. Matrix Σ would then be *m* x *n* but would only have non-zero entries in the diagonal starting at (0, 0) and ending at (*n*, *n*) if $n < m$ and at (*m*, *m*) if $m < n$ (Manning et al., 2008).

The importance of the Σ matrix is that because it is mostly zeros, the computational complexity of the product of the three matrices is greatly reduced, remembering that zero multiplied by anything is zero. This allows much larger data sets of terms and documents to be indexed using SVD techniques including LSI.

As noted earlier, much more thorough presentations of SVD and the supporting linear algebra mathematics behind SVD are found in many publications including, Manning and Schütze (1999), Kalman (1996), Manning et al. (2008), Strang (1980) and Albright (2004).

Related to LSI, latent semantic analysis (LSA) is the process of using the statistical approaches to create indexes which will allow the retrieval of similar concepts (Deerwester et al., 1990). LSA does not require, necessarily, a vocabulary, but, instead, finds similar documents based on latent semantic indexing (LSI) or enhancements to LSI like Random Indexing (Cohen et al., 2010) which is discussed in following sub-section. The LSA concept assumes that if terms or concepts are found in similar sets of text (not always the same text, but similar) then these terms or concepts may be related concepts, may represent similar concepts, or may be the same concept. Although slightly different in meaning, LSA and LSI are used interchangeably in this thesis. LSA is the concept, LSI is a mathematical implementation.

Random Indexing and Reflective Random Indexing

LSI proved to be more efficient than previous methods for information retrieval involving terms and documents and has been moderately successful. However, it is still slow and has computationally complexity of $O(mn^2 + m^2n + n^3)$ for m rows of terms and n columns of documents in the matrix (Widdows and Ferraro, 2008). More recently, Cohen et al. (2010) have experimented with random indexing (RI) – a more scalable version of LSI – and extended the RI concepts to support indirect inference. Indirect inferences are what Cohen, et al., sometimes call LBD. RI uses a random approach to further reduce the size of matrices being analyzed to discover similar terms in documents. Instead of a full term by document matrix, documents are placed into small sets of columns. For example, if there are 10,000 documents, a document may be assigned to one of 20 randomly chosen columns. Each document's term frequency information is tallied in each of its columns along with any other document that was randomly assigned (Kanerva, 2009).

Cohen et al. (2010) also experimented with variations of RI – Sliding windows on RI, Term based reflective random indexing (RRI) to find related terms, and document based RRI to find related documents. RRI uses multiple passes of RI where the results of one pass are fed into the next. Term and document based RRI vary how the random indexing is chosen – by term or by document in various passes through the RRI. Their claim is that these techniques provide more related terms/concepts that may not co-occur in the same document but are possibly related (i.e. LBD candidates). They state that their use of RRI techniques is better suited for LBD than other LSA techniques primarily because the size of the matrices are reduced by orders of 10^4 or more (Widdows and Cohen, 2010).

Widdows and Cohen (2010) along with Widdows and Ferraro (2008) have developed a package called Semantic Vectors for performing LSI using RI and RRI. Semantic Vectors expands on LSI appoaches and introduces random indexing and reflective random indexing. Both are improvements to LSI and the computationally expensive problems of LSA.

2.3 Evaluation Methodologies for LBD Systems

Ultimately, LBD systems try to discover previously unknown knowledge. For example, an LBD system may discover a drug that may cure a disease but in the corpus examined, the drug and disease were never mentioned together. If the corpus includes the most recent literature, then it may be impossible to determine if the discovery is valid without performing additional time consuming tests. When studying MEDLINE or other medical domain literature, performing medical trials is the best way to prove that a discovery is valid or not (Kostoff, 2008). However, there is still a need to evaluate the performance of LBD systems (Yetisgen-Yildiz and Pratt, 2009). Most current LBD systems are evaluated in one of four general ways (Henry and McInnes, 2017; Yetisgen-Yildiz and Pratt, 2009, 2006; Bruza and Weeber, 2008; Ganiz et al., 2005):

1. Replicating previous discoveries: For example, using Swanson's discoveries as the gold standards and proving that the system is able to find the same discoveries using the same older set of documents.

2. Use of medical experts: When a drug is studied of which a medical doctor or researcher has much knowledge, their expertise can help to validate or discount a discovery. Henry and McInnes (2017) refers to this as taking a new proposal applying empirical evaluation.

3. Time slicing the corpus: Here the LBD is performed on a corpus from an older time range and then is validated against newer documents. This is similar to item 1 in that the LBD system would perform LBD on the same corpus used for Swanson's discoveries. It is different because the discovery need not be previously identified like

Raynaud's and fish oil. Once the system identifies a candidate discovery, the system searches newer documents to see if the discovery is mentioned there.

4. Publishing results in medical journals: Put discoveries in front of medical experts and let them validate or discount the discoveries. Henry and McInnes (2017) include evaluation of how users interact and engage with systems thus providing real-world usefulness of discoveries. This is a variation on publishing in medical journals.

Ganiz et al. (2005) indicate that there has not been much research in the area of evaluating LBD results. Evaluation using good scientific methodologies is lacking in many prior works related to LBD – often researchers simply try to recreate the Swanson results (fish oil to Raynaud's). Ganiz, et al., point out that evaluation of LBD is difficult when one is trying to discover previously undiscovered knowledge. In early LBD works, clinical trials were used or studied to validate the LBD discovery. This was how Swanson's fish oil helping to cure Raynaud's phenomenon was proven. More recently, Sebastian et al. (2017) also report that evaluation of LBD results is difficult and that reproducing previous discoveries may cause overfitting of solutions. They propose that more work needs to be done to develop consensus in LBD evaluation metrics.

Yetisgen-Yildiz and Pratt (2009) proposed a way to evaluate LBD systems. In their work they propose using a before and after model. They propose that if LBD is run against data from a specific date and earlier, discoveries may be found which were never mentioned in the same set of data. Then, to validate if the discovery is a real one, examine data after the specific date and see if the discovery is mentioned in more recent documents. If the discovery is not mentioned prior to the specific date but is mentioned in more recent documents, then LBD successfully discovered something not previously known. In this thesis, this technique is called time slicing.

Time slicing provides interesting advantages over other methods. It does not require experts to validate or discount the discovery whether by immediate consultation (item 2, above) or by publishing and waiting (item 4). Instead it uses newer publications which provide the expertise to validate or discount discoveries.

2.4 Medical Domain Data

Although LBD was first applied to medical documents and abstracts, the format of the text data analyzed is not limited to documents nor to the medical domain. An example of a non-medical LBD study is the work applying LBD to water purification (Kostoff et al., 2008b) and to the intelligence domain (Bradford, 2006). The data may reside in databases, blogs or other sentence or multi-sentence corpora. The basic concept behind LBD depends on multiple mentions of A, B and C concepts – or at least enough mentions to make the connections with more than just one-mention links. The medical domain is attractive for research in the text analysis fields because it is freely available for research and provides a large quantity of documents.

This section describes two datasets used in this research that are focused from the medical domain – one is the MEDLINE corpus (MEDLINE, 2002), against which LBD is performed (Section 2.4.1), and the other is the Semantic MEDLINE Database or SemMedDB (Kilicoglu et al., 2012), which provides data for training, relationship explanation and result validation (Section 2.4.2).

2.4.1 Medical Corpus and Concept Names

Citations and abstracts of medical journals are available from the US National Library of Medicine (NLM) National Institutes of Health (NIH) and their systems called PubMed and the Unified Medical Language System or UMLS (Bodenreider, 2004).[2] In particular, the MEDLINE corpus provides access to over 24 million abstracts and the Medical Subject Headings (MeSH) provides vocabulary of concepts including synonyms and other linking of medical concepts (MEDLINE, 2002).[3] The actual repository of abstracts and associated lists of concepts including chemical terms and MeSH may be found in the MEDLINE/PubMed Baseline Repository.[4]

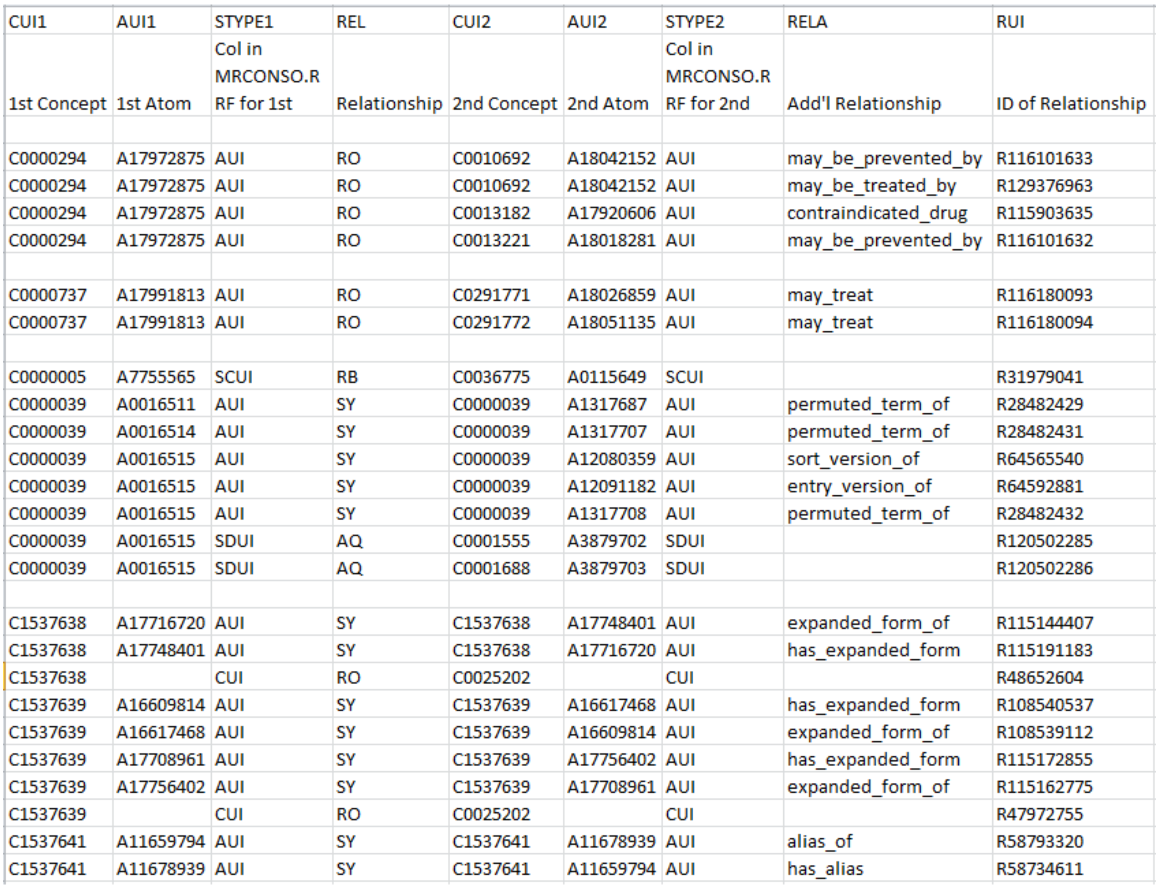
Biomedical text mining or BioNLP has been a large place of research and often includes studies involving the MEDLINE corpus. BioNLP touches on a combination of information extraction including natural language processing (NLP), information extraction and bioinformatics (Cohen and Hersh, 2005).

Building on the MEDLINE corpus, Rindflesch and Fiszman (2003) have developed a system called SemRep that extracts relationship information for concepts identified in the Unified Medical Language System (UMLS). They used techniques that interpret MEDLINE abstracts identifying relationships but generalizing the relationships into a small set of types of relationships. For example, TREATS and CAUSES are the general relationships but

[2] https://uts.nlm.nih.gov/home.html

[3] http://www.ncbi.nlm.nih.gov/pubmed/ or http://www.PubMed.gov/

[4] https://mbr.nlm.nih.gov/



CUI1	AUI1	STYPE1	REL	CUI2	AUI2	STYPE2	RELA	RUI
1st Concept	1st Atom	Col in MRCONSO.RRF for 1st	Relationship	2nd Concept	2nd Atom	Col in MRCONSO.RRF for 2nd	Add'l Relationship	ID of Relationship
C0000294	A17972875	AUI	RO	C0010692	A18042152	AUI	may_be_prevented_by	R116101633
C0000294	A17972875	AUI	RO	C0010692	A18042152	AUI	may_be_treated_by	R129376963
C0000294	A17972875	AUI	RO	C0013182	A17920606	AUI	contraindicated_drug	R115903635
C0000294	A17972875	AUI	RO	C0013221	A18018281	AUI	may_be_prevented_by	R116101632
C0000737	A17991813	AUI	RO	C0291771	A18026859	AUI	may_treat	R116180093
C0000737	A17991813	AUI	RO	C0291772	A18051135	AUI	may_treat	R116180094
C0000005	A7755565	SCUI	RB	C0036775	A0115649	SCUI		R31979041
C0000039	A0016511	AUI	SY	C0000039	A1317687	AUI	permuted_term_of	R28482429
C0000039	A0016514	AUI	SY	C0000039	A1317707	AUI	permuted_term_of	R28482431
C0000039	A0016515	AUI	SY	C0000039	A12080359	AUI	sort_version_of	R64565540
C0000039	A0016515	AUI	SY	C0000039	A12091182	AUI	entry_version_of	R64592881
C0000039	A0016515	AUI	SY	C0000039	A1317708	AUI	permuted_term_of	R28482432
C0000039	A0016515	SDUI	AQ	C0001555	A3879702	SDUI		R120502285
C0000039	A0016515	SDUI	AQ	C0001688	A3879703	SDUI		R120502286
C1537638	A17716720	AUI	SY	C1537638	A17748401	AUI	expanded_form_of	R115144407
C1537638	A17748401	AUI	SY	C1537638	A17716720	AUI	has_expanded_form	R115191183
C1537638		CUI	RO	C0025202		CUI		R48652604
C1537639	A16609814	AUI	SY	C1537639	A16617468	AUI	has_expanded_form	R108540537
C1537639	A16617468	AUI	SY	C1537639	A16609814	AUI	expanded_form_of	R108539112
C1537639	A17708961	AUI	SY	C1537639	A17756402	AUI	has_expanded_form	R115172855
C1537639	A17756402	AUI	SY	C1537639	A17708961	AUI	expanded_form_of	R115162775
C1537639		CUI	RO	C0025202		CUI		R47972755
C1537641	A11659794	AUI	SY	C1537641	A11678939	AUI	alias_of	R58793320
C1537641	A11678939	AUI	SY	C1537641	A11659794	AUI	has_alias	R58734611

Table 2.1 UMLS MRREL.RRF Sample Data with column headers added for readability

may be derived from more specific and technical phrases. They use the phrase *hypernymic propositions* to describe this simplification of relationships types.[5]

The UMLS Metathesaurus is a source of medically related terms and relationships between those terms (UMLS, 2012). UMLS provides a browsing system called MetamorphoSys which allows some navigation of concepts and terms related to those concepts. The actual underlying files like MRREL.RRF provide even more information about relationships than is exposed by MetamorphSys. Some examples of the information are in Table 2.1. The CUI1 and CUI2 columns are keys that map to concepts in the MRCONSO.RRF file. There are also columns in this table that map concepts to other concepts that are basically the same thing – for example, Raynaud's disease (possessive) and Raynaud disease (not possessive).

[5] Hypernyms are words that are the general class of another set of words. For example, color is a hypernym of red.

MetaMap is another system available with the UMLS which can identify concepts in texts including handling the synonyms and word sense disambiguation. MetaMap provides a Java API (MetaMapAPI) to assist in programmatically identifying the CUI associated with a concept including confidence scores – higher scores mean closer match.

2.4.2 Medical Relationships Data

In addition to raw UMLS data, data derived from the Metathesaurus and SemRep, the Semantic MEDLINE Database (SemMedDB) is also available for use from NLM. SemMedDB is a MySQL database that contains information derived from the full MEDLINE and UMLS data. The content of SemMedDB is information on the concepts and relationships to other concepts along with the supporting sentence containing the relationship. Information is also available that links back to the original MEDLINE citation (Rindflesch et al., 2011). SemMedDB contains subject-predicate-object triple information that allows easy display and processing of the data as a graph. There are currently over 58 million subject-predicate-object triples in the MySQL SemMedDB. A more recent application of the SemRep and the data found in SemMedDB is that which is described in Hristovski et al. (2015), where they present a question answering system built off the semantic relationships between concepts stored in the SemMedDB.

The steps used to create SemMedDB are the following:

1. Issue a query to pubmed to retrieve MEDLINE citations
2. Use results from SemRep to provide relationships
3. Normalize the relationships so that they match relationships defined in UMLS
4. Remove sets that are too general like those referring to “individual” or “person”
5. Store data in SemMedDB
6. Present data in graphs (link charts)

There are eight tables in the MySQL SemMedDB.[6] The predication_aggregate table contains relationship triples including the subject, predicate, object and references to corresponding sentences. Sentences are in another table and that table contains information linking the sentence back to the actual PubMed MEDLINE document.

[6] https://skr3.nlm.nih.gov/SemMedDB/dbinfo.html

2.5 Limitations of Current LBD Systems

This section discusses some specific limitations in current LBD systems. First, the nature of the discovered relationship is left up to the researcher to determine after the candidate LBD discovery has been made (Section 2.5.1). Second, they include lack of accuracy in LBD discoveries and the fact that there can be very large numbers of candidate LBD pairs discovered using co-occurrence approaches (Section 2.5.2). And last, using statistical approaches like LSA do not identify candidate linking B terms (Section 2.5.3).

2.5.1 Explanation of LBD Candidate Relationships

Current LBD systems do not attempt to automatically identify why A and C are related, even when candidate B terms are identified. Even when A and B or B and C are found to be in close proximity of each other, no system tries to automatically identify what the relationship is between A and B or between B and C. Likewise, the inferred relationships between A and C are not automatically generated, either. That is, given a relationship between A and B and a relationship between B and C, no system automatically suggests what the relationship might be between A and C – even when not explicitly mentioned anywhere in the corpus. For example, as noted before, if A is Raynaud's disease, B is blood viscosity problems and C is fish oil. If A and B co-occur with proof in the documents that A *is caused by* B, and B and C co-occur with proof documenting that C *treats* B, but A and C are never mentioned together, a conclusion may be that Raynaud's may be *treated* by fish oil.

2.5.2 Lack of Accuracy and Excess Quantity Using Co-Occurrence

Some techniques used in LBD depend solely on co-occurrence of terms to decide that they may be linked. Co-occurrence may not restrict where and how linking B terms appear in documents. For example, in the Raynaud's disease. Making conclusions based simply on co-occurrence is not possible as it leaves out the reason or explanation of the relationships. This leaves much more work for the researcher in validating the mass quantities of hypotheses discovered in LBD (Preiss et al., 2015; Hristovski et al., 2006; Wren, 2008a; Ganiz et al., 2005). Henry and McInnes (2017) and Kostoff (2008) discuss the difficulty in validating as being accurate the vast quantities of discoveries that LBD systems generate as being common problems preventing wider adoption of LBD systems.

There have been improvements to co-occurrence approaches to LBD like limiting the vocabulary for the candidate A and C terms (a variation on closed LBD) and also in having a limited set of B terms rather than considering all of them. By using a vocabulary of terms related to the subject matter being studied, B terms may be identified that are more likely to have a relationship with the LBD A and C concepts (Hristovski et al., 2008). In the medical domain which will be described more in Section 2.4, a common vocabulary of medical terms is available in MEDLINE (2002) and is called the Medical Subject Headings (MeSH). However, even with something like the MeSH vocabulary, more post processing work is required to determine if the B terms makes sense and really point to a discovery of a relationship between A to C. That is, depending on co-occurrence to find the linking B terms may introduce too many false positives or noise. Other improvements to co-occurrence involve examining the general proximity of the A, B and C terms. If A, B and C are relatively closely located in the documents, they are given higher probability of being related. Again, however, this does not explain the relationships.

2.5.3 Using LSA Does Not Identify Candidate B Terms

LSA, including RI and RRI, uses statistical mathematics and singular value decomposition on term by term or term by document matrices to identify terms that may be related (See Section 2.2.3). When found to be related, the terms may or may not appear together in the same documents. When pairs of terms are found to be related, but are not mentioned in the same documents, they are an LBD candidate pair. The candidate pair is identified without any B linking terms. Additional processing outside of the LBD discovery is required to find linking B terms.

2.6 Information and Relationship Extraction

In Section 2.5.1, lack of automatic explanation of LBD candidate relationships was identified as a limitation of current LBD systems. This section presents an overview of information extraction (Section 2.6.1) and relationship extraction (Section 2.6.2) including two related topics: conditional random fields (Section 2.6.3) and open information extraction (Section 2.6.4).

2.6.1 Information Extraction

Information extraction (IE) aims to identify structured results describing things found inside the documents. IR (Section 2.2.1) is usually at the document level and assists in retrieving documents that are related to or actually mention specified terms or concepts (Jurafsky and Martin, 2009). To describe how IE returns structured information, consider the paragraph:

> "Eastman Kodak Co. (EKDKQ), based in Rochester, New York, emerged from bankruptcy today as a commercial-printing company that sells nothing to consumers. The new, smaller Kodak has shed its cameras, film sales and consumer photo developing that made it a household name. U.S. Bankruptcy Judge John Doe last month approved Kodak's exit plan. He cut about $4.1 billion of its debt and left shareholders empty-handed." [7]

One may wish to know more about entities described in this document. They may wish to find out more about "John Doe" and "Kodak". What is John Doe's title? Where is Kodak located? They may also wish to find out the relationship between John Doe and the shareholders and between Kodak and its product lines. IE performed on the paragraph determines that John Doe left shareholders empty-handed and Kodak used to sell cameras and film.

Named Entity Recognition (NER) may be employed to identify things found in the text being processed. For example, a person, location or organization. A named entity may have multiple annotations describing it. For example, "John Doe" is a named entity that is a person and is (most likely) a male. Person and male are annotations describing the actual named entity instance, "John Doe". NER is the process of identifying named entities and other meaningful concepts in text. (Jurafsky and Martin, 2009).

NER may be performed by systems that use grammatical or statistical processing techniques. Grammar-based NER systems like GATE (Cunningham et al., 2002, 2011)[8] and UIMA (Ferrucci and Lally, 2004)[9] usually make multiple passes across the text being analyzed with each pass providing more information about the text. Usually these steps are performed: The text is first tokenized or broken down to words, spaces, punctuation, etc. Next the text is compared to known lists of words that help identify them – this is done using gazetteers. An example would be a list of male first names. Then sentences are identified. Next the tokens are further analyzed to identify what parts of speech they represent. For example, identifying nouns, verbs, adjectives, participles, etc. Then the NER system will apply various sets of rules to further identify the semantics of the text. This may include pronominal and orthographical co-referencing which clarifies what pronouns like he, it, she or they are really referring to and joins repeated occurrences together as the same

[7] paraphrased from http://www.bloomberg.com/news/2013-09-03/kodak-exits-bankruptcy-as-printer-without-photographs.html

[8] https://gate.ac.uk/

[9] http://uima.apache.org

named entity. For example, in the Kodak example, Kodak is mentioned multiple times and is sometimes referred to as "it". Pronomial co-reference will clarify that "it" refers to Kodak and the orthographic co-reference will join together all instances of Kodak, including the references by pronouns.

Using the Kodak example, a grammatically based NER system may look at the paragraph and annotate it with the information that "John Doe" is a person and has a feature of being a male and he works as a "U.S. Bankruptcy Judge". The NER system will also find that "Rochester, NY" is a location, that "Eastman Kodak Co." is located there, that "Kodak" is the same as the organization "Eastman Kodak Co.", and that "$4.1 billion" is a money reference.

2.6.2 Relationship Extraction

In the previous section, an example about Kodak was studied. NER identifies the nouns, verbs and may even relate that Kodak is located in Rochester, New York. This *located in* relationship touches on another area of IE that is called relationship extraction which adds more semantics to the extracted entities. The relationship between Kodak and Rochester is the phrase "based in". This is an example of a binary (or 2-ary) relationship.

Jurafsky and Martin (2009) state that a relationship is a set of ordered tuples of instances of objects in a specific domain. For example, Kodak is "located in" Rochester, NY and John Doe "has job" as a judge. Here the relationships are binary. Relationship analysis builds off of named entity extraction where, for example, people, places, organizations, dates, etc, are identified in the text being analyzed. Then the relationships between these entities are identified.

Relationships may be more complicated (n-ary) where three or more entities are related in some sort of way. Multiple drugs having an ability to cure a disease is an n-ary relationship when each drug alone is not able to cure the disease. Another example may be that John makes $50,000 salary as of 1 October 2012 – this is a 3-ary relationship and does not as easily break up into two binary relationships because the salary is tied to the date and to John.

Li et al. (2008) group relationship extraction approaches into three types – co-occurrence analysis, rule-based approaches and statistical learning. They state that their work is specific to biomedical documents, but the same concepts should apply to other domains. Co-occurrence and rule-based approaches are basically an extension of the NER approaches to identifying concepts in text. Statistical learning is where relationship extraction may be treated as a classification problem. Relationships may also be identified using supervised and semi-supervised techniques (Jurafsky and Martin, 2009). An example is conditional random fields described in Section 2.6.3. Another type of relationship extraction is from the Open Information Extraction research (Open IE) (Etzioni et al., 2011) and is discussed more in Section 2.6.4. This form of relationship extraction is a semi-supervised approach – models are trained on known relationship models and then system tries to find hidden relationships.

SemMedDB (Section 2.4.2) is a compilation of relationship data that has been normalized to match UMLS relationship types and also to remove relationships that were deemed to be too general. In addition to providing binary relationships between concepts, SemMedDB also preserves the pedigree of the relationship by tracking the source document and sentence references.

2.6.3 Conditional Random Fields

Conditional Random Fields or CRFs are undirected graphical models that are trained to maximize the conditional probability of outputs given a set of inputs (Lafferty et al., 2001; Sutton and McCallum, 2007). Lafferty introduced the concept in 2001 and he and others have done much in applying CRFs to problems of relationship learning and extraction (Lafferty et al., 2001; Sutton and McCallum, 2007; Banko and Etzioni, 2008; Culotta et al., 2006; Fader et al., 2011; Tsuruoka et al., 2011). Sutton and McCallum (2007) describe two classes of models for analyzing graphical models. Generative models are those that are based on joint distributions – you must know the probability of both sides of the relationship. So for X and Y being random variables, the probability of X noted as P(X), and the probability of Y noted as P(Y), must both be known. An example of generative models are naïve Bayes models. Discriminative models are based on a model of conditional distributions and one does not need to know the P(X). Logistic regression is an example and is part of CRF methodologies (Lafferty et al., 2001; Sutton and McCallum, 2007). McCallum et al. (2000) describe maximum entropy Markov models (MEMMs) and note that MEMMs and CRFs are conditional probability finite state machines making them conditional and not generative.

In an interesting application of CRFs for identifying relationships, Culotta et al. (2006) proposed a supervised machine learning method that would assist in identifying familial relationships that may not be explicitly mentioned in the texts. The basic example they present discovers the unmentioned relationship that George W. Bush is the cousin of John Prescott Ellis. The facts found in the text include that George W. Bush is the son of George H. W. Bush, that George H. W. Bush is the sibling of Nancy Ellis Bush and that Nancy Bush's son is John Prescott Ellis. Using CRFs as implemented in their software system MALLET and its GRMM package, they extracted relations from text. The CRFs are trained using annotated corpus (70/30 split). They captured the models for 53 different relationship types including, for example, father, son, sister, brother, cousin, superior, member of, religion, job title, and friend. Once they captured the knowledge they used a closed-loop system that alternates between bottom-up extraction and top-down pattern discovery. Bottom-up where they learn everything from the data and then top-down to augment the data with discoveries based on known patterns and they would identify, for example, that George W. Bush is the cousin of John Prescott Ellis – a previously unmentioned relationship.

Relationship extraction using CRFs may not be needed in the MEDLINE domain studied here since SemMedDB will provide relationship data. However, CRFs may provide a possible solution for relationship explanation of LBD candidates in other domains.

2.6.4 Open Information Extraction

Banko and Etzioni (2008) define relationship extraction to be open relationship extraction when the relationship types or names are not known in advance. The system will use a set of relation-independent heuristics (or models). They classify CRF as an open extraction methodology. Etzioni et al. (2011) present an improved OpenIE methodology that improves precision and recall by two fold. In both papers, a system called ReVerb is presented that is able to extract relationships between entities without starting nor ending terms. ReVerb finds verb-based phrases and suggests them as the relationship between left and right sides of the relationship (Fader et al., 2011).

Etzioni et al. (2011) also present R2A2 which adds a learning capability called ArgLearner. ArgLearner uses classifiers to identify the left and right side arguments. The R2A2 system can learn phrases that represent relationships where ReVerb tries to do this automatically. the R2A2 system is trained using 20,000 sentences from the CoNLL 2005 Shared Task (Carreras and Màrquez, 2005) and generated 29,000 OpenIE tuples.

Both of these OpenIE systems are only able to extract simple relationships. However, they cannot identify n-ary relationships nor can they break up phrases like "Seattle Symphony Orchestra" and infer that there is a relationship tuple of (Orchestra, is the Symphony of, Seattle).

2.7 Classifiers

Classification is a supervised machine learning technique that discovers patterns in samples of understood data with known outcomes which are then used to to predict outcomes in new data (Witten et al., 2011; Cios et al., 2007; Marsland, 2015). Today, there are hundreds and maybe thousands of different classifiers available. Fernández-Delgado et al. (2014) studied 179 of them that spanned 17 families of classifier. In their work, they were trying to determine if they could identify the best classifier by studying many classifiers across many families of algorithms and across many data sets. They concluded that the family of random forest classifiers perform best but they also noted that many classifiers perform very similarly with only slight differences when compared with other classifiers. Demšar (2006), Lotte et al. (2007) and Williams et al. (2006) also studied various classifiers and conclude that many classifiers perform similarly. Demšar (2006), in particular, tried to find the best classifier using various tests of significance including t-tests and Wilcoxon tests to do side-by-side comparisons of classifier pairs. When considering classifier selection, attention to how they perform with respect to compute time is important. Some classifiers are very compute intensive and may not be practical for providing results in a timely fashion. The focus of this work is to determine if classification can assist in explaining LBDs and was not to seek out the perfect classifier for this problem set. The remainder of this section provides overviews of the classifiers applied to the problem of explaining LBDs.

Naïve Bayes

Naïve Bayes classifiers are based on Bayes theorem with the naïve part being the simplification that all features are independent. Bayes theorem states:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.3}$$

where $P(A)$ and $P(B)$ are the probability of A and of B respectively and $P(A|B)$ is the conditional probability of A given that B has been observed and $P(B|A)$ is the conditional probability of B given that A has been observed. (Barber, 2012; Jurafsky and Martin, 2009; Manning et al., 2008)

The general algorithm for a naïve Bayes classifier is to identify the probability of each feature instance occurring for each outcome independent of any other feature. Then identify the most likely outcome for a set of features using the simplified or naïve mathematics based on Bayes theorem assuming independence:

$$P(A|B) = \frac{P(b_1|A) \times P(b_2|A) \times \ldots \times P(A)}{P(B)} \tag{2.4}$$

where each possible feature data value of B is represented as b with subscripts (e.g. b_2) and $P(b_2|A)$ means the probability of b_2 given that the outcome A has been observed.

Decision Tree Classifier (J48 - C4.5)

J48 is an implementation of the C4.5 algorithm that is a decision tree classifier that extends the original work on the ID3 algorithm developed by Quinlan (1986). The ID3 algorithm is a recursive algorithm that creates subsets of the data based on values of single attributes. For example, break the training dataset into multiple subsets based on one of the features. Then the algorithm tries to decide if further subsets are required based on other features. Another subset is required if, after splitting based on one of the features, the remaining outcomes are not all the same. The algorithm completes when all outcomes at each leaf node are homogeneous. C4.5 algorithm is a refinement on the ID3 algorithm in that the choice of feature at each level is picked based on an information gain calculation.

An often used example that helps to describe decision tree classification is that which decides when to play outdoors (Quinlan, 1986; Witten et al., 2011). The variables are the outlook (sunny, overcast, raining), the temperature (hot, mild), the humidity (high, normal) and the wind (strong, weak). The logic for a decision tree will be to select an attribute, say, the first one of outlook. Three branches are generated for each possible value of the outlook. Then, the resulting outcomes are examined. If they are all the same, then stop expanding that branch. If they are not the same on the current branch, select another feature and perform the same branching steps and make the decision on whether or not to stop. This continues until all leaf nodes have the same outcome.

Decision trees are fundamentally easy to understand and work well with categorical data as is in the LBD problem space presented in this work. They also are tolerant to noisy data – cases where data may contain some nonsensical training instances. A downfall of decision trees is overfitting to the data.

Decision Tree Grafting

Work by Webb (1999) introduced additional nodes to a decision tree in an attempt to reduce errors in predicting outcomes. The goal was to obtain some benefits realized in decision committee methods but not be as slow to perform as those techniques. The additional nodes contain additional information not normally presented to the leaf nodes. These are the grafts, and they are handled as a post processing step to normal decision tree analysis. The key is that the basic computation is still tree based but accuracy of predictions may be improved. The implementation of this algorithm used here is called J48Graft.

Decision Table

Kohavi (1995) describes decision tables as a supervised learning algorithm in domains with discrete feature sets. Decision tables are similar to decision trees but present a different way to view the possibilities. A tree studies each feature in succession seeking to find ultimate leaf nodes where all outcomes are the same (this is when to stop computing on the tree traversal). A table has, for example all possible feature types as rows including the outcome (e.g. A semantic type, B semantic type, etc, and the A to C relationship) and the columns represent possible values identified in the training data. Additional information about each row-column intersection may exist like a percentage score of how many times that cell is encountered in the training data. This classifier is one that bases its choices on majorities.

Decision Table and Naïve Bayes Hybrid

Hall and Frank (2008) present a combination of a decision tree and the naïve Bayes algorithms as a classifier. This classifier uses the decision table as a source of conditional probabilities and then applies naïve Bayes logic to produce ultimate outcomes. Effectively, a decision table is first developed for the entire training set and then parallel decision tables and naïve Bayes models are generated and results from each set are combined to produce predictions. The implementation of this algorithm used is called DTNB.

Partial Decision List With Separate and Conquer on C4.5

Frank and Witten (1998) present a partial decision tree classifier that is based on both the C4.5 algorithm and on pruning techniques that they call PART. In Cohen (1995), the RIPPER algorithm is described which presents some of the pruning techniques used to create the PART algorithm. RIPPER is more efficient and better for larger data sets than pure C4.5 algorithms. The PART algorithm focuses on developing partial trees and employs a divide and conquer approach to pruning that doesn't require global optimizations. Effectively, the PART algorithm ends up pruning some leaves or even partial branches leaving simpler sets of rules in the final decision tree.

Random Forest

A random forest classifier builds on single decision tree classifiers by randomly select different starting points for the trees and combine these results in, theoretically, a more accurate classifier (Breiman, 2001; Liaw and Wiener, 2002; Rodriguez et al., 2006). By combining multiple runs, random forest classifiers are an example of an ensemble classifier (See Section 2.8) based on a single tree classifier algorithm. Random forest algorithms help prevent overfitting to the data – a problem that a decision tree algorithm, by itself, may have.

Sequential Minimal Optimization

Sequential Minimal Optimization (SMO) is an implementation of an SVM that is optimized by breaking the quadratic programming (QP) problem associated with SVMs into the smallest QP problems (Platt, 1998). Support-Vector Machines (Cortes and Vapnik, 1995), which are also known as SVMs, present a classification methodology that, in addition to providing a solution to linearly separated data, provides solutions to classifying non-linear separations of training data sets and provides maximum separation of the hyperplanes that separate the classes of data.

The mathematics behind SVMs inherently become very computationally expensive as each hyperplane represents, at a minimum, a quadratic equation and in a multi-feature classification problem, many equations must be solved solve. Vapnik, Cortes, Platt and others developed various optimizations to SVM classification including SMO.

K-nearest neighbors classifier

K-nearest neighbor (k-NN) classifiers are instance based learning algorithms (Witten et al., 2011). IBk is an implementation of k-NN. The concept of this algorithm when applied to numerical data is to classify a feature based on its closest neighbor for pairwise comparison with one common method being the use of euclidean distance calculations to identify most closely matching entities. When applied as a text classifiers, some form of matching to the training data is necessary. The k refers to how many neighbors to consider in identifying a match.

Instance based learning may be thought of as rote or memorization learning – patterns are stored and indexed for lookup, later, during the task of classification. The outcome of the closest match of features is returned as the prediction. When k is greater than one, some form of voting decides which outcome to present.

2.8 Ensemble Learning

Ensemble learning is a specialization of classification. Instead of using a single classifier and its predictions, ensemble learning combines the results of multiple classifiers to identify predictions. There have been many ensemble schemes identified by various researchers – for example, Dietterich (2000); Witten et al. (2011); Hung and Chen (2009); Rodriguez et al. (2006); Tan and Gilbert (2003); Wang et al. (2011). Three prominent ensemble methods discussed here are bagging (Section 2.8.1), boosting (Section 2.8.2) and stacking (Section 2.8.3).

2.8.1 Bagging

The term bagging is derived from *bootstrap aggregation* and consists of combining the outcomes of multiple classifier models created using the same classifier algorithm by taking some form of vote – for example, majority (most match) or consensus (all match) with each contributor having equal weighting. Bagging is often used on decision tree or other classifier algorithms that may have overfitting tendencies on some data sets (usually smaller data sets) (Wang et al., 2011; Witten et al., 2011). Random forest classification algorithm is a bagging algorithm applied to random trees (Liaw and Wiener, 2002; Breiman, 2001). These are a summary of the steps in training a bagging algorithm classifier:

- Break the training data into multiple subsets
- Train a classifier multiple times using some of the training subsets (not all but overlap is fine)
- Use all of the trained classifier models on the test data and choose the outcome that shows up most often (voting majority)

Breiman (1996) concludes with the following quote:

> "Bagging goes a ways toward making a silk purse out of a sow's ear, especially if the sow's ear is twitchy."

2.8.2 Boosting

Boosting combines results of classifiers in ways where some classifier results are weighted more than others. Where bagging training may be run for each subset of training data in parallel, boosting requires that each model is trained serially – successful or unsuccessful classifications from each model trained are fed to the next model allowing weighting to be applied to the next model's outcomes thus boosting its performance (Witten et al., 2011; Bauer and Kohavi, 1999; Schapire, 2013). That is, boosting algorithms adjust their design on each learning run where bagging algorithms do not (Bauer and Kohavi, 1999). These are a summary of steps in training a boosting algorithm classifier:

- Start with all training instances having equal weight
- For n iterations
    - Train the classifier
    - Compare results with expected outcome
    - For next iteration, weight higher the results *misclassified*

2.8.3 Stacking

Stacking, also known as stacked generalization, was identified by (Wolpert, 1992) and is a method that effectively uses bagging but not necessarily on the same classification algorithm. Stacking is not as commonly used as bagging and boosting schemes usually because it more difficult to analyze theoretically (Wang et al., 2011). These are a summary of steps in a stacking algorithm classifier:

- Train a number of base classifiers using normal cross-validation techniques and different classifier algorithms
- Build a new training set that includes results from the base classifiers as additional instance information (or use only the prediction data as instance data)
- Training a higher level classifier sometimes called a metalearner or metaclassifier using the new training set

2.9 Chapter Summary

This chapter has presented a literature review including an overview of LBD – what it is, how it is done and what limitations there are in current approaches to it. Then this chapter continued into studies of relationship extraction, evaluation methodologies and ensemble learning techniques which are all important for subsequent experimentation. The rest of this chapter discussed topics more tightly integrated in this thesis – one was an introduction to the MEDLINE domain and the other was definition of terminology used throughout the rest of this work.

Chapter 3

Approach and Data for Relationship Prediction

This chapter first describes the overall approach used to explain the natures of the relationships between candidate LBD pairs using classification and then this chapter describes the data designs presented to classifiers. Chapter 4 uses these data designs to study performance with various classifiers and ensemble learners and Chapter 5 applies the classifiers to the task of explaining actual LBD relationships.

Explaining the relationships of candidate LBD pairs (the A to C relationship) will require the identification of linking B terms and the natures of the relationships between A and B and B and C (the A-B-C triples nomenclature is described in Sections 2.1.1 and 2.1.3). Information from these partially defined LBD triples will be presented to the classifiers in an effort to predict the natures of the relationships of candidate LBD pairs.

To develop a classification solution, a feature set must be designed that captures enough distinguishing information about the A, B and C concepts and the A to B and B to C relationships that the classifiers may be able to perform better than baselines but not be too specific that the classifiers cannot predict nor compute. Then training data needs to be identified and developed. This will include finding a source of gold standard facts from which the classifiers may be trained and evaluated. Additionally, variations of the data will be required to more thoroughly evaluate the designs and performance of the explored solutions. These data-related items are discussed in the rest of this chapter.

After the feature designs and training data are established, Chapter 4 presents the studies of the creation and evaluation of the classifier models using various standard methods like cross-validation and hypothesis testing. Additionally, that chapter studies improvements to the data designs with the goal of identifying the feature sets and classifier models for use in explaining LBD relationships.

Chapter 5 presents LBD on an older time-slice of documents using Semantic Vectors, a co-occurrence approach. That chapter also presents the steps that prepare the discoveries for the classification that will explain the nature of the LBD relationships. This is done by completing the partial triples by identifying linking B terms and by finding the A to B and B to C relationship types. The next and a very important step presented in Chapter 5 is to validate any prediction suggested by the classifiers by studying newer literature to see if the discoveries have ever been explained (either agreeing with prediction or contradicting the prediction). The last topic presented in Chapter 5 is to study ensemble learners in an attempt to improve the accuracy of LBD A to C relationship predictions.

To summarize, the approach presented in this work to explain LBD relationships includes these steps:

1. Definition of classification feature sets (Chapter 3)
2. Identification and development of training data (Chapter 3)
3. Application and evaluation of classifiers (Chapter 4)
4. Improvements of feature designs (Chapter 4)
5. Identification and preparation of LBD candidates (Chapter 5)
6. Application of classifiers to the LBD candidate pairs (Chapter 5)
7. Validation of the results using time-slice approach (Chapter 5)
8. Study of ensemble learners including validation (Chapter 5)

The rest of this chapter discusses the data used for relationship prediction including concept selection (Section 3.1), data used for training classifiers (Section 3.2), the feature design for supervised learning or classification (Section 3.3), the data format presented to the classifiers (Section 3.4), the refinement of data designs that involve removal of instances containing commonly occurring feature values, and regularization of the predicted outcomes (both discussed in Section 3.5). This chapter ends with a summary of all of the experimental data that will be used throughout Chapter 4 for classifier experimentation (Section 3.6).

3.1 Concept Selection

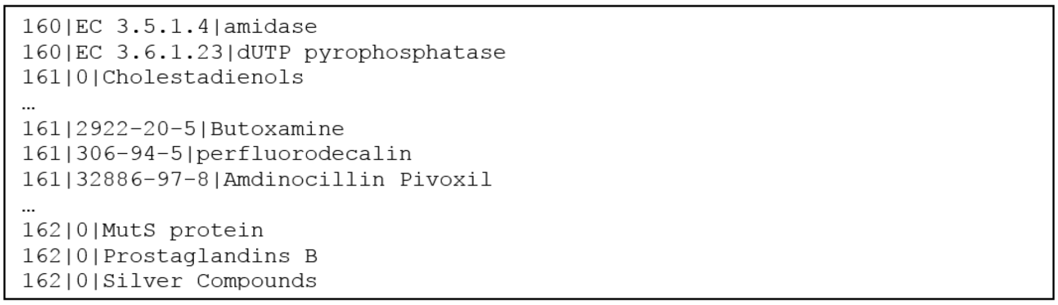
This section discusses concept selection. The approach in this work uses statistical methods to perform LBD (see Sections 2.2.3 and 5.2). To do this, concepts are known ahead of time and, after indexing the corpus, LBD candidates are identified by searching for concept pairs and examining their relatedness scores. If the score is considered significant and the concepts are not mentioned together in any document in the corpus, a candidate LBD pair has been identified. This methodology requires a set of concepts for which look-ups are performed. An exhaustive list could be used but there are two problems with that approach – one is simply the computational complexity of identifying how all concepts relate to all other concepts. Even with enhancements based on LSI and the implementations with RRI like Semantic Vectors, these processes still take a great amount of time to compute. The other problem is that the statistical methods will not identify concepts as being related if the concept is only mentioned once or twice in the corpus. Therefore the set of concepts needs to include those concepts that appear a significant number of times in order for the statistical methods to present meaningful results. The approach to LBD in this work that uses statistical co-occurrence methods in a closed LBD system uses a limited set of concepts as also suggested by Weeber et al. (2001). The system is closed because the set of candidate C terms that will be found to be related to some A terms is limited by the list of concepts described in this section.

The concepts used in this work come from MEDLINE (Section 2.4) and in particular from the 2002 version of the MEDLINE/PubMed Baseline Repository.[1] The early year was chosen to provide a smaller data set with which to work and to reduce the computational expense that the newer, larger versions may have introduced. The MEDLINE repository contains two files containing statistics on how many times certain terms appear in the corpus. The files are Chemical_freq_alpha that lists chemical concepts, and MH_freq_count that lists medical subject heading concepts (MeSH). The terms in the MeSH file are the primary topics found in MEDLINE repository and the chemical list is a list primarily focused on chemicals and drugs. In addition to those chosen from the MEDLINE files, some additional concepts are identified in works by Swanson (1986a,b) and Kostoff et al. (2007). These sources combine provide the set of concepts used in this work. The set is not meant to be exhaustive, but instead, was meant to provide a list large enough to be interesting.

[1] https://mbr.nlm.nih.gov/Download

The Chemical_freq_alpha file is a text file contains that frequency count information on 123,550 chemicals. Figure 3.1 contains a few samples from this pipe (|) delimited file. The three columns separated by the pipes represent:

1. The overall frequency count in the MEDLINE corpus
2. The MEDLINE registry number for the chemical
3. The chemical name



```
160|EC 3.5.1.4|amidase
160|EC 3.6.1.23|dUTP pyrophosphatase
161|0|Cholestadienols
…
161|2922-20-5|Butoxamine
161|306-94-5|perfluorodecalin
161|32886-97-8|Amdinocillin Pivoxil
…
162|0|MutS protein
162|0|Prostaglandins B
162|0|Silver Compounds
```

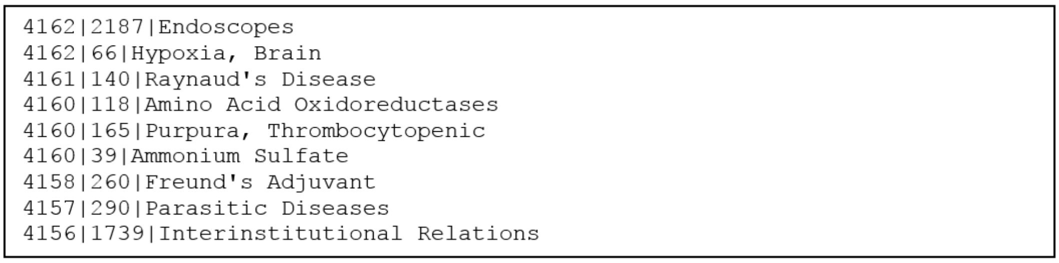
Fig. 3.1 Sample from Chemical_freq_alpha file

Chemicals appearing 100 or more times are selected and added to the concept list. This number was selected after considering the full list of chemicals and realizing that infrequently appearing concepts would not provide enough documents from which B linking terms could be identified. If a concept only appears a few times in the corpus, there is less chance of finding common linking terms and less chance that statistical co-occurrence techniques would find any related concept pairs that include the infrequently occurring concept. After applying this filter, a total of 9,317 chemicals were identified and added to the list of concepts.

The MH_freq_count text file contains the frequency count information on 19,781 main subject headings. Figure 3.2 contains a few samples from this file which is also pipe delimited. The three columns represent:

1. The overall frequency count in the MEDLINE corpus
2. The count when the heading is the main or starred item
3. The main heading name

Starred main headings appearing 100 or more times are selected and added to the concept list. Again, this limit was chosen to help provide a large enough document set from which statistically related concepts and B linking terms could be identified. After applying this filter, a total of 8,103 starred main headings were identified and added to the list of concepts.



```
4162|2187|Endoscopes
4162|66|Hypoxia, Brain
4161|140|Raynaud's Disease
4160|118|Amino Acid Oxidoreductases
4160|165|Purpura, Thrombocytopenic
4160|39|Ammonium Sulfate
4158|260|Freund's Adjuvant
4157|290|Parasitic Diseases
4156|1739|Interinstitutional Relations
```

Fig. 3.2 Sample from MH_freq_count file

The list of concepts is further augmented by adding concepts from the Swanson (1986a, 1988) and Kostoff et al. (2007) papers. These additional 248 concepts were added so that there may be more familiar literature based discoveries like those related to Raynaud's syndrome, migraines, Parkinson's disease, multiple sclerosis and cataracts. The concepts from others' research include those found as candidate LBD pairs in addition to linking B terms (i.e. they were from the set of A, B and C concepts).

A total of 17,668 total concepts are produced by these three selection steps. After duplicates are removed along with those that are not found to be mapped in the UMLS Metathesaurus (more on why this is important is discussed in Section 5.1.1), a master list of 15,427 concepts remains. This is the set that will be used when performing relatedness studies using Semantic Vectors package as described in Section 5.2. Additionally, this list is used to initiate the queries that build the fully qualified triples in Section 3.2.

3.2 Generation of Training Data

In Sections 2.1.1 and 2.1.3, the A-B-C triples nomenclature was introduced and will continue to be used extensively throughout the rest of this work. Classifiers will be trained using fully qualified triples which are those where the A, B and C concepts are known along with the nature of the relationships between A and B, B and C and C and A term pairs of the triple.

Supervised machine learning classifiers are trained using sets of training data that contain a set of feature instances and their predicted outcomes. A discovery identified with LBD consists of an A and a C concept and associated linking B terms. Since the primary purpose of this thesis is to automatically explain LBD relationships, the nature of the A to C relationship must be known in the training data. That is, the A to C relationship is the outcome or the prediction of the classifier. Other features may be presented to the classifiers – for example, the nature of the relationships between the A and B and between the B and C concepts. Details of these features are discussed in Section 3.3.

This section describes how a set of training data is generated using a set of facts identified by other researchers. SemMedDB (Section 2.4.2) provides a set of data that, for the purposes of this research, will be considered as a gold standard from which fully qualified triples are identified and turned into training data for the classification tasks. These triples consist of a set of A, B and C concepts along with A-B, B-C and C-A relationship types including the direction of each relationship.

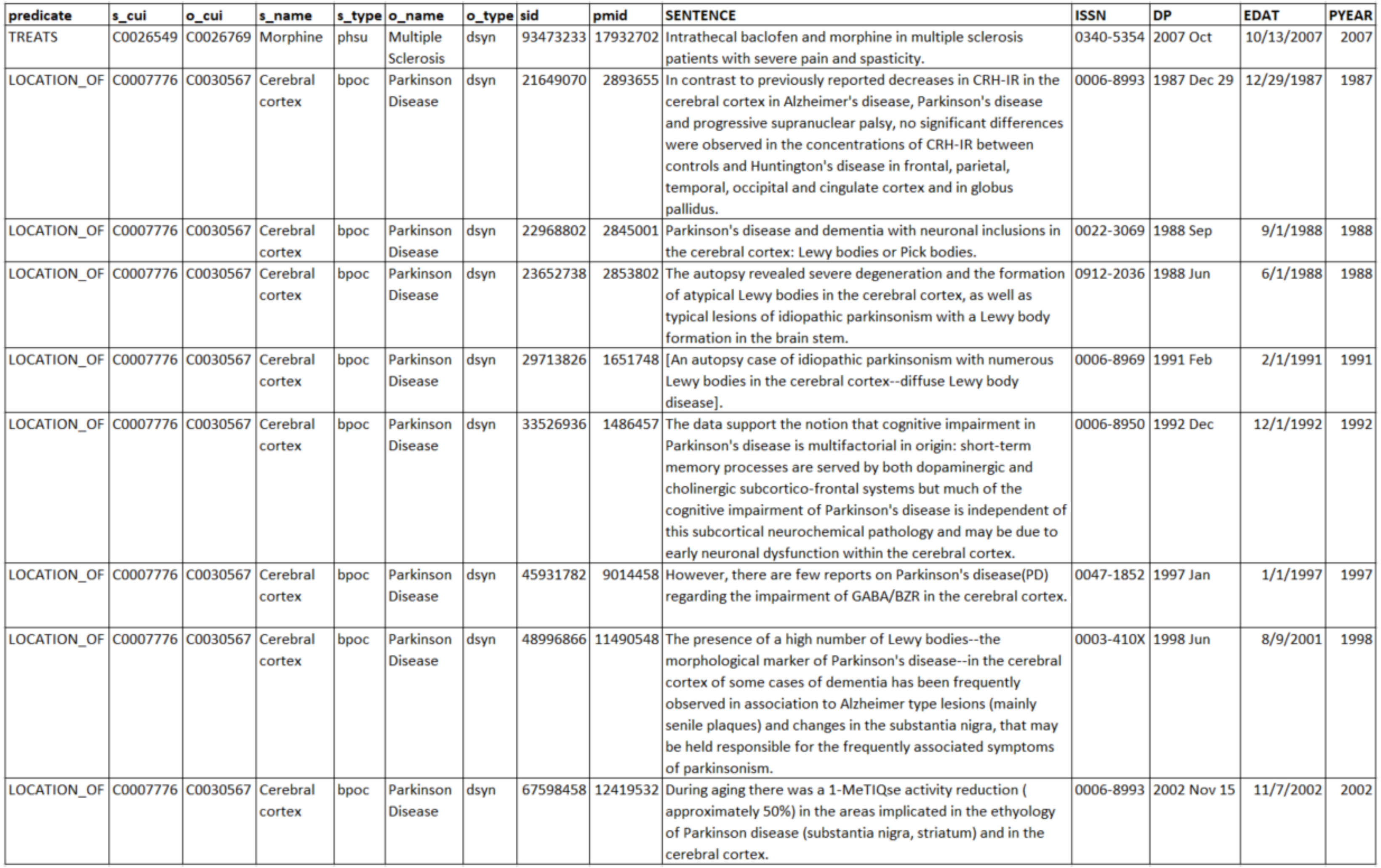
SemMedDB is a MySQL database and has tables containing the concepts, their semantic types and relationships between concepts. A total of 1,339,227 concepts, 68,000,470 relationships between concepts and 139,957,647 sentences are stored in the version of SemMedDB used in this work. Additional tables contain semantic type information for the concepts and citation information for the stored sentences. The relationships stored in SemMedDB are stored with references to the actual sentences, article ids and dates from where the relationship was derived.

A sample query of SemMedDB to find sentences mentioning Multiple Sclerosis (CUI of C0026769) and Morphine (C0026549) and mentioning Parkinson's Disease (C0030567) and Cerebral Cortex (C0007776) is as shown below.

```
SELECT distinct(pa.predicate), pa.s_cui, pa.o_cui, pa.s_name,
       pa.s_type, pa.o_name, pa.o_type, pa.sid, pa.pmid,
       s.SENTENCE, c.ISSN, c.DP, c.EDAT, c.PYEAR
FROM semmedver24.predication_aggregate pa,
     semmedver24.sentence s, semmedver24.citations c
where (((pa.s_cui = "C0026769" and pa.o_cui = "C0026549")
   or (pa.s_cui = "C0026549" and pa.o_cui = "C0026769")) or
   ((pa.s_cui = "C0030567" and pa.o_cui = "C0007776")
   or (pa.s_cui = "C0007776" and pa.o_cui = "C0030567")))
   and s.sentence_id = pa.sid and pa.pmid = c.pmid
```

This query produces results like the subset shown in Table 3.1. Note that the tables in this query are the predication_aggregate table where the relationship between two concepts is found and in the sentence and citation tables where the information supporting the relationship is found.

The logic used to find fully qualified triples in SemMedDB to use as training data is as shown in Figure 3.3. The initial A concepts are the only ones that come from the list of concepts identified in Section 3.1. The B and C concepts need not be constrained because including any related concepts is acceptable. Having a list of concepts becomes most



predicate	s_cui	o_cui	s_name	s_type	o_name	o_type	sid	pmid	SENTENCE	ISSN	DP	EDAT	PYEAR
TREATS	C0026549	C0026769	Morphine	phsu	Multiple Sclerosis	dsyn	93473233	17932702	Intrathecal baclofen and morphine in multiple sclerosis patients with severe pain and spasticity.	0340-5354	2007 Oct	10/13/2007	2007
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	21649070	2893655	In contrast to previously reported decreases in CRH-IR in the cerebral cortex in Alzheimer's disease, Parkinson's disease and progressive supranuclear palsy, no significant differences were observed in the concentrations of CRH-IR between controls and Huntington's disease in frontal, parietal, temporal, occipital and cingulate cortex and in globus pallidus.	0006-8993	1987 Dec 29	12/29/1987	1987
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	22968802	2845001	Parkinson's disease and dementia with neuronal inclusions in the cerebral cortex: Lewy bodies or Pick bodies.	0022-3069	1988 Sep	9/1/1988	1988
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	23652738	2853802	The autopsy revealed severe degeneration and the formation of atypical Lewy bodies in the cerebral cortex, as well as typical lesions of idiopathic parkinsonism with a Lewy body formation in the brain stem.	0912-2036	1988 Jun	6/1/1988	1988
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	29713826	1651748	[An autopsy case of idiopathic parkinsonism with numerous Lewy bodies in the cerebral cortex--diffuse Lewy body disease].	0006-8969	1991 Feb	2/1/1991	1991
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	33526936	1486457	The data support the notion that cognitive impairment in Parkinson's disease is multifactorial in origin: short-term memory processes are served by both dopaminergic and cholinergic subcortico-frontal systems but much of the cognitive impairment of Parkinson's disease is independent of this subcortical neurochemical pathology and may be due to early neuronal dysfunction within the cerebral cortex.	0006-8950	1992 Dec	12/1/1992	1992
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	45931782	9014458	However, there are few reports on Parkinson's disease(PD) regarding the impairment of GABA/BZR in the cerebral cortex.	0047-1852	1997 Jan	1/1/1997	1997
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	48996866	11490548	The presence of a high number of Lewy bodies--the morphological marker of Parkinson's disease--in the cerebral cortex of some cases of dementia has been frequently observed in association to Alzheimer type lesions (mainly senile plaques) and changes in the substantia nigra, that may be held responsible for the frequently associated symptoms of parkinsonism.	0003-410X	1998 Jun	8/9/2001	1998
LOCATION_OF	C0007776	C0030567	Cerebral cortex	bpoc	Parkinson Disease	dsyn	67598458	12419532	During aging there was a 1-MeTIQse activity reduction (approximately 50%) in the areas implicated in the ethyology of Parkinson disease (substantia nigra, striatum) and in the cerebral cortex.	0006-8993	2002 Nov 15	11/7/2002	2002

Table 3.1 SemMedDB Sample Data

```
Require: consider only articles from 1980 and newer
for each A concept do
    query SemMedDB for relationships between A and any B
    for each B concept with A to B relationship do
        query SemMedDB for relationships between B and any C
        for each C concept with B to C relationship do
            query SemMedDB for relationship between C and current A

            if relationship found between A and C then
                if none of A, B or C relationships include a disease of interest then
                    if all relationships are from 1980-1984 then
                        save in set of 1980-1984 training data set
                    else
                        save in set of post-1984 training data set
                    end if
                end if
            end if

        end for
    end for
end for
```

Fig. 3.3 Finding fully qualified triples in SemMedDB

important when performing LBD (Section 5.2) so that there are known concepts to index and compare to identify statistically related concepts. A selection of all distinct concepts in the database would have been another way to initiate the queries for finding fully qualified triples (e.g., a SQL statement like "select distinct concept from table").

To identify training data, first, using a starting set of A concepts, perform a query for A and related B candidates, including the nature of the relationship between the A and B concept. The articles must also be from 1980, and newer. Then, for each B identified in the first query, another query is performed that identifies a set of C candidates and the relationship between the B and C concept. Last, a query for C concepts back to the A concept is performed and, if found, a possible training candidate has been identified. Additionally, if A, B or C includes a disease of interest, it is ignored. This is important because the LBD studies presented in this work focus on a set of diseases, and the trained classifiers would be tainted if diseases of interest were included in the training data. If the date of all of the articles supporting the A-B-C relationship triple is from 1980-1984, the training candidate is stored in the 1980-1984 set. Otherwise, the candidate is stored in the post-1984 set.

The results of the SemMedDB queries consist of two sets of data – one set of fully qualified triples for the date range of 1980-1984 and another set for any date after 1984 (referred to as post-1984). These dates were chosen to coincide with the time-slice dates that will be used in the LBD experiments in Chapter 5. A total of 4997 fully qualified triples were identified in the 1980-1984 time slice and 203,323 in the post-1984 set. The post-1984 set of data may be considered for training since it does not contain the diseases of interest and it provides a much larger set of data than the 1980-1984 time slice.

As a note of further clarification, the results of work presented in this chapter include a set of classifiers trained on older 1980-1984 data. This set of classifiers will be used on the LBD candidates to explain the nature of their A to C relationships. The LBD candidates are also identified in the older time slice of data, thus all the data used in the identification of LBD candidates and the explanation of them comes from the same time slice of data.

3.3 Feature Design for Supervised Learning

The feature design for the inputs to classification, in this work, consists of information about the nodes of the triples and the relationships between them. These nodes and relationships are what provides enough detail that the supervised learning is able to produce plausible predictions about the A and C terms' relationships of the partial triples. When modeling medical data and simulating the previous work in this area of Swanson (1986b) and of Kostoff (2008) diseases are found to be related to other diseases, pharmaceuticals or other medical semantic types. The nodes of the triples represent the literal concepts, but for machine learning in the form of classification, only the semantic type of the concept is used. Marsland (2015) points out that, when defining the knowledge representation in supervised machine learning, the number of classes used must be small enough for the classifiers to be able to compute. With this in mind, only the semantic type is used and not each of the tens of thousands of concepts names when training the classifiers. The concept named Raynaud's disease, for example, is represented in the feature design as a disease or syndrome (which is the *dsyn* semantic type in MEDLINE).

The nature and direction of the relationships are more literal but must use generalized names (see Section 2.4.1). The direction could have been treated as a separate class from the relationship type, but very early experimentation showed this refinement of classes did not help and only complicated and confused the classifiers. Therefore, to keep the number of classes as small as was reasonable, the direction and type of relationship are combined as one class type. Additionally, Section 3.5.2 discusses some other class design ideas that were explored.

To summarize, these five features are used for LBD relationship prediction using classification:

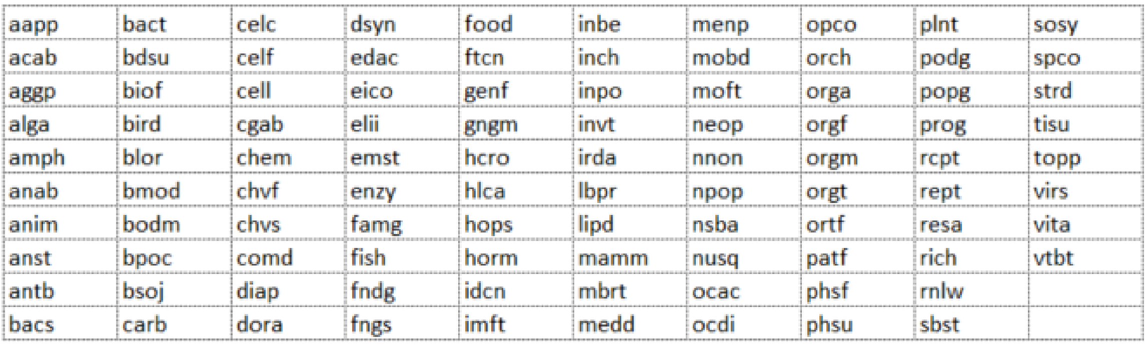
- The semantic type of A
- The semantic type of B
- The semantic type of C
- The relationship and direction between A and B
- The relationship and direction between B and C

The relationship and direction between C and A is the class being predicted by the classifiers – it is the unknown or the outcome of the classifier.

This representation using the five classes is how fully and partially qualified triples are presented to the classifier algorithms with partial triples being those where the the relationship and direction between C and A nodes of the triple is omitted. Training is performed with data that includes the additional facts about the outcome to be predicted by the classifier (that is, C to A relationship types including the direction of the relationship).

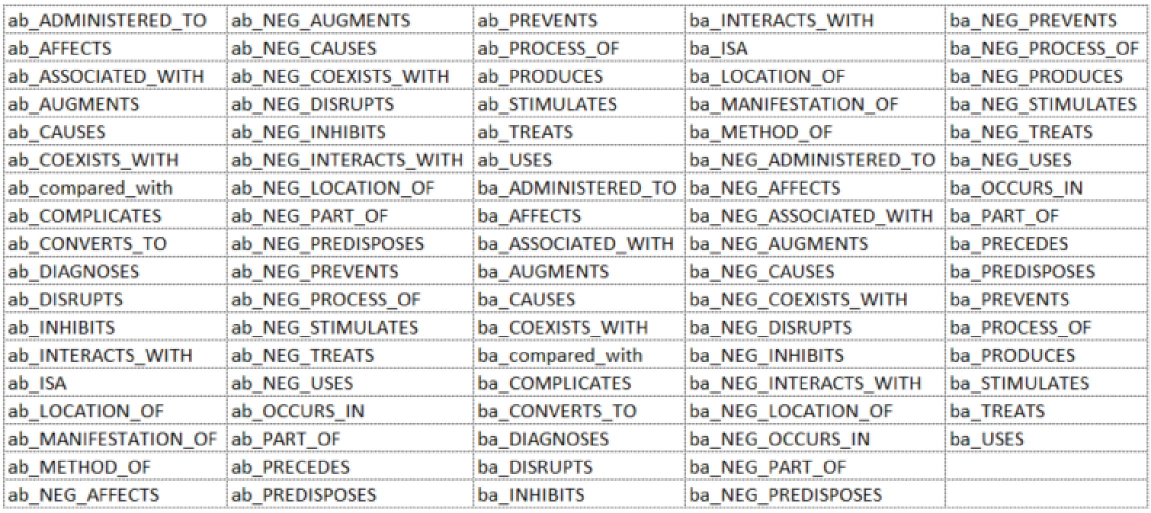
For the various classification experiments presented in the rest of this work, a set of 98 unique semantic types was drawn from the training data – the set is not an exhaustive list of all possible 134 semantic types because some semantic types never show up in the data studied and, therefore, need not be included.[2] 69 of the semantic types appear in the training data for 1980-1984 and all 98 appear in the Post-1984 training data. For the same reason of not including data that never appears, the set of relationships differs for three relationship and direction classes (A to B with direction, B to C with direction and C to A with direction). The differences in the relationships are based on training data available. A to B has 88 unique relationship training instances with 46 of these appearing in the 1980-1984 data and 88 appearing in the Post-1984 data; B to C has 89 total with 57 in the 1980-1984 and 89 in the post-1984 data; and C to A has 107 with 67 in the 1980-1984 and 107 in the Post-1984 data ("unknown" is not included in the C to A totals since it is a requirement of the software used to perform classifier training). Tables 3.2, 3.3, 3.4 and 3.5 contain the complete lists of semantic types and relationships considered in initial experiments.

[2] Full list of semantic types may be found here: https://mmtx.nlm.nih.gov/MMTx/semanticTypes.shtml



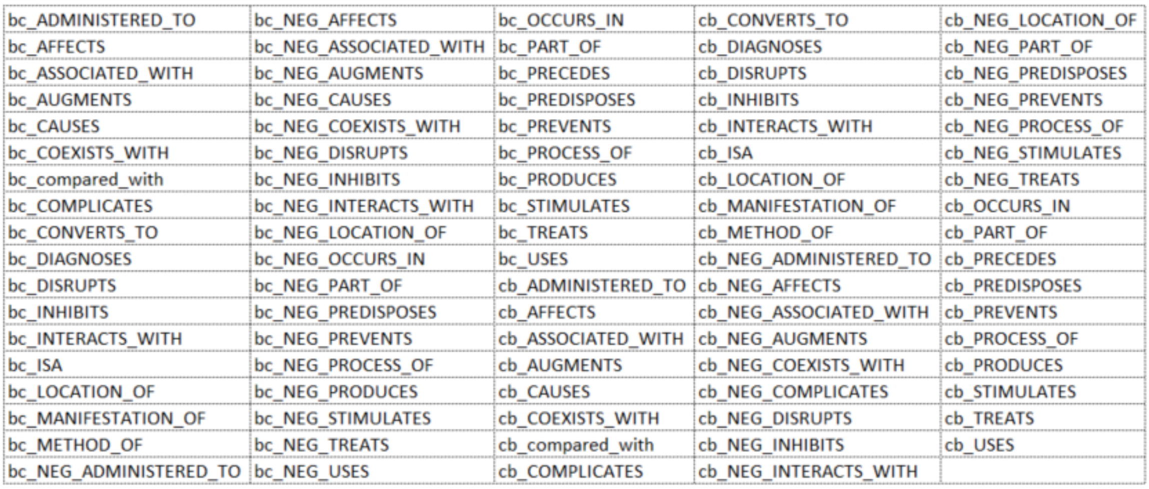
aapp	bact	celc	dsyn	food	inbe	menp	opco	plnt	sosy
acab	bdsu	celf	edac	ftcn	inch	mobd	orch	podg	spco
aggp	biof	cell	eico	genf	inpo	moft	orga	popg	strd
alga	bird	cgab	elii	gngm	invt	neop	orgf	prog	tisu
amph	blor	chem	emst	hcro	irda	nnon	orgm	rcpt	topp
anab	bmod	chvf	enzy	hlca	lbpr	npop	orgt	rept	virs
anim	bodm	chvs	famg	hops	lipd	nsba	ortf	resa	vita
anst	bpoc	comd	fish	horm	mamm	nusq	patf	rich	vtbt
antb	bsoj	diap	fndg	idcn	mbrt	ocac	phsf	rnlw	
bacs	carb	dora	fngs	imft	medd	ocdi	phsu	sbst	

Table 3.2 List of Feature Values for A, B and C Semantic Types (98 total)



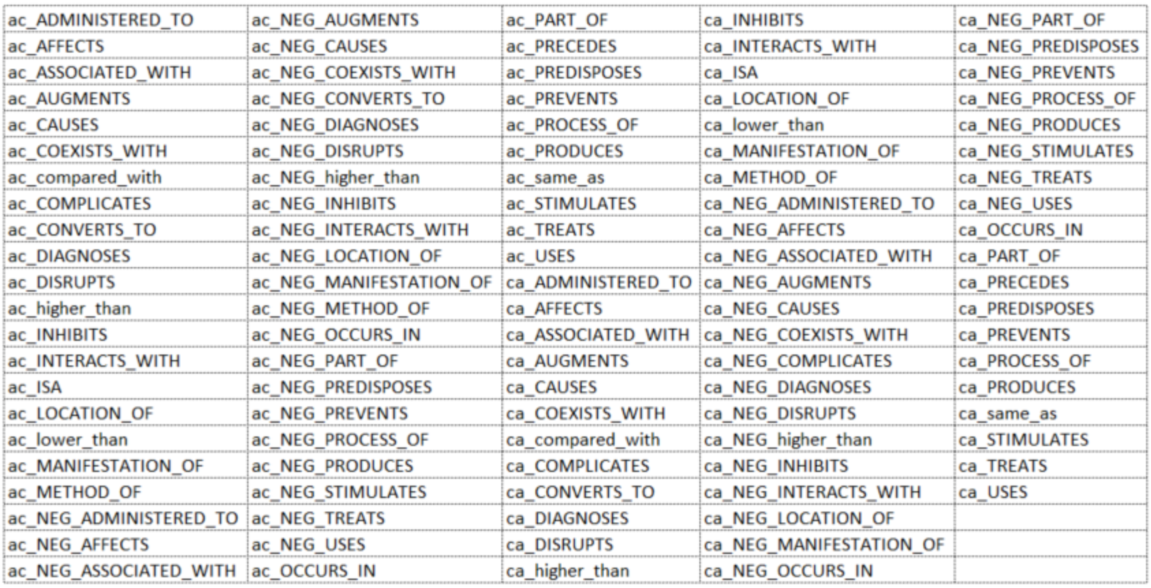
ab_ADMINISTERED_TO	ab_NEG_AUGMENTS	ab_PREVENTS	ba_INTERACTS_WITH	ba_NEG_PREVENTS
ab_AFFECTS	ab_NEG_CAUSES	ab_PROCESS_OF	ba_ISA	ba_NEG_PROCESS_OF
ab_ASSOCIATED_WITH	ab_NEG_COEXISTS_WITH	ab_PRODUCES	ba_LOCATION_OF	ba_NEG_PRODUCES
ab_AUGMENTS	ab_NEG_DISRUPTS	ab_STIMULATES	ba_MANIFESTATION_OF	ba_NEG_STIMULATES
ab_CAUSES	ab_NEG_INHIBITS	ab_TREATS	ba_METHOD_OF	ba_NEG_TREATS
ab_COEXISTS_WITH	ab_NEG_INTERACTS_WITH	ab_USES	ba_NEG_ADMINISTERED_TO	ba_NEG_USES
ab_compared_with	ab_NEG_LOCATION_OF	ba_ADMINISTERED_TO	ba_NEG_AFFECTS	ba_OCCURS_IN
ab_COMPLICATES	ab_NEG_PART_OF	ba_AFFECTS	ba_NEG_ASSOCIATED_WITH	ba_PART_OF
ab_CONVERTS_TO	ab_NEG_PREDISPOSES	ba_ASSOCIATED_WITH	ba_NEG_AUGMENTS	ba_PRECEDES
ab_DIAGNOSES	ab_NEG_PREVENTS	ba_AUGMENTS	ba_NEG_CAUSES	ba_PREDISPOSES
ab_DISRUPTS	ab_NEG_PROCESS_OF	ba_CAUSES	ba_NEG_COEXISTS_WITH	ba_PREVENTS
ab_INHIBITS	ab_NEG_STIMULATES	ba_COEXISTS_WITH	ba_NEG_DISRUPTS	ba_PROCESS_OF
ab_INTERACTS_WITH	ab_NEG_TREATS	ba_compared_with	ba_NEG_INHIBITS	ba_PRODUCES
ab_ISA	ab_NEG_USES	ba_COMPLICATES	ba_NEG_INTERACTS_WITH	ba_STIMULATES
ab_LOCATION_OF	ab_OCCURS_IN	ba_CONVERTS_TO	ba_NEG_LOCATION_OF	ba_TREATS
ab_MANIFESTATION_OF	ab_PART_OF	ba_DIAGNOSES	ba_NEG_OCCURS_IN	ba_USES
ab_METHOD_OF	ab_PRECEDES	ba_DISRUPTS	ba_NEG_PART_OF	
ab_NEG_AFFECTS	ab_PREDISPOSES	ba_INHIBITS	ba_NEG_PREDISPOSES	

Table 3.3 List of Feature Values for A to B and B to A Relationships (88 total)



bc_ADMINISTERED_TO	bc_NEG_AFFECTS	bc_OCCURS_IN	cb_CONVERTS_TO	cb_NEG_LOCATION_OF
bc_AFFECTS	bc_NEG_ASSOCIATED_WITH	bc_PART_OF	cb_DIAGNOSES	cb_NEG_PART_OF
bc_ASSOCIATED_WITH	bc_NEG_AUGMENTS	bc_PRECEDES	cb_DISRUPTS	cb_NEG_PREDISPOSES
bc_AUGMENTS	bc_NEG_CAUSES	bc_PREDISPOSES	cb_INHIBITS	cb_NEG_PREVENTS
bc_CAUSES	bc_NEG_COEXISTS_WITH	bc_PREVENTS	cb_INTERACTS_WITH	cb_NEG_PROCESS_OF
bc_COEXISTS_WITH	bc_NEG_DISRUPTS	bc_PROCESS_OF	cb_ISA	cb_NEG_STIMULATES
bc_compared_with	bc_NEG_INHIBITS	bc_PRODUCES	cb_LOCATION_OF	cb_NEG_TREATS
bc_COMPLICATES	bc_NEG_INTERACTS_WITH	bc_STIMULATES	cb_MANIFESTATION_OF	cb_OCCURS_IN
bc_CONVERTS_TO	bc_NEG_LOCATION_OF	bc_TREATS	cb_METHOD_OF	cb_PART_OF
bc_DIAGNOSES	bc_NEG_OCCURS_IN	bc_USES	cb_NEG_ADMINISTERED_TO	cb_PRECEDES
bc_DISRUPTS	bc_NEG_PART_OF	cb_ADMINISTERED_TO	cb_NEG_AFFECTS	cb_PREDISPOSES
bc_INHIBITS	bc_NEG_PREDISPOSES	cb_AFFECTS	cb_NEG_ASSOCIATED_WITH	cb_PREVENTS
bc_INTERACTS_WITH	bc_NEG_PREVENTS	cb_ASSOCIATED_WITH	cb_NEG_AUGMENTS	cb_PROCESS_OF
bc_ISA	bc_NEG_PROCESS_OF	cb_AUGMENTS	cb_NEG_COEXISTS_WITH	cb_PRODUCES
bc_LOCATION_OF	bc_NEG_PRODUCES	cb_CAUSES	cb_NEG_COMPLICATES	cb_STIMULATES
bc_MANIFESTATION_OF	bc_NEG_STIMULATES	cb_COEXISTS_WITH	cb_NEG_DISRUPTS	cb_TREATS
bc_METHOD_OF	bc_NEG_TREATS	cb_compared_with	cb_NEG_INHIBITS	cb_USES
bc_NEG_ADMINISTERED_TO	bc_NEG_USES	cb_COMPLICATES	cb_NEG_INTERACTS_WITH	

Table 3.4 List of Feature Values for B to C and C to B Relationships (89 total)

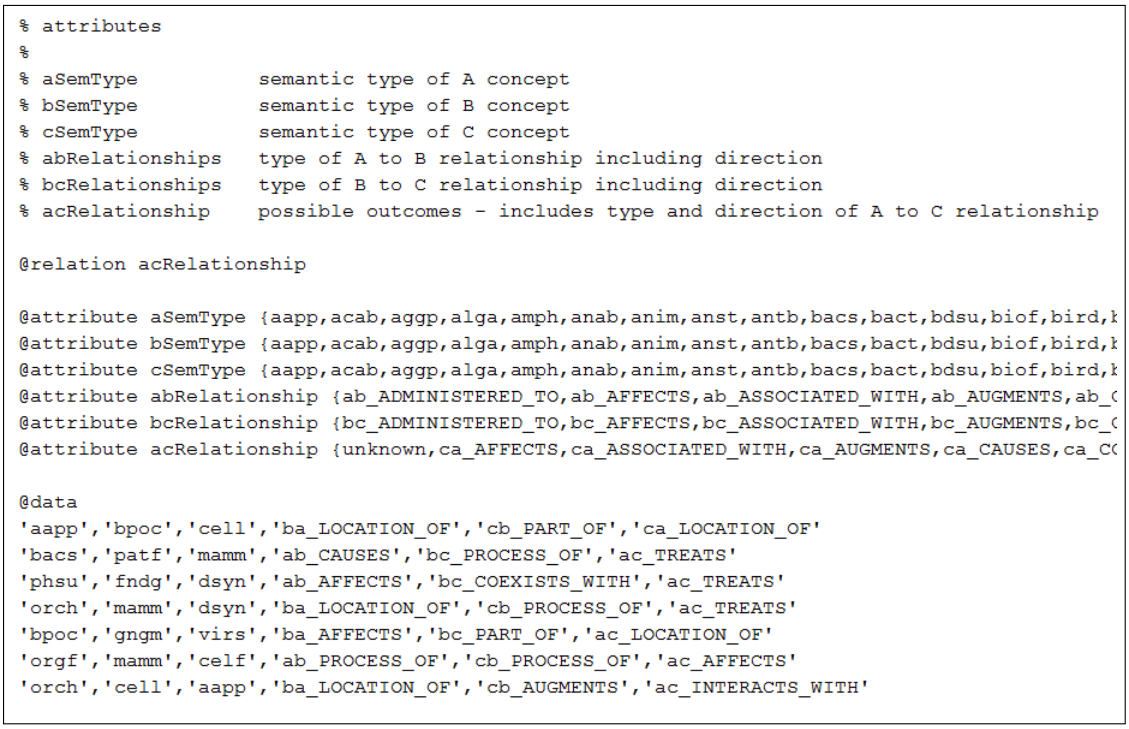


ac_ADMINISTERED_TO	ac_NEG_AUGMENTS	ac_PART_OF	ca_INHIBITS	ca_NEG_PART_OF
ac_AFFECTS	ac_NEG_CAUSES	ac_PRECEDES	ca_INTERACTS_WITH	ca_NEG_PREDISPOSES
ac_ASSOCIATED_WITH	ac_NEG_COEXISTS_WITH	ac_PREDISPOSES	ca_ISA	ca_NEG_PREVENTS
ac_AUGMENTS	ac_NEG_CONVERTS_TO	ac_PREVENTS	ca_LOCATION_OF	ca_NEG_PROCESS_OF
ac_CAUSES	ac_NEG_DIAGNOSES	ac_PROCESS_OF	ca_lower_than	ca_NEG_PRODUCES
ac_COEXISTS_WITH	ac_NEG_DISRUPTS	ac_PRODUCES	ca_MANIFESTATION_OF	ca_NEG_STIMULATES
ac_compared_with	ac_NEG_higher_than	ac_same_as	ca_METHOD_OF	ca_NEG_TREATS
ac_COMPLICATES	ac_NEG_INHIBITS	ac_STIMULATES	ca_NEG_ADMINISTERED_TO	ca_NEG_USES
ac_CONVERTS_TO	ac_NEG_INTERACTS_WITH	ac_TREATS	ca_NEG_AFFECTS	ca_OCCURS_IN
ac_DIAGNOSES	ac_NEG_LOCATION_OF	ac_USES	ca_NEG_ASSOCIATED_WITH	ca_PART_OF
ac_DISRUPTS	ac_NEG_MANIFESTATION_OF	ca_ADMINISTERED_TO	ca_NEG_AUGMENTS	ca_PRECEDES
ac_higher_than	ac_NEG_METHOD_OF	ca_AFFECTS	ca_NEG_CAUSES	ca_PREDISPOSES
ac_INHIBITS	ac_NEG_OCCURS_IN	ca_ASSOCIATED_WITH	ca_NEG_COEXISTS_WITH	ca_PREVENTS
ac_INTERACTS_WITH	ac_NEG_PART_OF	ca_AUGMENTS	ca_NEG_COMPLICATES	ca_PROCESS_OF
ac_ISA	ac_NEG_PREDISPOSES	ca_CAUSES	ca_NEG_DIAGNOSES	ca_PRODUCES
ac_LOCATION_OF	ac_NEG_PREVENTS	ca_COEXISTS_WITH	ca_NEG_DISRUPTS	ca_same_as
ac_lower_than	ac_NEG_PROCESS_OF	ca_compared_with	ca_NEG_higher_than	ca_STIMULATES
ac_MANIFESTATION_OF	ac_NEG_PRODUCES	ca_COMPLICATES	ca_NEG_INHIBITS	ca_TREATS
ac_METHOD_OF	ac_NEG_STIMULATES	ca_CONVERTS_TO	ca_NEG_INTERACTS_WITH	ca_USES
ac_NEG_ADMINISTERED_TO	ac_NEG_TREATS	ca_DIAGNOSES	ca_NEG_LOCATION_OF	
ac_NEG_AFFECTS	ac_NEG_USES	ca_DISRUPTS	ca_NEG_MANIFESTATION_OF	
ac_NEG_ASSOCIATED_WITH	ac_OCCURS_IN	ca_higher_than	ca_NEG_OCCURS_IN	

Table 3.5 List of Outcomes for C to A and A to C Relationship Outcomes (107 total)

3.4 Data Representation for Classifier Training

Weka (Hall et al., 2009) is used in this work to perform the classification and provides data to generate statistics and other evaluation data. This implementation detail is presented here, because the file format found in Weka is used in this work to capture the data used for classification. Weka uses an ASCII text file data representation called Attribute-Relation File Format or ARFF. The ARFF file contains in its header the definition of all classes including valid feature values of each. The rest of an ARFF file contains either the fully qualified training data or the data on which classification will be performed. In the latter, a question mark (?) is used to represent the unknown outcome in the data section which, in this work, is the A-C relationship and direction. An abbreviated example of ARFF data extracted from experimental files is shown in Figure 3.4. Since, ultimately, the results from classification need to include the disease names, a parallel reference map is maintained that relates each line in training or LBD ARFF file representations back to a human readable form. That is, for example, if the semantic types of *tisu* (abbreviation for Tissue), *aapp* (Amino Acid, Peptide, or Protein) and *orch* (Organic Chemical) appear in the training data for the A, B and C semantic types, respectively, the human readable information about that row are also maintained and might be, for example, "Tissue membrane", "Bleomycin" and Mitoxantrone".



```
% attributes
%
% aSemType          semantic type of A concept
% bSemType          semantic type of B concept
% cSemType          semantic type of C concept
% abRelationships   type of A to B relationship including direction
% bcRelationships   type of B to C relationship including direction
% acRelationship    possible outcomes - includes type and direction of A to C relationship

@relation acRelationship

@attribute aSemType {aapp,acab,aggp,alga,amph,anab,anim,anst,antb,bacs,bact,bdsu,biof,bird,
@attribute bSemType {aapp,acab,aggp,alga,amph,anab,anim,anst,antb,bacs,bact,bdsu,biof,bird,
@attribute cSemType {aapp,acab,aggp,alga,amph,anab,anim,anst,antb,bacs,bact,bdsu,biof,bird,
@attribute abRelationship {ab_ADMINISTERED_TO,ab_AFFECTS,ab_ASSOCIATED_WITH,ab_AUGMENTS,ab_
@attribute bcRelationship {bc_ADMINISTERED_TO,bc_AFFECTS,bc_ASSOCIATED_WITH,bc_AUGMENTS,bc_
@attribute acRelationship {unknown,ca_AFFECTS,ca_ASSOCIATED_WITH,ca_AUGMENTS,ca_CAUSES,ca_C

@data
'aapp','bpoc','cell','ba_LOCATION_OF','cb_PART_OF','ca_LOCATION_OF'
'bacs','patf','mamm','ab_CAUSES','bc_PROCESS_OF','ac_TREATS'
'phsu','fndg','dsyn','ab_AFFECTS','bc_COEXISTS_WITH','ac_TREATS'
'orch','mamm','dsyn','ba_LOCATION_OF','cb_PROCESS_OF','ac_TREATS'
'bpoc','gngm','virs','ba_AFFECTS','bc_PART_OF','ac_LOCATION_OF'
'orgf','mamm','celf','ab_PROCESS_OF','cb_PROCESS_OF','ac_AFFECTS'
'orch','cell','aapp','ba_LOCATION_OF','cb_AUGMENTS','ac_INTERACTS_WITH'
```

Fig. 3.4 Example of ARFF file used to train classifiers (the @attribute lines have been truncated for readability)

3.5 Data Variations for Experiments

The previous sections, in this chapter, described how the LBD and linking data can be represented and presented to classifiers. The data described basic features and outcomes that classifiers will be able to process with the goal of predicting the outcomes using classification. In this research, various data representations were explored. They will be explained in this section to document details of these data designs. The purpose of presenting the details of these variations here, rather than with the experimentation, is to keep the details of these variations separated from the discussion of the experiments and their results. These data descriptions may be thought of as reference materials for the experiments to be discussed in Section 4.3.

This section presents two variations of the data that will be used to try to improve performance of classifiers in explaining the LBD relationship. The first variation removes instances containing very commonly occurring feature values (experiment presented in Section 4.3.2). The thought is that these instances may introduce noise. The second variation reduces the number of outcomes to predict by only using the A to C direction and converting all of the C to A training data feature values into the A to C direction (experiment presented in Section 4.3.3). This results in a re-ordering of some of the triples to regularize the outcomes.

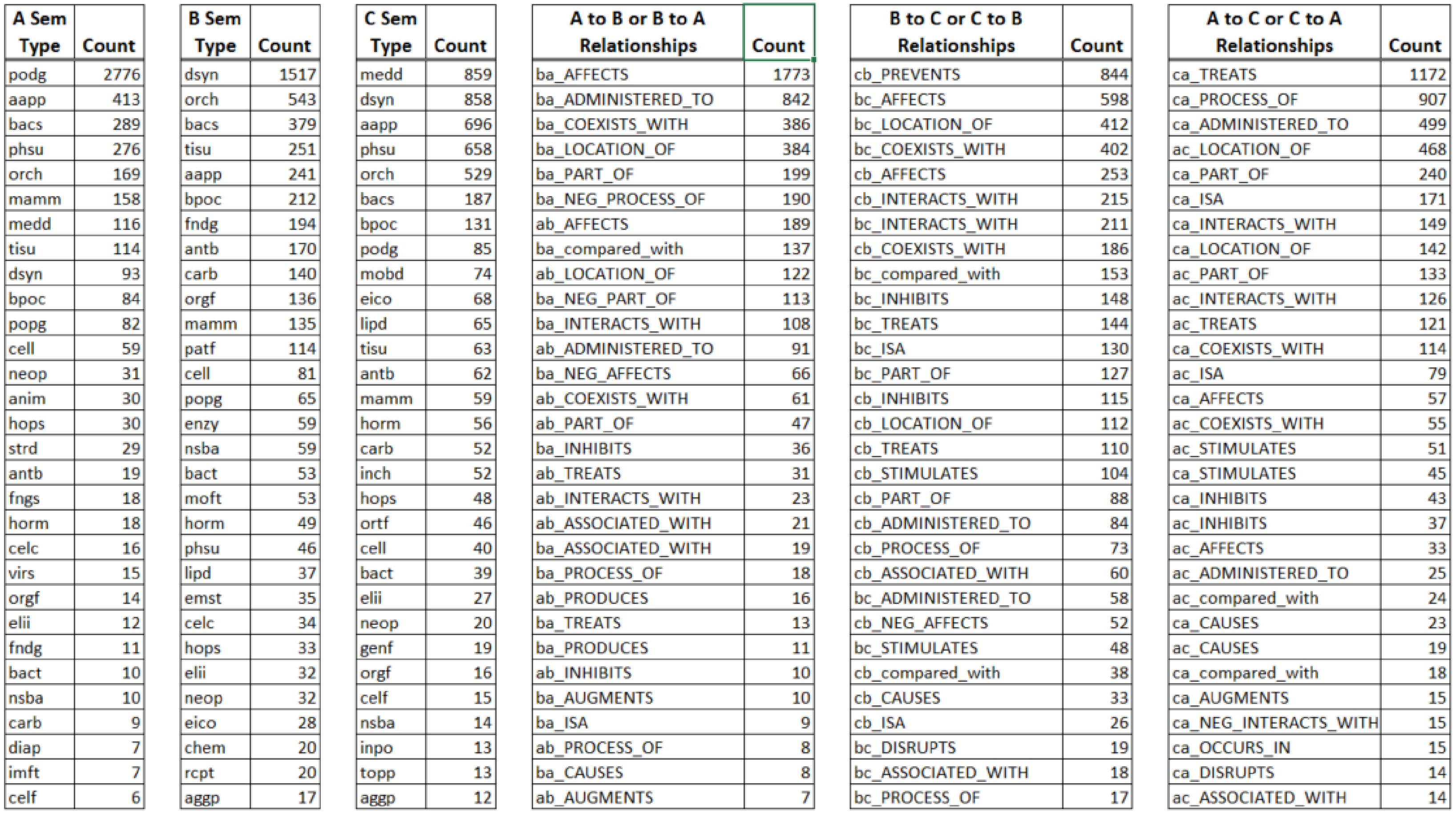
3.5.1 Removal of Very Common Feature Values

After initial experiments were performed, the data used in those initial trials were examined and it was noticed that one feature value occurred much more frequently in the data than all others. An experiment was performed that studied the affect of removing commonly occurring classes (see Section 4.3.2). This section describes the details of the data used in that experiment. Tables 3.6 and 3.7 show the distribution, from most frequent to least, of the top 30 A, B, and C semantic types and A to B, B to C and C to A relationships. In both the 1980-1984 and the post-1984 data, *podg* (which translates to "Patient or Disabled Group") occurs much more than any of the other A semantic types – *podg* appears 2,776 times in the 1980-1984 data set with the next highest occurring semantic type, *aapp*, appearing 413 times (less than one sixth of the *podg* occurrences), and, in the post-1984 data set, 84,210 occurrences of *podg* vs. 21,784 of *aapp* (approximately one quarter of the *podg* occurrences).

Examples of the *podg* semantic type include "Patients", "Outpatients" and "Inpatients". Including the *podg* semantic type in the training data may be introducing noise, because knowing that some type of patient, for example, is related to some other semantic type does not provide useful information in discovering cures to diseases or other more compelling discoveries. Removing *podg* makes sense from a pure theoretical point of view with respect to classification training. However, others may argue that the *podg* semantic type is important in the medical domain, so may not be wise to omit it in classification results. This will be discussed more in the actual experiment presented in Section 4.3.2.

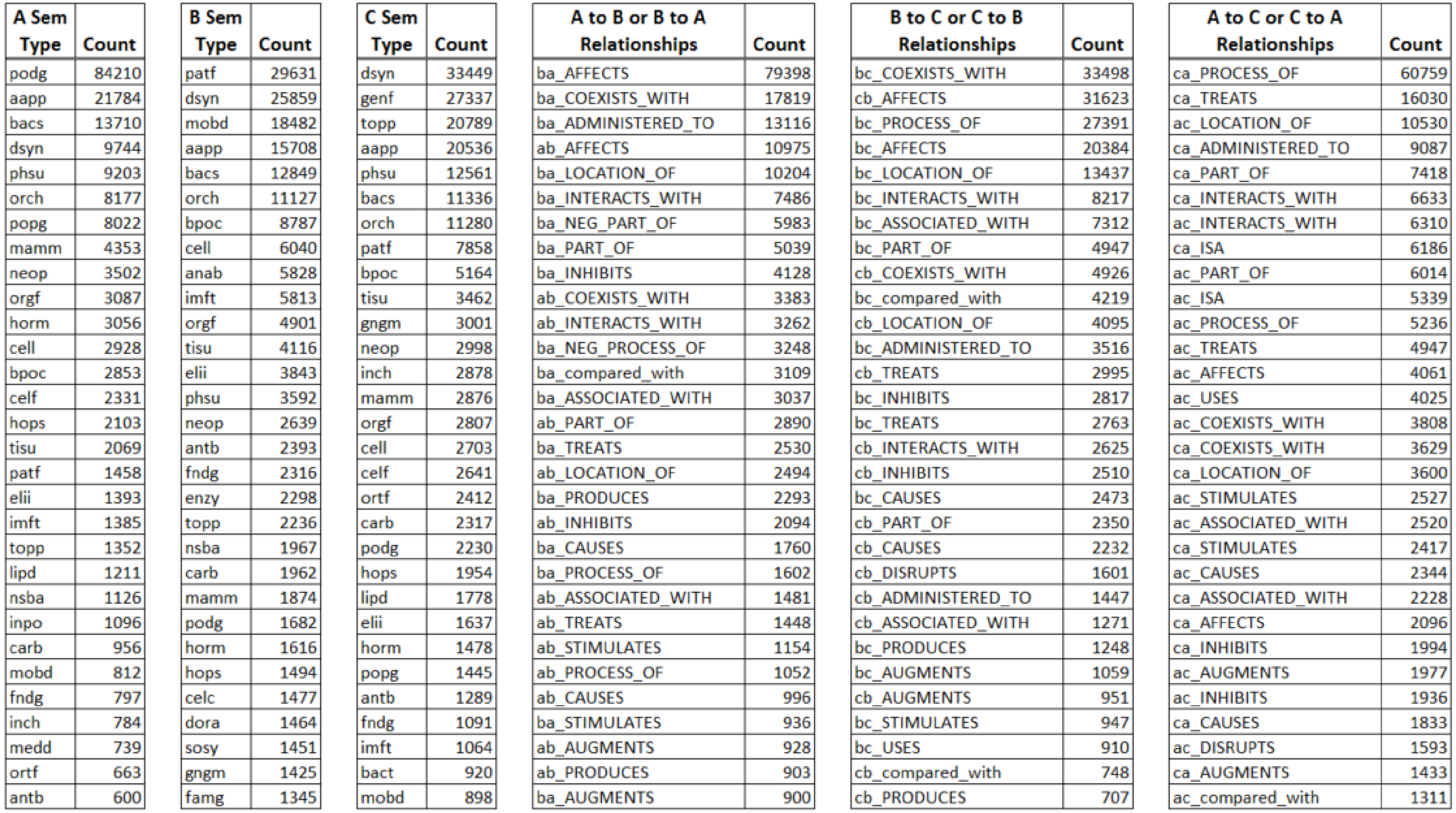
Tables showing the distributions of the remaining semantic types after removal training sets with *podg* in them are shown in Tables 3.8 and 3.9. Removing *podg* references mean that entire instances (fully qualified triples) are removed from the training data. After removal, the 1980-1984 training set was reduced from 4,997 instances of training data down to 2,135 and for the post-1984 set, 203,323 down to 116,624 instances.

In the original training data, the relationships *ba_AFFECTS* shows up a disproportionately large number of times in both the 1980-1984 and the post-1984 data. The relationships *ca_PROCESS_OF* shows up a very large number of times in the post-1984 data. After removing the *podg* semantic type from the training data, neither of these relationships remained at the top of the list. Therefore, remaining occurrences of these two relationships were preserved and not removed.



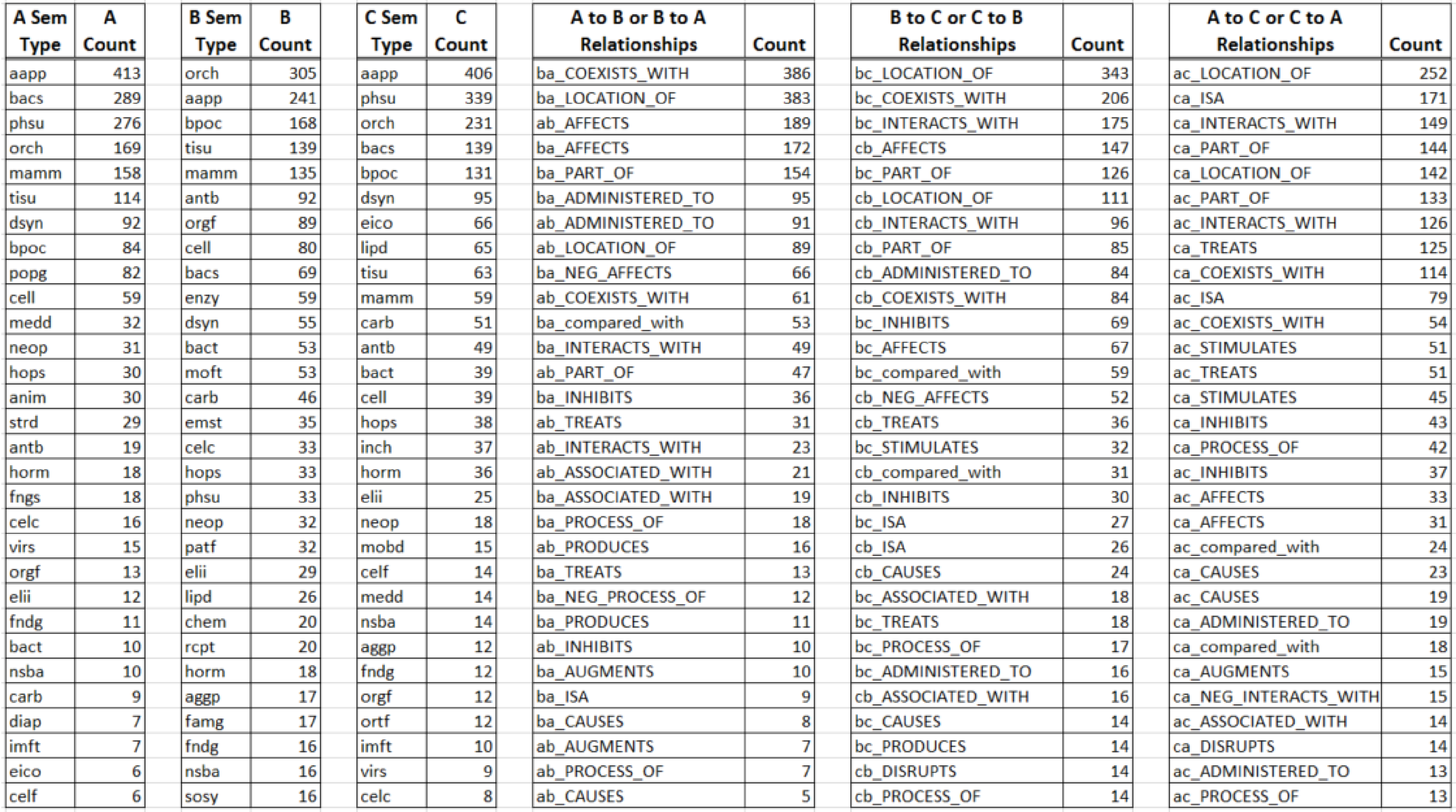
A Sem		B Sem		C Sem		A to B or B to A		B to C or C to B		A to C or C to A	
Type	Count	Type	Count	Type	Count	Relationships	Count	Relationships	Count	Relationships	Count
podg	2776	dsyn	1517	medd	859	ba_AFFECTS	1773	cb_PREVENTS	844	ca_TREATS	1172
aapp	413	orch	543	dsyn	858	ba_ADMINISTERED_TO	842	bc_AFFECTS	598	ca_PROCESS_OF	907
bacs	289	bacs	379	aapp	696	ba_COEXISTS_WITH	386	bc_LOCATION_OF	412	ca_ADMINISTERED_TO	499
phsu	276	tisu	251	phsu	658	ba_LOCATION_OF	384	bc_COEXISTS_WITH	402	ac_LOCATION_OF	468
orch	169	aapp	241	orch	529	ba_PART_OF	199	cb_AFFECTS	253	ca_PART_OF	240
mamm	158	bpoc	212	bacs	187	ba_NEG_PROCESS_OF	190	cb_INTERACTS_WITH	215	ca_ISA	171
medd	116	fndg	194	bpoc	131	ab_AFFECTS	189	bc_INTERACTS_WITH	211	ca_INTERACTS_WITH	149
tisu	114	antb	170	podg	85	ba_compared_with	137	cb_COEXISTS_WITH	186	ca_LOCATION_OF	142
dsyn	93	carb	140	mobd	74	ab_LOCATION_OF	122	bc_compared_with	153	ac_PART_OF	133
bpoc	84	orgf	136	eico	68	ba_NEG_PART_OF	113	bc_INHIBITS	148	ac_INTERACTS_WITH	126
popg	82	mamm	135	lipd	65	ba_INTERACTS_WITH	108	bc_TREATS	144	ac_TREATS	121
cell	59	patf	114	tisu	63	ab_ADMINISTERED_TO	91	bc_ISA	130	ca_COEXISTS_WITH	114
neop	31	cell	81	antb	62	ba_NEG_AFFECTS	66	bc_PART_OF	127	ac_ISA	79
anim	30	popg	65	mamm	59	ab_COEXISTS_WITH	61	cb_INHIBITS	115	ca_AFFECTS	57
hops	30	enzy	59	horm	56	ab_PART_OF	47	cb_LOCATION_OF	112	ac_COEXISTS_WITH	55
strd	29	nsba	59	carb	52	ba_INHIBITS	36	cb_TREATS	110	ac_STIMULATES	51
antb	19	bact	53	inch	52	ab_TREATS	31	cb_STIMULATES	104	ca_STIMULATES	45
fngs	18	moft	53	hops	48	ab_INTERACTS_WITH	23	cb_PART_OF	88	ca_INHIBITS	43
horm	18	horm	49	ortf	46	ab_ASSOCIATED_WITH	21	cb_ADMINISTERED_TO	84	ac_INHIBITS	37
celc	16	phsu	46	cell	40	ba_ASSOCIATED_WITH	19	cb_PROCESS_OF	73	ac_AFFECTS	33
virs	15	lipd	37	bact	39	ba_PROCESS_OF	18	cb_ASSOCIATED_WITH	60	ac_ADMINISTERED_TO	25
orgf	14	emst	35	elii	27	ab_PRODUCES	16	bc_ADMINISTERED_TO	58	ac_compared_with	24
elii	12	celc	34	neop	20	ba_TREATS	13	cb_NEG_AFFECTS	52	ca_CAUSES	23
fndg	11	hops	33	genf	19	ba_PRODUCES	11	bc_STIMULATES	48	ac_CAUSES	19
bact	10	elii	32	orgf	16	ab_INHIBITS	10	cb_compared_with	38	ca_compared_with	18
nsba	10	neop	32	celf	15	ba_AUGMENTS	10	cb_CAUSES	33	ca_AUGMENTS	15
carb	9	eico	28	nsba	14	ba_ISA	9	cb_ISA	26	ca_NEG_INTERACTS_WITH	15
diap	7	chem	20	inpo	13	ab_PROCESS_OF	8	bc_DISRUPTS	19	ca_OCCURS_IN	15
imft	7	rcpt	20	topp	13	ba_CAUSES	8	bc_ASSOCIATED_WITH	18	ca_DISRUPTS	14
celf	6	aggp	17	aggp	12	ab_AUGMENTS	7	bc_PROCESS_OF	17	ac_ASSOCIATED_WITH	14

Table 3.6 Distributions of Original 1980-1984 Semantic Types and Relationships (notice A Sem Type with 2776 *podg* occurrences)



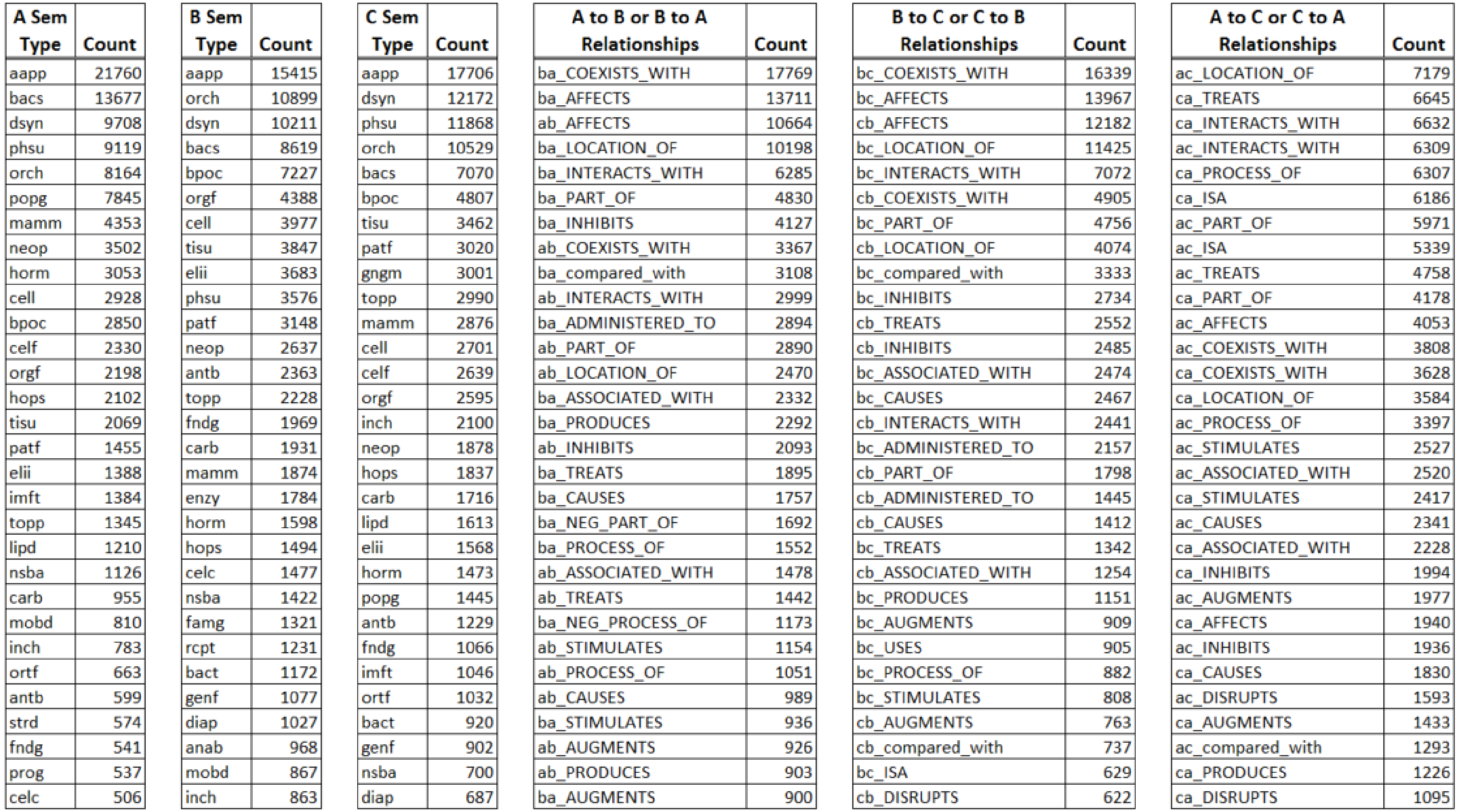
A Sem Type	Count	B Sem Type	Count	C Sem Type	Count	A to B or B to A Relationships	Count	B to C or C to B Relationships	Count	A to C or C to A Relationships	Count
podg	84210	patf	29631	dsyn	33449	ba_AFFECTS	79398	bc_COEXISTS_WITH	33498	ca_PROCESS_OF	60759
aapp	21784	dsyn	25859	genf	27337	ba_COEXISTS_WITH	17819	cb_AFFECTS	31623	ca_TREATS	16030
bacs	13710	mobd	18482	topp	20789	ba_ADMINISTERED_TO	13116	bc_PROCESS_OF	27391	ac_LOCATION_OF	10530
dsyn	9744	aapp	15708	aapp	20536	ab_AFFECTS	10975	bc_AFFECTS	20384	ca_ADMINISTERED_TO	9087
phsu	9203	bacs	12849	phsu	12561	ba_LOCATION_OF	10204	bc_LOCATION_OF	13437	ca_PART_OF	7418
orch	8177	orch	11127	bacs	11336	ba_INTERACTS_WITH	7486	bc_INTERACTS_WITH	8217	ca_INTERACTS_WITH	6633
popg	8022	bpoc	8787	orch	11280	ba_NEG_PART_OF	5983	bc_ASSOCIATED_WITH	7312	ac_INTERACTS_WITH	6310
mamm	4353	cell	6040	patf	7858	ba_PART_OF	5039	bc_PART_OF	4947	ca_ISA	6186
neop	3502	anab	5828	bpoc	5164	ba_INHIBITS	4128	cb_COEXISTS_WITH	4926	ac_PART_OF	6014
orgf	3087	imft	5813	tisu	3462	ab_COEXISTS_WITH	3383	bc_compared_with	4219	ac_ISA	5339
horm	3056	orgf	4901	gngm	3001	ab_INTERACTS_WITH	3262	cb_LOCATION_OF	4095	ac_PROCESS_OF	5236
cell	2928	tisu	4116	neop	2998	ba_NEG_PROCESS_OF	3248	bc_ADMINISTERED_TO	3516	ac_TREATS	4947
bpoc	2853	elii	3843	inch	2878	ba_compared_with	3109	cb_TREATS	2995	ac_AFFECTS	4061
celf	2331	phsu	3592	mamm	2876	ba_ASSOCIATED_WITH	3037	bc_INHIBITS	2817	ac_USES	4025
hops	2103	neop	2639	orgf	2807	ab_PART_OF	2890	bc_TREATS	2763	ac_COEXISTS_WITH	3808
tisu	2069	antb	2393	cell	2703	ba_TREATS	2530	cb_INTERACTS_WITH	2625	ca_COEXISTS_WITH	3629
patf	1458	fndg	2316	celf	2641	ab_LOCATION_OF	2494	cb_INHIBITS	2510	ca_LOCATION_OF	3600
elii	1393	enzy	2298	ortf	2412	ba_PRODUCES	2293	bc_CAUSES	2473	ac_STIMULATES	2527
imft	1385	topp	2236	carb	2317	ab_INHIBITS	2094	cb_PART_OF	2350	ac_ASSOCIATED_WITH	2520
topp	1352	nsba	1967	podg	2230	ba_CAUSES	1760	cb_CAUSES	2232	ca_STIMULATES	2417
lipd	1211	carb	1962	hops	1954	ba_PROCESS_OF	1602	cb_DISRUPTS	1601	ac_CAUSES	2344
nsba	1126	mamm	1874	lipd	1778	ab_ASSOCIATED_WITH	1481	cb_ADMINISTERED_TO	1447	ca_ASSOCIATED_WITH	2228
inpo	1096	podg	1682	elii	1637	ab_TREATS	1448	cb_ASSOCIATED_WITH	1271	ca_AFFECTS	2096
carb	956	horm	1616	horm	1478	ab_STIMULATES	1154	bc_PRODUCES	1248	ca_INHIBITS	1994
mobd	812	hops	1494	popg	1445	ab_PROCESS_OF	1052	bc_AUGMENTS	1059	ac_AUGMENTS	1977
fndg	797	celc	1477	antb	1289	ab_CAUSES	996	cb_AUGMENTS	951	ac_INHIBITS	1936
inch	784	dora	1464	fndg	1091	ba_STIMULATES	936	bc_STIMULATES	947	ca_CAUSES	1833
medd	739	sosy	1451	imft	1064	ab_AUGMENTS	928	bc_USES	910	ac_DISRUPTS	1593
ortf	663	gngm	1425	bact	920	ab_PRODUCES	903	cb_compared_with	748	ca_AUGMENTS	1433
antb	600	famg	1345	mobd	898	ba_AUGMENTS	900	cb_PRODUCES	707	ac_compared_with	1311

Table 3.7 Distributions of Original Post-1984 Semantic Types and Relationships (notice A Sem Type with 84210 *podg* occurrences)



A Sem Type	A Count	B Sem Type	B Count	C Sem Type	C Count	A to B or B to A Relationships	Count	B to C or C to B Relationships	Count	A to C or C to A Relationships	Count
aapp	413	orch	305	aapp	406	ba_COEXISTS_WITH	386	bc_LOCATION_OF	343	ac_LOCATION_OF	252
bacs	289	aapp	241	phsu	339	ba_LOCATION_OF	383	bc_COEXISTS_WITH	206	ca_ISA	171
phsu	276	bpoc	168	orch	231	ab_AFFECTS	189	bc_INTERACTS_WITH	175	ca_INTERACTS_WITH	149
orch	169	tisu	139	bacs	139	ba_AFFECTS	172	cb_AFFECTS	147	ca_PART_OF	144
mamm	158	mamm	135	bpoc	131	ba_PART_OF	154	bc_PART_OF	126	ca_LOCATION_OF	142
tisu	114	antb	92	dsyn	95	ba_ADMINISTERED_TO	95	cb_LOCATION_OF	111	ac_PART_OF	133
dsyn	92	orgf	89	eico	66	ab_ADMINISTERED_TO	91	cb_INTERACTS_WITH	96	ac_INTERACTS_WITH	126
bpoc	84	cell	80	lipd	65	ab_LOCATION_OF	89	cb_PART_OF	85	ca_TREATS	125
popg	82	bacs	69	tisu	63	ba_NEG_AFFECTS	66	cb_ADMINISTERED_TO	84	ca_COEXISTS_WITH	114
cell	59	enzy	59	mamm	59	ab_COEXISTS_WITH	61	cb_COEXISTS_WITH	84	ac_ISA	79
medd	32	dsyn	55	carb	51	ba_compared_with	53	bc_INHIBITS	69	ac_COEXISTS_WITH	54
neop	31	bact	53	antb	49	ba_INTERACTS_WITH	49	bc_AFFECTS	67	ac_STIMULATES	51
hops	30	moft	53	bact	39	ab_PART_OF	47	bc_compared_with	59	ac_TREATS	51
anim	30	carb	46	cell	39	ba_INHIBITS	36	cb_NEG_AFFECTS	52	ca_STIMULATES	45
strd	29	emst	35	hops	38	ab_TREATS	31	cb_TREATS	36	ca_INHIBITS	43
antb	19	celc	33	inch	37	ab_INTERACTS_WITH	23	bc_STIMULATES	32	ca_PROCESS_OF	42
horm	18	hops	33	horm	36	ab_ASSOCIATED_WITH	21	cb_compared_with	31	ac_INHIBITS	37
fngs	18	phsu	33	elii	25	ba_ASSOCIATED_WITH	19	cb_INHIBITS	30	ac_AFFECTS	33
celc	16	neop	32	neop	18	ba_PROCESS_OF	18	bc_ISA	27	ca_AFFECTS	31
virs	15	patf	32	mobd	15	ab_PRODUCES	16	cb_ISA	26	ac_compared_with	24
orgf	13	elii	29	celf	14	ba_TREATS	13	cb_CAUSES	24	ca_CAUSES	23
elii	12	lipd	26	medd	14	ba_NEG_PROCESS_OF	12	bc_ASSOCIATED_WITH	18	ac_CAUSES	19
fndg	11	chem	20	nsba	14	ba_PRODUCES	11	bc_TREATS	18	ca_ADMINISTERED_TO	19
bact	10	rcpt	20	aggp	12	ab_INHIBITS	10	bc_PROCESS_OF	17	ca_compared_with	18
nsba	10	horm	18	fndg	12	ba_AUGMENTS	10	bc_ADMINISTERED_TO	16	ca_AUGMENTS	15
carb	9	aggp	17	orgf	12	ba_ISA	9	cb_ASSOCIATED_WITH	16	ca_NEG_INTERACTS_WITH	15
diap	7	famg	17	ortf	12	ba_CAUSES	8	bc_CAUSES	14	ac_ASSOCIATED_WITH	14
imft	7	fndg	16	imft	10	ab_AUGMENTS	7	bc_PRODUCES	14	ca_DISRUPTS	14
eico	6	nsba	16	virs	9	ab_PROCESS_OF	7	cb_DISRUPTS	14	ac_ADMINISTERED_TO	13
celf	6	sosy	16	celc	8	ab_CAUSES	5	cb_PROCESS_OF	14	ac_PROCESS_OF	13

Table 3.8 Distributions After PODG Semantic Type Removal from 1980-1984 Data



A Sem Type	Count	B Sem Type	Count	C Sem Type	Count	A to B or B to A Relationships	Count	B to C or C to B Relationships	Count	A to C or C to A Relationships	Count
aapp	21760	aapp	15415	aapp	17706	ba_COEXISTS_WITH	17769	bc_COEXISTS_WITH	16339	ac_LOCATION_OF	7179
bacs	13677	orch	10899	dsyn	12172	ba_AFFECTS	13711	bc_AFFECTS	13967	ca_TREATS	6645
dsyn	9708	dsyn	10211	phsu	11868	ab_AFFECTS	10664	cb_AFFECTS	12182	ca_INTERACTS_WITH	6632
phsu	9119	bacs	8619	orch	10529	ba_LOCATION_OF	10198	bc_LOCATION_OF	11425	ac_INTERACTS_WITH	6309
orch	8164	bpoc	7227	bacs	7070	ba_INTERACTS_WITH	6285	bc_INTERACTS_WITH	7072	ca_PROCESS_OF	6307
popg	7845	orgf	4388	bpoc	4807	ba_PART_OF	4830	cb_COEXISTS_WITH	4905	ca_ISA	6186
mamm	4353	cell	3977	tisu	3462	ba_INHIBITS	4127	bc_PART_OF	4756	ac_PART_OF	5971
neop	3502	tisu	3847	patf	3020	ab_COEXISTS_WITH	3367	cb_LOCATION_OF	4074	ac_ISA	5339
horm	3053	elii	3683	gngm	3001	ba_compared_with	3108	bc_compared_with	3333	ac_TREATS	4758
cell	2928	phsu	3576	topp	2990	ab_INTERACTS_WITH	2999	bc_INHIBITS	2734	ca_PART_OF	4178
bpoc	2850	patf	3148	mamm	2876	ba_ADMINISTERED_TO	2894	cb_TREATS	2552	ac_AFFECTS	4053
celf	2330	neop	2637	cell	2701	ab_PART_OF	2890	cb_INHIBITS	2485	ac_COEXISTS_WITH	3808
orgf	2198	antb	2363	celf	2639	ab_LOCATION_OF	2470	bc_ASSOCIATED_WITH	2474	ca_COEXISTS_WITH	3628
hops	2102	topp	2228	orgf	2595	ba_ASSOCIATED_WITH	2332	bc_CAUSES	2467	ca_LOCATION_OF	3584
tisu	2069	fndg	1969	inch	2100	ba_PRODUCES	2292	cb_INTERACTS_WITH	2441	ac_PROCESS_OF	3397
patf	1455	carb	1931	neop	1878	ab_INHIBITS	2093	bc_ADMINISTERED_TO	2157	ac_STIMULATES	2527
elii	1388	mamm	1874	hops	1837	ba_TREATS	1895	cb_PART_OF	1798	ac_ASSOCIATED_WITH	2520
imft	1384	enzy	1784	carb	1716	ba_CAUSES	1757	cb_ADMINISTERED_TO	1445	ca_STIMULATES	2417
topp	1345	horm	1598	lipd	1613	ba_NEG_PART_OF	1692	cb_CAUSES	1412	ac_CAUSES	2341
lipd	1210	hops	1494	elii	1568	ba_PROCESS_OF	1552	bc_TREATS	1342	ca_ASSOCIATED_WITH	2228
nsba	1126	celc	1477	horm	1473	ab_ASSOCIATED_WITH	1478	cb_ASSOCIATED_WITH	1254	ca_INHIBITS	1994
carb	955	nsba	1422	popg	1445	ab_TREATS	1442	bc_PRODUCES	1151	ac_AUGMENTS	1977
mobd	810	famg	1321	antb	1229	ba_NEG_PROCESS_OF	1173	bc_AUGMENTS	909	ca_AFFECTS	1940
inch	783	rcpt	1231	fndg	1066	ab_STIMULATES	1154	bc_USES	905	ac_INHIBITS	1936
ortf	663	bact	1172	imft	1046	ab_PROCESS_OF	1051	bc_PROCESS_OF	882	ca_CAUSES	1830
antb	599	genf	1077	ortf	1032	ab_CAUSES	989	bc_STIMULATES	808	ac_DISRUPTS	1593
strd	574	diap	1027	bact	920	ba_STIMULATES	936	cb_AUGMENTS	763	ca_AUGMENTS	1433
fndg	541	anab	968	genf	902	ab_AUGMENTS	926	cb_compared_with	737	ac_compared_with	1293
prog	537	mobd	867	nsba	700	ab_PRODUCES	903	bc_ISA	629	ca_PRODUCES	1226
celc	506	inch	863	diap	687	ba_AUGMENTS	900	cb_DISRUPTS	622	ca_DISRUPTS	1095

Table 3.9 Distributions After PODG Semantic Type Removal from Post-1984 Data

After *podg* was removed from the training data, some feature instances no longer occurred in the data. Semantic types, that no longer occur after removing training data instances that included the *podg* semantic type, are *amph* (amphibian), *anst* (anatomical structure), *clnd* (clinical drug), *edac* (educational activity), *ftcn* (functional concept), *inbe* (individual behavior), *nusq* (nucleotide sequence), *ocac* (occupational activity), and *spco* (spatial concept). Clinical drugs and nucleotide sequence may be important when trying to identify treatments or causes of diseases but the other semantic types are noise similar to *podg*. The unique semantic types found in the training data dropped from 90 to 88 for the A semantic type, 85 to 84 for B, and 88 to 87 for C. The relationships between A to B dropped from 88 to 87, did not change for B to C and dropped from 107 to 106 for C to A relationships.

3.5.2 Regularization of Outcomes - Convert to AC Direction only

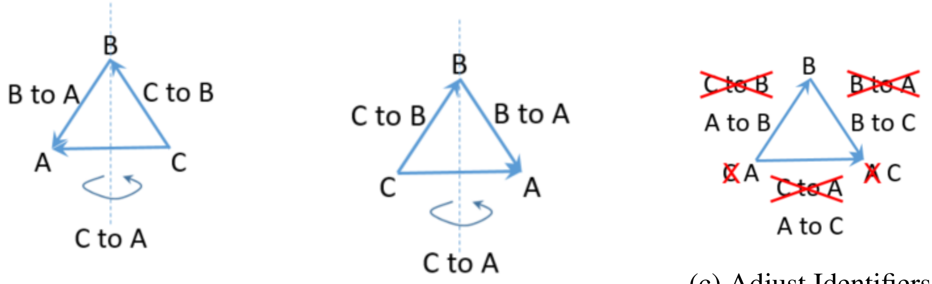
Another of the classification experiments reduces the number of possible outcomes to reduce the inherent redundancy by turning all triples into relationships with only the A to C direction (the actual experiment is presented in Section 4.3.3). Some other explorations into other representations are described in the net two paragraphs. These were performed before settling on the data representation with the A to C direction only.

One of these explorations was to simplify outcomes split the training data into two sets – one with only the A to C relationships, and the other with only C to A relationships. That is, the original experiment design includes both A to C and C to A relationships in the training data. This produced 110 unique A to C and C to A relationships. When only A to C are considered in one training set and C to A in another separate training set, 55 outcomes for each remained. After *podg* is removed, 54 and 53, respectively remain in the separated A to C and the C to A relationship sets.

Additional exploration involved separation of the A to B relationships from the B to A relationships and to separate B to C from C to B directed relationships. This created a total of 7 features – the three A, B and C Semantic Types along with the A to B relationships, B to A relationships, B to C relationships, C to B relationships.

Neither one of these explorations produced results worth considering further. They are mentioned only as a reference of other data designs that were considered.

Outcome regularization used in this work involved converting the training data so that only the A to C direction is being predicted. That is, the direction of the A to C relationship was set to always be in the direction from A to C and the other supporting data was flipped



(a) Triple with a C to A Relationship

(b) Flip Triple Around B

(c) Adjust Identifiers

Fig. 3.5 Visualization of Triple Flipping

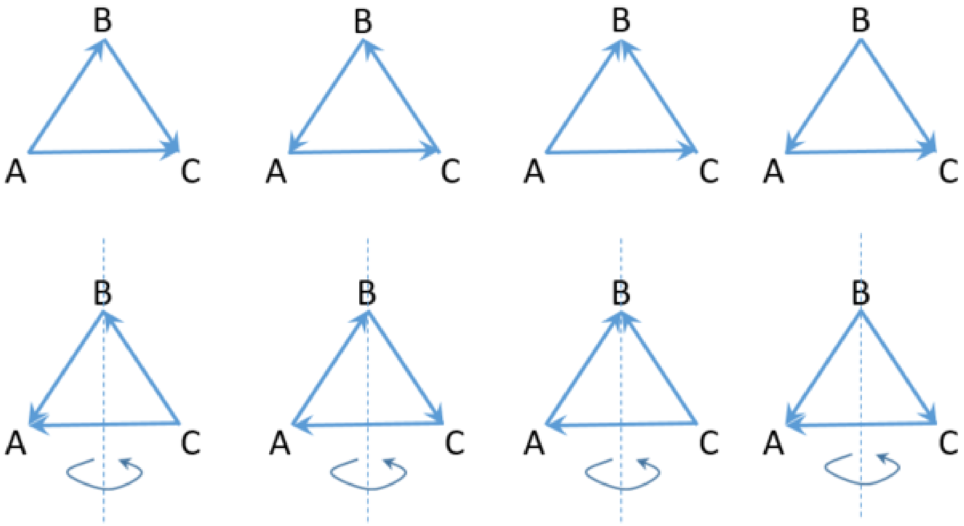


Fig. 3.6 Permutations of Triples and Indication of C to A Directions to Flip

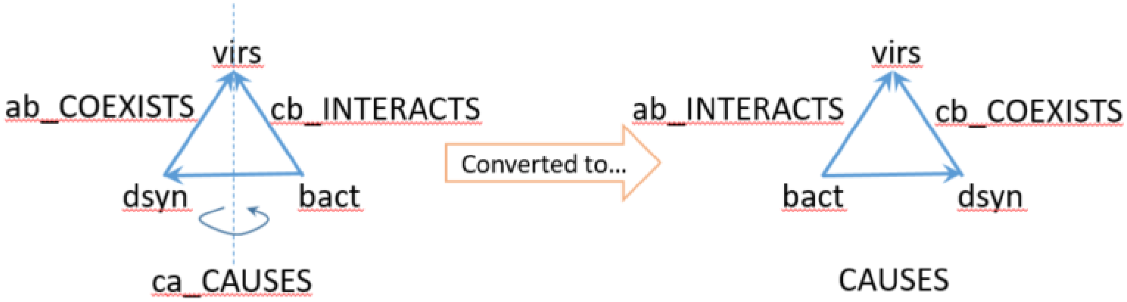
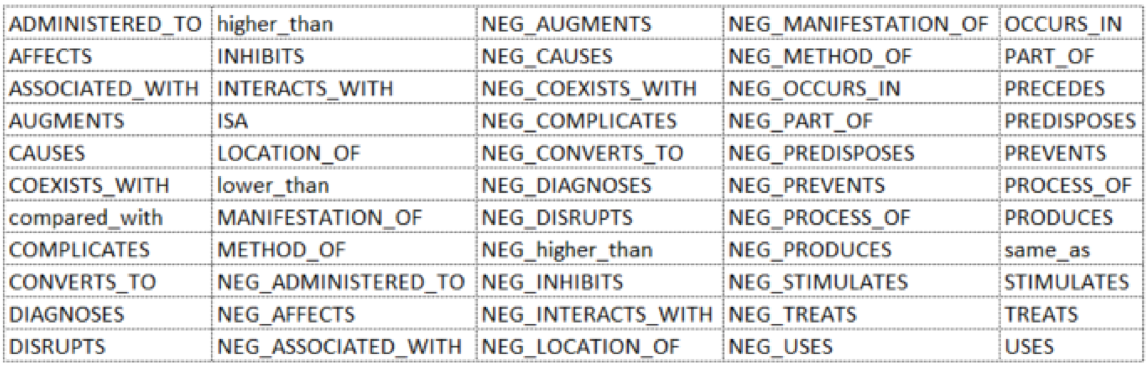


Fig. 3.7 Example of Triple Rotation



ADMINISTERED_TO	higher_than	NEG_AUGMENTS	NEG_MANIFESTATION_OF	OCCURS_IN
AFFECTS	INHIBITS	NEG_CAUSES	NEG_METHOD_OF	PART_OF
ASSOCIATED_WITH	INTERACTS_WITH	NEG_COEXISTS_WITH	NEG_OCCURS_IN	PRECEDES
AUGMENTS	ISA	NEG_COMPLICATES	NEG_PART_OF	PREDISPOSES
CAUSES	LOCATION_OF	NEG_CONVERTS_TO	NEG_PREDISPOSES	PREVENTS
COEXISTS_WITH	lower_than	NEG_DIAGNOSES	NEG_PREVENTS	PROCESS_OF
compared_with	MANIFESTATION_OF	NEG_DISRUPTS	NEG_PROCESS_OF	PRODUCES
COMPLICATES	METHOD_OF	NEG_higher_than	NEG_PRODUCES	same_as
CONVERTS_TO	NEG_ADMINISTERED_TO	NEG_INHIBITS	NEG_STIMULATES	STIMULATES
DIAGNOSES	NEG_AFFECTS	NEG_INTERACTS_WITH	NEG_TREATS	TREATS
DISRUPTS	NEG_ASSOCIATED_WITH	NEG_LOCATION_OF	NEG_USES	USES

Table 3.10 List of Outcomes for Only A to C Direction of Relationships

accordingly. For an example, refer to Figure 3.5. In order to turn a C to A direction into a A to C direction, the A and C nodes are flopped; the A to B relationship becomes a C to B relationship; and the B to C relationship becomes a B to A relationship. There are four of the possible triple permutations that need to be flipped as shown in Figure 3.6 by the triples with axis and arrow rotation indication (second row of triples).

A concrete example using the notation used in ARFF representations (Section 3.4) would be "*dsyn, virs, bact, ab_COEXISTS_WITH, cb_INTERACTS_WITH, ca_CAUSES*" for A, B and C semantic types where *dsyn* is the semantic type abbreviation for for "Disease or Syndrome", *virs* for "Virus" and *bact* for "Bacterium", and where the three relationship types with indicated directions are "Coexists with" in the A to B direction, "Interacts with" in the C to B direction and "Causes" in the C to A direction. See Figure 3.7. This triple is converted so that the "Causes" in the C to A direction (*ca_CAUSES*) will become simply a *CAUSES* relationship with the direction always being from A to C. To do this, the *COEXISTS_WITH* in the direction from A to B becomes a *COEXISTS_WITH* in the C to B direction and the *INTERACTS_WITH* in the direction from C to B becomes *INTERACTS_WITH* in the A to B direction. The resulting training set is: "*bact, virs, dsyn, ab_INTERACTS_WITH, cb_COEXISTS_WITH, CAUSES*" where the direction of the third relationship is always A to C.

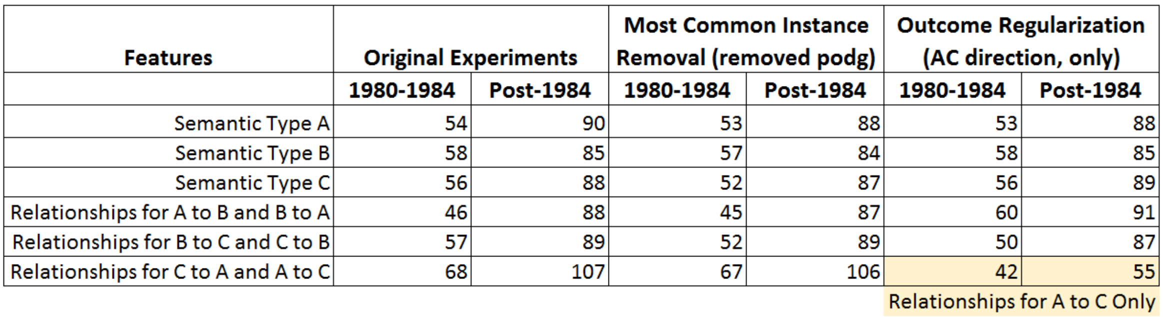
Table 3.10 shows the set of 55 possible outcomes after normalizing the data to have only the A to C direction. The number of outcomes is now 42 in the 1980-1984 data, down from 68, and 55 in the post-1984 data set, down from 107. The number of semantic types for A and C nodes also changed after regularization – some increased, some decreased. The results of the top thirty of each feature and predicted outcome after normalizing to using only the A to C outcome direction are shown in Tables 3.11 and 3.12.

A Sem Type	Count	B Sem Type	Count	C Sem Type	Count	A to B or B to A Relationship	Count	B to C or C to B Relationship	Count	A to C Relationship	Count
medd	948	dsyn	1517	podg	2641	ab_PREVENTS	842	bc_AFFECTS	1715	TREATS	1293
dsyn	814	orch	543	aapp	587	ba_AFFECTS	656	bc_ADMINISTERED_TO	727	PROCESS_OF	921
phsu	611	bacs	379	phsu	323	ba_COEXISTS_WITH	490	bc_LOCATION_OF	353	LOCATION_OF	610
orch	584	tisu	251	bacs	223	ba_LOCATION_OF	443	bc_COEXISTS_WITH	298	ADMINISTERED_TO	524
aapp	522	aapp	241	mamm	159	ab_AFFECTS	245	cb_AFFECTS	197	PART_OF	373
bacs	253	bpoc	212	dsyn	137	ba_compared_with	234	bc_PART_OF	191	INTERACTS_WITH	275
podg	220	fndg	194	orch	114	ba_ADMINISTERED_TO	173	bc_NEG_PROCESS_OF	180	ISA	250
bpoc	134	antb	170	bpoc	81	ab_INTERACTS_WITH	170	bc_INTERACTS_WITH	152	COEXISTS_WITH	169
tisu	99	carb	140	tisu	78	ba_INTERACTS_WITH	167	cb_COEXISTS_WITH	139	STIMULATES	96
antb	67	orgf	136	popg	73	ba_PART_OF	135	cb_LOCATION_OF	125	AFFECTS	90
horm	60	mamm	135	cell	48	ba_INHIBITS	126	bc_NEG_PART_OF	106	INHIBITS	80
mobd	60	patf	114	lipd	46	ab_TREATS	124	cb_ADMINISTERED_TO	84	compared_with	42
mamm	58	cell	81	inch	37	ab_INHIBITS	111	cb_INTERACTS_WITH	68	CAUSES	42
cell	51	popg	65	hops	34	ab_LOCATION_OF	109	bc_TREATS	67	AUGMENTS	27
hops	44	enzy	59	neop	32	ab_COEXISTS_WITH	108	cb_STIMULATES	66	DISRUPTS	26
eico	43	nsba	59	eico	31	ba_ISA	107	bc_INHIBITS	58	ASSOCIATED_WITH	26
ortf	41	bact	53	bact	27	ab_ADMINISTERED_TO	91	cb_PART_OF	57	NEG_INTERACTS_WITH	21
carb	38	moft	53	medd	27	ba_TREATS	90	bc_compared_with	56	OCCURS_IN	15
strd	29	horm	49	carb	23	ab_PART_OF	78	cb_ASSOCIATED_WITH	40	PRODUCES	14
elii	25	phsu	46	fndg	21	ab_PROCESS_OF	65	bc_ISA	32	higher_than	13
lipd	23	lipd	37	orgf	20	ba_NEG_AFFECTS	52	bc_PROCESS_OF	29	USES	12
bact	22	emst	35	mobd	15	ab_ASSOCIATED_WITH	41	bc_ASSOCIATED_WITH	29	NEG_PROCESS_OF	11
inch	21	celc	34	elii	14	ba_STIMULATES	41	bc_CAUSES	19	NEG_PART_OF	10
genf	19	hops	33	anim	14	ab_NEG_AFFECTS	40	cb_PRODUCES	18	NEG_ADMINISTERED_TO	8
neop	19	elii	32	antb	14	ab_STIMULATES	40	bc_AUGMENTS	17	NEG_TREATS	7
celc	19	neop	32	horm	14	ab_compared_with	28	cb_TREATS	17	DIAGNOSES	6
topp	17	eico	28	aggp	13	ab_CAUSES	25	cb_PROCESS_OF	16	NEG_STIMULATES	5
anim	16	chem	20	virs	13	ab_ISA	23	bc_NEG_AFFECTS	15	NEG_INHIBITS	5
fngs	14	rcpt	20	nsba	13	ba_DISRUPTS	17	cb_INHIBITS	14	PREVENTS	4
inpo	13	opco	17	strd	12	ba_PRODUCES	15	cb_compared_with	13	NEG_LOCATION_OF	3

Table 3.11 Distributions With Only AC Relationship Direction from 1980-1984 Data

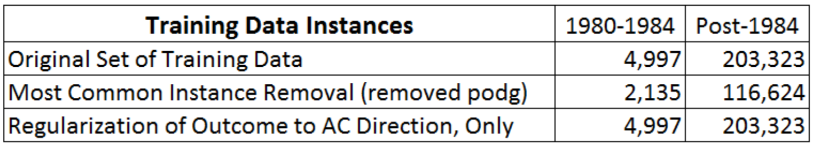
A Sem		B Sem		C Sem		A to B or B to A		B to C or C to B			
Type	Count	Type	Count	Type	Count	Relationship	Count	Relationship	Count	A to C Relationship	Count
dsyn	29499	patf	29631	podg	79389	ba_COEXISTS_WITH	35022	bc_AFFECTS	80332	PROCESS_OF	65995
genf	26849	dsyn	25859	aapp	17116	ab_AFFECTS	27732	bc_COEXISTS_WITH	16295	TREATS	20977
aapp	25204	mobd	18482	dsyn	13694	ba_PROCESS_OF	26833	cb_AFFECTS	14866	LOCATION_OF	14130
topp	17954	aapp	15708	phsu	11513	ba_AFFECTS	19450	bc_ADMINISTERED_TO	14338	PART_OF	13432
bacs	15734	bacs	12849	bacs	9312	ba_LOCATION_OF	13305	bc_LOCATION_OF	10336	INTERACTS_WITH	12943
orch	14180	orch	11127	popg	8601	ba_INTERACTS_WITH	8895	bc_INTERACTS_WITH	6808	ISA	11525
phsu	10251	bpoc	8787	mamm	5974	ba_ASSOCIATED_WITH	7413	bc_PART_OF	6024	ADMINISTERED_TO	9594
podg	7051	cell	6040	orch	5277	ab_COEXISTS_WITH	5070	bc_NEG_PART_OF	4442	COEXISTS_WITH	7437
patf	7024	anab	5828	topp	4187	ba_compared_with	4496	cb_LOCATION_OF	3574	AFFECTS	6157
bpoc	4530	imft	5813	neop	4010	ba_INHIBITS	4181	bc_NEG_PROCESS_OF	3350	STIMULATES	4944
orgf	3108	orgf	4901	celf	3965	ba_PART_OF	3962	cb_COEXISTS_WITH	3239	ASSOCIATED_WITH	4748
horm	3098	tisu	4116	bpoc	3487	ab_TREATS	3802	bc_TREATS	3192	USES	4298
cell	2834	elii	3843	tisu	3194	ab_INTERACTS_WITH	3117	bc_ASSOCIATED_WITH	2936	CAUSES	4177
hops	2634	phsu	3592	cell	2797	ab_LOCATION_OF	3015	bc_compared_with	2832	INHIBITS	3930
neop	2490	neop	2639	orgf	2786	ab_PART_OF	2886	bc_CAUSES	2782	AUGMENTS	3410
tisu	2337	antb	2393	patf	2292	ab_INHIBITS	2827	cb_INTERACTS_WITH	2770	DISRUPTS	2688
gngm	2208	fndg	2316	inch	2133	ab_CAUSES	2357	bc_INHIBITS	2764	PRODUCES	2162
carb	2137	enzy	2298	horm	1436	ba_ADMINISTERED_TO	2294	cb_PART_OF	2354	compared_with	2155
lipd	1926	topp	2236	hops	1423	ba_PRODUCES	2180	bc_PROCESS_OF	2160	PREDISPOSES	1272
imft	1819	nsba	1967	elii	1334	ba_TREATS	2101	cb_INHIBITS	1777	OCCURS_IN	906
ortf	1760	carb	1962	ortf	1315	ab_ASSOCIATED_WITH	1746	bc_PRODUCES	1361	DIAGNOSES	886
elii	1696	mamm	1874	gngm	1290	ba_NEG_PART_OF	1714	bc_AUGMENTS	1164	NEG_PROCESS_OF	767
inch	1529	podg	1682	fndg	1188	ab_DISRUPTS	1459	cb_ASSOCIATED_WITH	1006	NEG_INTERACTS_WITH	653
antb	1463	horm	1616	carb	1136	ba_CAUSES	1451	cb_CAUSES	871	NEG_AFFECTS	576
mamm	1255	hops	1494	lipd	1063	ab_ADMINISTERED_TO	1299	cb_PROCESS_OF	835	PREVENTS	575
mobd	1085	celc	1477	genf	935	ba_USES	1279	cb_ADMINISTERED_TO	808	higher_than	497
inpo	1077	dora	1464	nsba	876	ab_PRODUCES	1187	bc_STIMULATES	802	PRECEDES	289
celf	1007	sosy	1451	aggp	822	ab_AUGMENTS	1091	cb_AUGMENTS	788	METHOD_OF	226
diap	963	gngm	1425	famg	822	ba_STIMULATES	1081	cb_compared_with	712	NEG_LOCATION_OF	203
nsba	959	famg	1345	bact	641	ab_STIMULATES	1025	cb_TREATS	641	NEG_TREATS	189

Table 3.12 Distributions With Only AC Relationship Direction from Post-1984 Data



Features	Original Experiments		Most Common Instance Removal (removed podg)		Outcome Regularization (AC direction, only)	
	1980-1984	Post-1984	1980-1984	Post-1984	1980-1984	Post-1984
Semantic Type A	54	90	53	88	53	88
Semantic Type B	58	85	57	84	58	85
Semantic Type C	56	88	52	87	56	89
Relationships for A to B and B to A	46	88	45	87	60	91
Relationships for B to C and C to B	57	89	52	89	50	87
Relationships for C to A and A to C	68	107	67	106	42	55
					Relationships for A to C Only	

Table 3.13 Summary of Features (Counts of unique values)



Training Data Instances	1980-1984	Post-1984
Original Set of Training Data	4,997	203,323
Most Common Instance Removal (removed podg)	2,135	116,624
Regularization of Outcome to AC Direction, Only	4,997	203,323

Table 3.14 Total Numbers of Training Records

3.6 Chapter Summary

To summarize, this chapter has presented the training data that will be used in classification experimentation. A total of 15,427 concepts have been identified. These concepts are used to identify the training data by using them in queries against SemMedDB. The fully qualified triples are converted into training data by retrieving the semantic types of the A, B and C concepts and by using the relationships found in SemMedDB. These are the relationships and directions between the concept pairs: A to B, B to C, and C to A. Summaries of the semantic types and relationships are shown in Table 3.13 and summaries of the training instances are shown in Table 3.14.

Additional features were considered during experimentation including breaking up of relationships as noted in the previous section (Section 3.5.2). Another idea was refinement of features to possible combine and further normalize relationship features that are very similar, for example, LOCATION_OF, PART_OF and COEXISTS_WITH. This would require validation by experts that the combinations were acceptable in meaning. Beyond changes to the actual features used, additional features related to language processing of LBD could have been considered. For example, the distance between concepts in the documents could have been added as a feature, where the distance could be an indication of how many words separated the pairs of concepts. Ultimately, the class designs chosen with the refinement of regularizing outputs produced very good results and no further experimentation was explored.

Chapter 4

Relationship Prediction Experiments

This chapter explores the creation and evaluation of classifier models using the data representations developed in the previous chapter. Evaluation methods will be discussed as will comparisons of classifier performance using various data designs.

The general steps to train and evaluate classifiers that may be used to explain LBD relationships are as follows:

1. Establish evaluation methodologies that will be used during classifier experimentation
2. Study performance of a range of classifiers
3. Perform more in-depth classification experiments including data refinements and feature ablation

Section 4.1 discusses the evaluation methodologies used in the experiments. Section 4.2 presents classification results using various classifiers on the regularized outcomes design. Section 4.3 discusses the various experiments performed including those that test against variations of the class designs, those that try various ablation tests, and those that regularize the outcomes. Section 4.4 has a summary of the performance of classification as a means to explain A to C relationships using training data.

4.1 Evaluation Methodologies for Experiments

This section provides a brief overview of the various evaluation techniques used to evaluate performance of the various experiments presented in Sections 4.2 and 4.3. Seven methods of evaluation are presented: cross-validation, confusion matrices, learning curves, comparison to most frequent class, deltas, hypothesis testing, and ablation.

Using Cross-Validation Cross-validation is a common technique that evaluates a classifier's *expected* performance when that classifier is ultimately used to predict unknown outcomes (for example, when the classifier is used to explain LBD relationships). This is achieved by repeatedly training on most of the data while holding out a small portion of the data which is then used to test the classifier's ability to accurately classify those held out and comparing with their known correct results. In this work, the accurately classified percentages are the result of ten-fold cross-validation. First train on 90% of the data, hold out 10% of the data for testing. Next select a different 90% on which to train and test on the remaining 10%. Do this for ten iterations and average the results to estimate the classifier's performance on unseen data.

Studying Confusion Matrices and other Metrics Confusion matrices provide useful information about how a classifier has performed during the training and evaluation of the classifier. In this work, the confusion matrix is not simply a binary prediction matrix showing true positives (often noted as TP), true negatives (TN), false positives (FP) and false negatives (FN) (Dietterich, 1998; Sokolova and Lapalme, 2009). The problem modeled in this work is a multi-class, multi-value problem that produces an *m x m* confusion matrix, where m is the number of possible outcomes. This *m x m* confusion matrix helps to identify outcomes that may be related if a specific outcome shows up as a false positive a significant number of times when compared to the expected outcome. The performance of the classifier may be degraded because of these false positives. In the medical domain, an A concept being *PART_OF* a C concept and a C concept having *LOCATION_OF* an A concept are examples of related relationships and they may both produce valid outcomes for a particular A to C pair of concepts. For example, if A is a toe and C is a foot, then it is logical that a toe is a part of a foot (*ac_PART_OF*) and the foot is the location of a toe (*ca_LOCATION_OF*). In the data presented here, *ac_PART_OF* and *ca_LOCATION_OF* would be the outcomes of this example where *ac_PART_OF* might be the correct classification and *ca_LOCATION_OF* may show up as a false positive for that classification. However, the toe and foot example shows that these two relationships, may be, equivalent inverse relationships for each other.

TP, FP along with precision and recall metrics provide additional information about the classifier performance at predicting outcomes (Witten et al., 2011). Precision is calculated as:

$$precision = \frac{TP}{(TP+FP)} \tag{4.1}$$

And recall is calculated as:

$$recall = \frac{TP}{(TP + FN)} \tag{4.2}$$

Learning Curves Learning curves break the data held out for training into smaller parts and show the results on how well, or poorly, the classifiers are able to predict the outcomes of the test data set. For example, the 90% of the data used in 10-fold cross-validation training is further broken down into smaller sets and tested against the 10% held out for testing the classifier results.

Comparison to Most Frequently Occurring Class Another common way to evaluate of the performance of a classifier is by comparing results against a baseline classifier. In this work, choosing the most frequently occurring outcome is the baseline classifier. Weka's ZeroR classifier implements this approach.

Looking at Deltas Simple differences or deltas between results are effective in comparing the performance of one data design to another or one classifier to another. This is especially true when the percent of accurately classified outcomes using 10-fold cross-validation present very different results. As an example where simple deltas help in analyzing performance, consider a baseline classifier that presents a result of 10% accurately classified using 10-fold cross-validation and another classifier that presents a result of 50% accurately classified also using 10-fold cross-validation. Here, the 40% difference shows that the classifier under test performs much better than the baseline. Deltas do not replace the more scientifically defensible methods like hypothesis testing presented in the next section. They just work well when the differences between results under test are are very large. Deltas are used when examining the average percent classified accurately using cross-validation (the mean percentage correct across the n-folds).

Using Null Hypothesis Testing or Significance This paragraph presents basic background information in how hypothesis testing provides statistical backing for the importance or significance of discovered results. Conceptually, hypothesis testing first identifies an experiment that is the null hypothesis. Then, another experiment is designed and carried out with an assumption that it *might* perform significantly better than the null hypothesis. If the new experiment does, then the null hypothesis of the original experiment may be rejected. (Demšar, 2006; Hastie et al., 2009; Dietterich, 1998) This section also discusses briefly how hypothesis testing may be accomplished in this work.

A normal distribution bell curve expects samples to mostly fall within some standard deviation multiple to the left or right of the mean of the distribution. For example, using a two sigma interval, we expect 95.45% of instances will be within the two sigma to the left or right of the average. In hypothesis testing, the goal is to identify the normal distribution statistics (mean, standard deviation and degrees of freedom) of the null hypothesis and of the other hypothesis or data set under test. Then, using these statistics, we are able to calculate a t-score[1] that will help to identify how significant the other hypothesis is in outperforming the null hypothesis. If the statistics show a 5% chance or less (or much less in some cases) of being the same as the null hypothesis, we may reject the null hypothesis.

In this work, the t value is calculated as follows:

$$\frac{(\bar{x_1} - \bar{x_2})}{\sqrt{\frac{(\sigma_1)^2}{n_1} + \frac{(\sigma_2)^2}{n_2}}} \tag{4.3}$$

where $\bar{x_1}$ and $\bar{x_2}$ are the means from the two samples; σ_1 and σ_2 are the standard deviations; and n_1 and n_2 are the total number in each sample.

When originally used by statisticians and other analysts, the t-value was turned into a t score by performing table look-ups based on degrees of freedom ($n_1 + n_2 - 2$) of the data where n_1 and n_2 are the same total numbers in each sample being studied. The t score represents how far to the left or right of the null hypothesis' bell curve the other hypothesis falls. If, for example, the t score is 0.01, this is in the 99% realm of non-likelihood, and therefore would indicate acceptable rejection of the null hypothesis. Hypothesis testing techniques are more informative than simple deltas, especially when the deltas between results are small.

In this work, the t value is calculated by using the percentage classified accurately from each of the classifier's ten folds. Each fold presents a percentage classified accurately during that fold's training. The 10-fold scores are used to generate various statistics including a t-score using Excel's T.Test function.

Ablation of Features Ablation involves systematically removing features from the training data and rerunning classifier training and evaluating the results. The purpose of ablation is to

[1] The t-score is based on the student's t-test which was developed by an employee of the Guinness brewery, William Gosset, while he tried to understand the quality of various grain crops grown by his employer. It is called Student's t-test because, as the story goes, in order to publish his research, his employer did not want his name to be used and associated with the Guinness brewery. Therefore the paper describing the t-test was authored, simply, by Student and his test is still called Student's t-test (Livingston, 2004)

identify features that may or may not contribute to the successful classification of data. For example, if there are five features used to predict an outcome, then, five ablation runs are used – each run removes one of the five features. After analysis of removing each feature, additional experiments may remove sets of features, again, to see and study how the features interact or how independent they are.

Ablation is able to help to identify feature dependencies. That is, if removing a feature degrades performance, then that feature is contributing to successful classification. If performance improves after removing a feature, then this may indicate noise being introduced by that feature, and there may be interaction between this feature and one, or more, of the others. Ablation is more of an experiment than an evaluation methodology. However, it does provide an evaluation of the feature design.

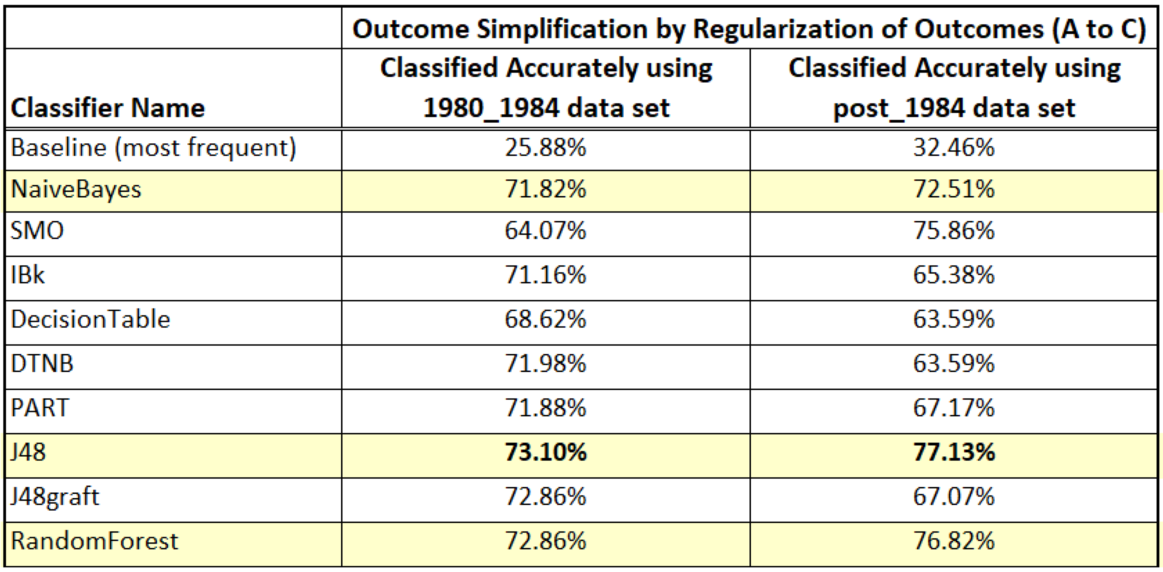
4.2 Classifier Comparisons

The purpose of this section is to explore how various classifier algorithms perform using self-test (10-fold cross-validation) on training data. Classification performance results are based on the regularized outcomes data design. This data design is described in Section 3.5.2 and the performance of this design is studied more in-depth in Section 4.3.3. The classifiers applied to the problem were introduced in Section 2.7. Section 4.3 will study a subset of these classifiers in more detail.

The classifiers evaluated in this section represent a cross-section of classifier families including support vector machines (SVM), rules, linear regression, tree algorithms and hybrids of classifiers. Chapter 3 describes details of the data sets. Classifiers are trained and tested on both the 1980-1984 and the post-1984 data sets. The results are shown in Table 4.1. The yellow highlighted rows (Naïve Bayes, J48 and Random Forest) are the classifiers that are the focus of more in-depth studies presented in Section 4.3.

All of the classifiers performed much better than the baselines. This is expected since there are a large number of possible outcomes in the data which makes a baseline that selects the most frequently occurring outcome to be less meaningful and less likely to be accurate. This is also good that the baselines performed poorly compared to all of the other classifier algorithms showing that the problem of explaining LBD relationships may be solved using classification.

The J48 classifier has the highest percentage of accurately classified results on both the 1980-1984 and the post-1984 data sets. Other non-baseline classifiers are able to predict with reasonable success with no classifier predicting less than 63% against either data set.



	Outcome Simplification by Regularization of Outcomes (A to C)	
Classifier Name	**Classified Accurately using 1980_1984 data set**	**Classified Accurately using post_1984 data set**
Baseline (most frequent)	25.88%	32.46%
NaiveBayes	71.82%	72.51%
SMO	64.07%	75.86%
IBk	71.16%	65.38%
DecisionTable	68.62%	63.59%
DTNB	71.98%	63.59%
PART	71.88%	67.17%
J48	**73.10%**	**77.13%**
J48graft	72.86%	67.07%
RandomForest	72.86%	76.82%

Table 4.1 Classifier Performance on the 1980-1984 and post-1984 Data Sets

The PART and J48Graft classifiers which are variations of the C4.5 algorithm that J48 also implements, interestingly, did not perform as well as the J48. The PART variation was meant to perform better on larger data sets than C4.5 by itself.

Some of the classifiers performed better on the smaller data set than the larger. This could be because of some overfitting affects using the smaller data sets. This is quite possibly the explanation of the J48graft results where the 1980-1984 data set performed better. Overfitting is mentioned as a problem with grafting in Webb (1999) who developed the J48graft algorithm.

The SMO classifier which is a support vector machine implementation performed reasonably well on the larger data set but was very slow to compute its outcomes on the hardware used in this work. This is the reason why SMO was not studied further in the next sections.

4.3 Experiments and Evaluation of Data Designs

This section presents classification experiments performed using various data designs presented in Chapter 3. J48, Naïve Bayes and basic Random Forest algorithms, implemented in Witten et al. (1999, 2011), are used for the deeper examination of data designs and related performance. These provide a cross-section of decision trees, Bayesian and random forest algorithmic approaches, respectively. They were described in more detail in Section 2.7.

The set of experiments, presented in this section, study how changes in data designs change the performance of the classifiers in their ability to predict the outcome of the A to C relationship. These include studies of a basic design which is also referred to as initial or original design (Section 4.3.1), a design that removes instances with the most commonly occurring feature values (Section 4.3.2), a design that reduces the number of possible predicted outcomes (Section 4.3.3), and some ablation studies where features are systematically removed and classification results are analyzed (Section 4.3.4). Evaluation techniques were presented in Section 4.1, and the classifiers used were discussed in Section 2.7. The sections that follow will each present the experiment performed, provide some form of evaluation and discuss the experiment's results.

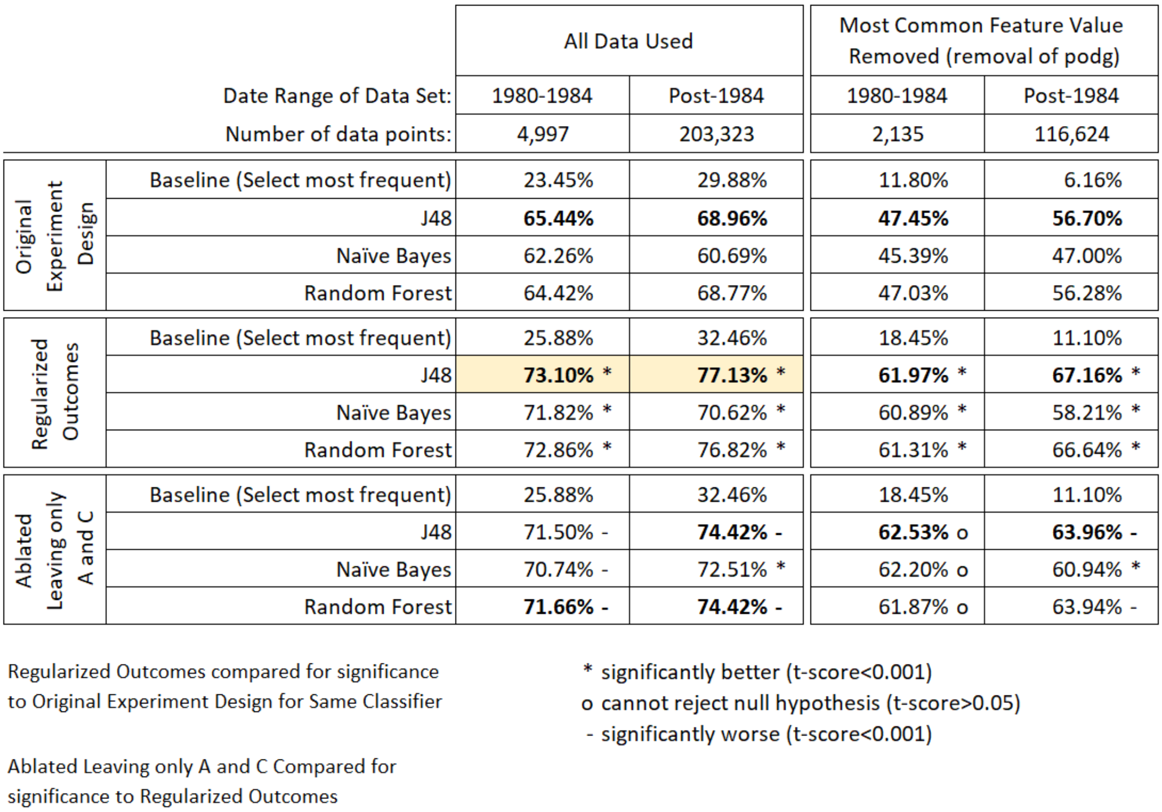
As noted in Section 3.2, data from both the 1980-1984 time slice and post-1984 are used in an effort to train with more significant numbers of data points. There are only 4997 in the original and the regularized 1980-1984 training set while the post-1984 data has many more with 203,323 records. Removal of instances with the most commonly occurring feature value, *podg*, resulted in fewer data points with 2135 in the 1980-1984 set, and, again, many more with 116,624 records in the post-1984 set.

4.3.1 Initial Data Design

Classification models were initially built using all training data identified from the SemMedDB (Section 3.2). This produced results that were consistently better than baselines that simply chose the most frequently occurring outcome on the same data sets. This section's experiment uses cross-validation and confusion matrix analysis for evaluation (see Section 4.1). This section also introduces the baseline for null hypothesis testing used in later experiments. The initial data design is also referred to as the original data, original feature design or original experiment design. The full data set was used in evaluating the original experiment design.

Initial Data Design - Comparison to Most Frequently Occurring Class

Selecting the most frequently occurring outcome provides the baseline during evaluation of classifier results. A baseline of 23.45% accurately predicted is found in the 4997 training data set derived from articles with dates ranging between 1980 and 1984. Applied to the 203,323 training data set derived from post-1984 articles, the data shows a baseline of 29.88% accurately classified outcomes. These results are shown in the row called baseline in the top



		All Data Used		Most Common Feature Value Removed (removal of podg)	
	Date Range of Data Set:	1980-1984	Post-1984	1980-1984	Post-1984
	Number of data points:	4,997	203,323	2,135	116,624
Original Experiment Design	Baseline (Select most frequent)	23.45%	29.88%	11.80%	6.16%
	J48	**65.44%**	**68.96%**	**47.45%**	**56.70%**
	Naïve Bayes	62.26%	60.69%	45.39%	47.00%
	Random Forest	64.42%	68.77%	47.03%	56.28%
Regularized Outcomes	Baseline (Select most frequent)	25.88%	32.46%	18.45%	11.10%
	J48	**73.10%** *	**77.13%** *	**61.97%** *	**67.16%** *
	Naïve Bayes	71.82% *	70.62% *	60.89% *	58.21% *
	Random Forest	72.86% *	76.82% *	61.31% *	66.64% *
Ablated Leaving only A and C	Baseline (Select most frequent)	25.88%	32.46%	18.45%	11.10%
	J48	71.50% -	**74.42%** -	**62.53%** o	**63.96%** -
	Naïve Bayes	70.74% -	72.51% *	62.20% o	60.94% *
	Random Forest	**71.66%** -	**74.42%** -	61.87% o	63.94% -

Regularized Outcomes compared for significance to Original Experiment Design for Same Classifier

Ablated Leaving only A and C Compared for significance to Regularized Outcomes

* significantly better (t-score<0.001)
o cannot reject null hypothesis (t-score>0.05)
- significantly worse (t-score<0.001)

Table 4.2 Primary Classifier Evaluation for All Experiments (Bold indicates best result for that experiment and data set. J48 using Regularized Outcomes performs best.)

third section labeled as "Original Experiment Design" under the "All Data Used" columns in Table 4.2.

Initial classifier training and evaluation was performed using the training data described in Section 3.2 and the original feature design presented in Section 3.3. As was noted at the end of Section 3.2, post-1984 data is included in training data for classifiers because it provides a much larger data set. Careful consideration of this data set ensured that the diseases of interest were not included in this set of training data. This is important because classifiers trained in this chapter will be used in the next chapter to explain literature based discoveries where some of the focus is on a set of diseases of interest. The post-1984 data set is more than forty times larger than the 1980-1984 time slice. The reason for using the 1980-1984 time slice was introduced in Section 2.3 and is only to provide an older time slice where LBD discoveries can then be validated using newer data.

Table 4.2 provides a summary of the results for the three primary classifiers on the original feature design using the average of accurately classified results from cross validation to evaluate performance. This table also contains results for the other experiments that will be explained later. The results for this experiment are listed in the rows for "Original Experiment Design" and under the "All Data Used" columns. J48 and Random Forest algorithms, with results ranging from 64.42 to 68.96% classified accurately, performed similarly which is logical as both algorithms are based on decision trees. Significance by simple observation of the large deltas between the baselines and the other classifiers show that these classifiers are much more successful at predicting the outcomes than simply picking the most frequently occurring outcome found in the training data. Hypothesis testing using the baseline classifier as the null hypothesis produce t-scores very, very close to zero supporting this observation (zero out to more than 10 decimal points). Significance statistics become more interesting as experiments are refined throughout the rest of this chapter. Naïve Bayes classifier was also much better than the baselines but had lower percents classified accurately when compared with the tree based classifiers.

Initial Data Design - Evaluation Using Confusion Matrices and other Metrics

The confusion matrices from the experiments produced information indicating that the relationships predicted included a number of false positives. Evaluation by studying confusion matrices was discussed in Section 4.1. To study some of the results in the confusion matrices, a threshold for the number of false positives was selected. The number of occurrences of the false positive had to be greater than 50 in the 1980-1984 training data (which had 4997 total training records) and greater than 1000 in the post-1984 data (203,323 total training records). The threshold numbers 50 and 1000 for the 1980-1984 and post-1984 training data, respectively, were chosen to provide a sample of a few of the most commonly occurring false positives. The entire confusion matrix for this work is an *m x m* matrix where m is the number of outcomes which is 68 for the 1980-1984 data set and 107 for the post-1984 data set using the original data design.

Confusion matrix of 1980-1984 trained classifiers: See confusion matrix summary results in Table 4.3. Actual relationship and direction of *ca_PART_OF* was found to be related to *ac_LOCATION_OF* which logically makes sense. Take the example, introduced in Section 4.1, of a toe which could represent an A concept. A toe is part of a foot (where foot could be the C concept). Concluding that inverse that a foot is the location of a toe is also true. A to C "part of" relationship may be described using the inverse of C to A "location of" relationship. Treats and administered to (*ca_TREATS* and *ca_ADMINISTERED_TO*) and the reverse (*ca_ADMINISTERED_TO* and *ca_TREATS*) were also found in this set of data to show false positive results. Some pharmaceutical would need to be administered before it

Actual	False Positive	J48	Random Forest	Naïve Bayes
ca_TREATS	ca_ADMINISTERED_TO	x	x	
ca_TREATS	ac_LOCATION_OF			x
ca_PART_OF	ac_LOCATION_OF	x	x	x
ca_ADMINISTERED_TO	ca_TREATS	x	x	x

Table 4.3 Original feature design, 1980-1984, most commonly occurring false positives

was determined to treat – this is also a logically similar relationship. The relationship and its inverse are presented as false positives in the training of the classifiers with this original feature design.

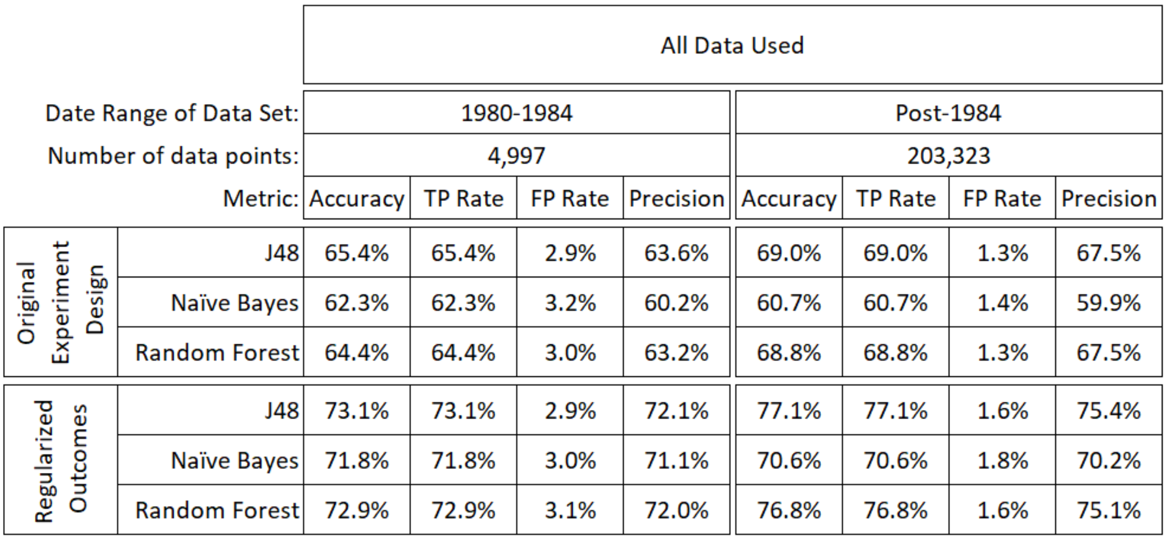
Confusion matrix of post-1984 trained classifiers: See confusion summary matrix results in Table 4.4. With more training data points came more false positives, especially with the naïve Bayes algorithm. J48 and Random Forest classifiers presented two of the same false positives. Naïve Bayes classifier produced many more false positives than with the smaller data set. One false positive, *ca_TREATS*, that appeared with the actual *ca_ADMINISTERED_TO* relationship in the old data set but did not show up in the large data set.

Two new false positives showed up with all three classifiers. These are the last two rows of the table. One is that *INTERACTS_WITH* in the ac direction is being confused with the ca direction. This makes sense as interacts with can logically be considered a commutative relationship where order does not matter. The other false positive that was new with the newer data set is that of *ac_USES* being falsely predicted as *ca_ADMINISTERED_TO*. These two relationships may also be equivalent relationships in some cases. For example, a drug may be administered to a patient means a very similar thing as a patient using a drug.

Additional metrics: Table 4.5 summarizes additional metrics about the classifier training performance. Because there are multiple outcomes to predict, these metrics do not show much other than the fact that there were some inaccurately classified results (the false positive rate) that led to some reduction in precision. Additionally, the post-1984 data set shows improved precision by the fact that the false positive rate is smaller.

Actual	False Positive	J48	Random Forest	Naïve Bayes
ca_TREATS	ca_PART_OF			x
ca_TREATS	ca_ADMINISTERED_TO	x	x	x
ca_TREATS	ac_LOCATION_OF			x
ca_PROCESS_OF	ac_LOCATION_OF			x
ca_PART_OF	ac_LOCATION_OF	x	x	x
ca_INTERACTS_WITH	ac_ISA			x
ca_INTERACTS_WITH	ac_INTERACTS_WITH			x
ac_USES	ca_ADMINISTERED_TO	x	x	x
ac_INTERACTS_WITH	ca_INTERACTS_WITH	x	x	x

Table 4.4 Original feature design, post-1984, most commonly occurring false positives



		All Data Used							
	Date Range of Data Set:	1980-1984				Post-1984			
	Number of data points:	4,997				203,323			
	Metric:	Accuracy	TP Rate	FP Rate	Precision	Accuracy	TP Rate	FP Rate	Precision
Original Experiment Design	J48	65.4%	65.4%	2.9%	63.6%	69.0%	69.0%	1.3%	67.5%
	Naïve Bayes	62.3%	62.3%	3.2%	60.2%	60.7%	60.7%	1.4%	59.9%
	Random Forest	64.4%	64.4%	3.0%	63.2%	68.8%	68.8%	1.3%	67.5%
Regularized Outcomes	J48	73.1%	73.1%	2.9%	72.1%	77.1%	77.1%	1.6%	75.4%
	Naïve Bayes	71.8%	71.8%	3.0%	71.1%	70.6%	70.6%	1.8%	70.2%
	Random Forest	72.9%	72.9%	3.1%	72.0%	76.8%	76.8%	1.6%	75.1%

Table 4.5 Metrics Evaluation for All Original and Regularized Outcomes Data Designs (Accuracy is the same as recall)

Initial Data Design - Evaluation Using Significance

Cross-validation provides an overall accurately classified percentage that may be used to understand how well the classifier performs. Comparing one classifier's results to another requires more work. As noted in Section 4.1, comparing the accurately classified percentages to baseline numbers and examining the gross deltas between these percentages is acceptable when the differences are large.

The focus on significance or hypothesis testing is not for this experiment and data design. Instead, this classifier design and results provide a null hypothesis that will be used to do significance testing with the experiments presented in Sections 4.3.3 and 4.3.4.

Discussion of Initial Data Design

To summarize the results of this section, the initial data and feature design, and the classifier algorithms chosen (J48, Naïve Bayes and Random Forest) performed reasonably well as is indicated by comparing with baseline results on the same data and classifiers. The J48 and Random Forest classifiers produced a similar percentage of accurately classified results and Naïve Bayes classifier produced slightly worse results using the 1980-1984 training set (approximately 2.5% less) and 8% lower results when trained on the post-1984 data.

Random Forest performed reasonably well as was suggested by Fernández-Delgado et al. (2014), but J48 also performed similarly. This is interesting because Random Forest and J48 are similar but the expectation is that Random Forests should perform better than J48 because of Random Forest algorithm's ability to not overfit to the data. Naïve Bayes performs relatively better than the baseline just as J48 and Random Forest did, but the confusion matrix showed that it presented more noise which may have contributed to its slightly lower classification accuracy.

This experiment and its data design will be used as the null hypothesis which the outcome regularization (Section 4.3.3) will attempt to reject. These two data sets, the 1980-1984 and post-1984 from "All Data Used" set, will be used as the basis of the modified data representations for the additional experiments that regularize outcomes and apply ablation that are presented in Sections 4.3.3 and 4.3.4.

4.3.2 Removal of Commonly Occurring Feature Values

This experiment studied the training data to determine if any feature value appeared disproportionately more then any other. The semantic type, *podg*, appeared much more than any other feature value (see Section 3.5 for details on the data). Evaluation for this experiment will be done by comparing against a baseline classifier that chooses the most commonly occurring outcome (in Weka, this is the ZeroR classifier) and against the initial design. Evaluation will also include looking at cross validation results and examining confusion matrices.

Removal of "podg" - Comparison to Most Frequently Occurring Class

As with previous experiment described in Section 4.3.1, selecting the most frequently occurring outcome also provides the baseline for this experiment. Table 4.2 shows the original experiment design applied to all of the data and to the subset of data that removed *podg*. These are the top third of the table's left and right sides respectively. The results for this experiment are under the "Most Common Feature Value Removed" columns on the right side of Table 4.2.

The 1980-1984 data set now has only 2135 training instances after removal of *podg* compared with 4997 in the previous "original experiment". The post-1984 data set contains 116,624 training instances, down from 203,323. After removal of *podg*, the performance of all classifiers dropped notably when compared to the previous experiment. For example, on the post-1984 dataset, the J48 classifier dropped from 68.96% to 56.70%. A possible explanation is that the *podg* semantic type was skewing the classifier training since it appeared so disproportionately more than any other data value – and it occurred across all three semantic type feature (A, B and C semantic types). The *podg* semantic type was generating noise.

The 1980-1984 data dropped in percent classified accurately more drastically when compared to the post-1984 data. For example J48 dropped from 65.44% to 47.45% – a drop of almost 18% – in the 1980-1984 data set. J48 dropped from 68.96% to 56.70% in the post-1984 data set – a drop of only 12.26%. This could be because the 1980-1984 data set is relatively small (2135) as described in Chapter 3 in Table 3.14, and there are quite a few outcomes to predict (67) out of this data set as shown in Table 3.13.

Despite the lower percent classified accurately with the new data design, a simple comparison of the three classifiers with baselines show that the classifiers (J48, Naïve Bayes and Random Forest) are performing better than the baseline results for this experiment, as was also seen in the original data design experiment. The post-1984 data had only 6.16% classified accurately using the baseline, and in the same data set, 56.70% classified accurately using the J48 algorithm. That's more than a 50% delta and is greater than the best of 39.08% delta between the baseline and J48 in the original experiment results.

Actual	False Positive	J48	Random Forest	Naïve Bayes
ca_PART_OF	ac_LOCATION_OF	x	x	x
ca_LOCATION_OF	ac_PART_OF			x

Table 4.6 Removal of "podg" feature design, 1980-1984, most commonly occurring false positives

Actual	False Positive	J48	Random Forest	Naïve Bayes
ca_PART_OF	ac_LOCATION_OF	x	x	x
ca_LOCATION_OF	ac_PART_OF			x
ca_INTERACTS_WITH	ac_ISA			x
ca_INTERACTS_WITH	ac_INTERACTS_WITH			x
ac_INTERACTS_WITH	ca_INTERACTS_WITH	x	x	x

Table 4.7 Removal of "podg" feature design, post-1984, most commonly occurring false positives

Removal of "podg" - Evaluation Using Confusion Matrices

The next two paragraphs are similar to the confusion matrix summaries presented in the Section 4.3.1 – first the 1980-1984 data is studied and then the post-1984 data is studied.

Confusion matrix summary for 1980-1984 minus "podg" trained classifiers: See confusion matrix summary results in Table 4.6. The actual relationship and direction of *ca_PART_OF* was found to still be related to *ac_LOCATION_OF*, and *ca_LOCATION_OF* was found to be related to *ac_PART_OF* as seen in the initial experiment that used the original data design. However, these are the only significantly occurring false positive. The *TREATS* being related to *ADMINISTERED_TO* found in previous experiment is not found when *podg* is removed from the 1980-1984 training data and classifiers are retrained.

Confusion matrix summary for post-1984 minus "podg" trained classifiers: See confusion matrix summary results in Table 4.7. As was seen in the original classifier design, Naïve Bayes classifier produced more false positives than J48 and Random Forest. Additionally, with this classifier design, *INTERACTS_WITH* produces false positives that might be expected – finding that A interacts with B is very similar and may be the same as saying B interacts with A.

Removal of "podg" - Evaluation Using Significance

As with the previous experiment, this experiment provides a null hypothesis for other experiments. That is, the results from this experiment will be used as the null hypothesis that the regularized outcomes experiment (Section 4.3.3) and the multi-feature ablation study (Section 4.3.4) may be compared against when the regularization is applied to the "Most Common Feature Value Removed" data sets. These are shown on the far right two columns in Table 4.2.

Discussion of Removal of "podg"

Removing the single most commonly occurring feature, *podg*, presented results that were sometimes 50 percentage points better than the baselines (J48 and Random Forest using the post 1984 training set). The removal of *podg* experiment leaves two data sets (1980-1984 and post 1984 sets minus *podg*) with good distributions of the other feature values. These two data sets will be used as the basis of the modified data representations for the additional experiments that regularize outcomes and apply ablation presented in Sections 4.3.3 and 4.3.4. These are the right columns in Table 4.2. These results also are the null hypothesis that the regularized outcomes and ablation experiments will try to reject.

Removing the commonly occurring *podg* semantic type produced a much smaller sets of training data. The 1980-1984 contained only 2135 training data points – less than half the original set. The outcomes being predicted still contain a relatively large number of unique values – 67 for the 1980-1984 data and 106 for the post-1984 data as shown in Table 3.13. The data that omitted *podg* was applied to the original data designs, and it was applied to the regularized outcomes data design and the ablation studies whose results are discussed in Sections 4.3.3 and 4.3.4.

Against the full data set using the original data design (top left side of Table 4.2), the baseline algorithm applied to the *podg*-omitted data set presented much lower accurately classified percentages which means that removing *podg* made it much more difficult for the simple baseline classifier to accurately predict the outcomes. As just noted, there are 67 and 106 possible outcomes for the respective data sets which is quite large – the classifiers should have a more difficult time predicting accurately, the outcomes which is exactly what was seen in this section's results (baseline numbers are first row of the top third, right side of Table 4.2). Just below the baseline row, the results of the other classifiers show that they performed worse than the original design against all of the data (left side of table) dropping from 60 to almost 70 percent range to the mid 40 to mid 50 percent range. J48 still performs with the highest percentages for this experiment.

This experiment may not be a wise or logical one to keep when studying the medical domain with the obvious goal of finding cures to or causes of diseases that afflict people. Removing *podg*, which is the semantic type for "population or disabled group", may be detrimental to work that seeks to find those cures to or causes of diseases. Its removal from the data sets is purely an theoretical exercise in understanding how very commonly occurring feature values may incorrectly influence results and make it harder for classifiers to perform well. Including the results that removed this very commonly occurring *podg* value helps to further prove that the classifiers using the feature designs presented here are able to predict the outcome with much better success than baselines. A concern is that the better performance of the original data design using all data (top left of table) could indicate that including instances with the *podg* value is swaying the results in a positive direction and, without *podg*, the classifiers perform much worse. Section 4.3.3 will continue this discussion.

4.3.3 Regularization of Outcomes

This section describes the results from using a redesigned set of data where the A to C relationship and direction being predicted is normalized to always assume the direction of A to C. Details of this data design are described in Section 3.5.2. The biggest difference with this data set is that there are only 42 and 55 outcomes to predict in the 1980-1984 and the post-1984 data sets, respectively. This is down from 68 and 107 possible outcomes in the original data designs and down from 67 and 106 after removal of *podg* in the previous experiment (see Table 3.13 for these summaries). The hypothesis is that having fewer outcomes to predict should allow the classifiers to be able to better predict outcomes.

Evaluation of this experiment will be performed using three methodologies. First, evaluation will be done by comparing baseline results to the other classifiers' cross-validation results. Second, evaluations will be done by comparing deltas and by examination of statistical significance against the previous two experiments and their types of data – the original experiment design using all of the data and the data that removed *podg*. Deltas will be examined by comparing against the two previous experiments and against baselines of the most frequent occurrence classifier. The last evaluation methodology will be a study learning curves produced by three of the classifiers and the two primary data sets (1980-1984 and post-1984).

Outcome Regularization - Comparison to Most Frequently Occurring Class

The center rows of Table 4.2 show the results of 10-fold cross-validation using the baseline classifier and the three other classifiers (J48, Naïve Bayes and Random Forest) for this experiment. The left side of the table used all data for training and evaluation. The right side used the data that removed occurrences of *podg*. The hypothesis of this experiment was that better classifier performance should be able to be obtained if the number of outcomes is reduced. This is, indeed, the result achieved with all of the classifiers. The baselines, shown in the first row of the middle section of Table 4.2, improved compared with the previous experiments' baselines that used the all of the data or that removed *podg* (top section, left and right side of Table 4.2). J48 still performs the best of the three classifiers. All classifiers performed much better when compared against the regularized outcomes baseline that selected the most frequently occurring outcome.

The older 1980-1984 data set did not perform quite as well as the post-1984 data with J48 and random forest algorithms, but as noted previously, the older set is much smaller than the post-1984 set. With all data sets and classifiers of this experiment, the delta between the baselines and the other classifiers is still 40 to over 50% indicating good success in being able to predict the outcomes using this data design with the three classifiers used.

The results for the data set that removed *podg*, shown in the center right side of Table 4.2, produced successful predictions more in line with the original data design against all data (top left of Table 4.2). The results using the regularized outcomes data design compared against removing instances with the most commonly occurring feature value are in 10 to 15% better than the original data design results. However, the performance still lags behind the full data set using regularized outcomes by 10% or more. As noted in the previous section, there is importance in understanding information about population groups when trying to find cures to diseases. Therefore, removing instances containing the most commonly occurring feature value of *podg* is probably not wise. The results of removing instances with the commonly occurring feature value are of theoretical interest for understanding classification and data design performance on solving the problem of explaining LBD relationships but may remove important semantic information.

Outcome Regularization - Evaluation Using Significance

The results from this experiment were studied more closely for significance using hypothesis testing. The basic hypothesis is that the previous experiment (Original Experiment Design) is the null hypothesis and this alternative experiment seeks to reject that experiment's results.

Table 4.2 summarizes the results and provides comparisons for the original data set where all data is used (left side of table results) and the data set that removed *podg* (right side of table). In all cases of the non-baseline classifiers, the Regularized Outcome experiments, shown in the center third of the table, performed significantly better than their Original Experiment Design counterparts in the upper third of the table. The actual t-scores were all less than 0.00001 and in reality were all less than 1×10^{-10}.

Outcome Regularization - Evaluation Using Learning Curves

Learning curves were generated by holding out varying percentages of the training data, training on the rest and plotting the classifier performance. The learning curves, shown in Figures 4.1 and 4.2, show that the classifiers are able to improve their learning with more data points used in training. The post-1984 training set produces better results for the J48 and Random Forest classifiers and, with both training sets, Naïve Bayes classifier begins to lag behind with lower percentages classified accurately. The plots shown in Figure 4.2 focus in on the elbow of the learning curve chart. The somewhat more gradual bend in the post-1984 data set, which is a much larger data set, may show that it takes more data before the classifiers are more consistent at predicting the outcomes. Most likely cause is that the larger data set has more instances of some of the more infrequently occurring feature values. The 1980-1984 training set required at least 1000 training instances versus more than 2500 before the post-1984 training set results leveled off at approximately 70% classified accurately. The much larger post-1984 data set continued to improve in its ability to predict outcomes until at least 30 to 40% of the data were used to train (over 60,000 data points).

Discussion of Outcome Regularization

Reducing the number of outcomes has proven to produce significantly better results than the previous experiments. The significance is based on hypothesis testing using Student's t-test calculations described in Section 4.1. The outcome simplification of 67 possible outcomes down to 42 in the older 1980-1984 data set, and of 106 possible outcomes down to 55 in the post-1984 data set, likely contributed to the improved performance of the classifiers. Another interesting observation is that the 1980-1984 results with outcome regularization data design did not seem to be quite as significantly better than the post-1984 set. This could have been because 42 is 63% of 67 for the 1980-1984 data whereas 55 is 52% of 106 for the post-1984 data. That is, the reduction of outcomes was a higher percentage more for the post-1984 data set when outcomes were simplified to have only the A to C direction.

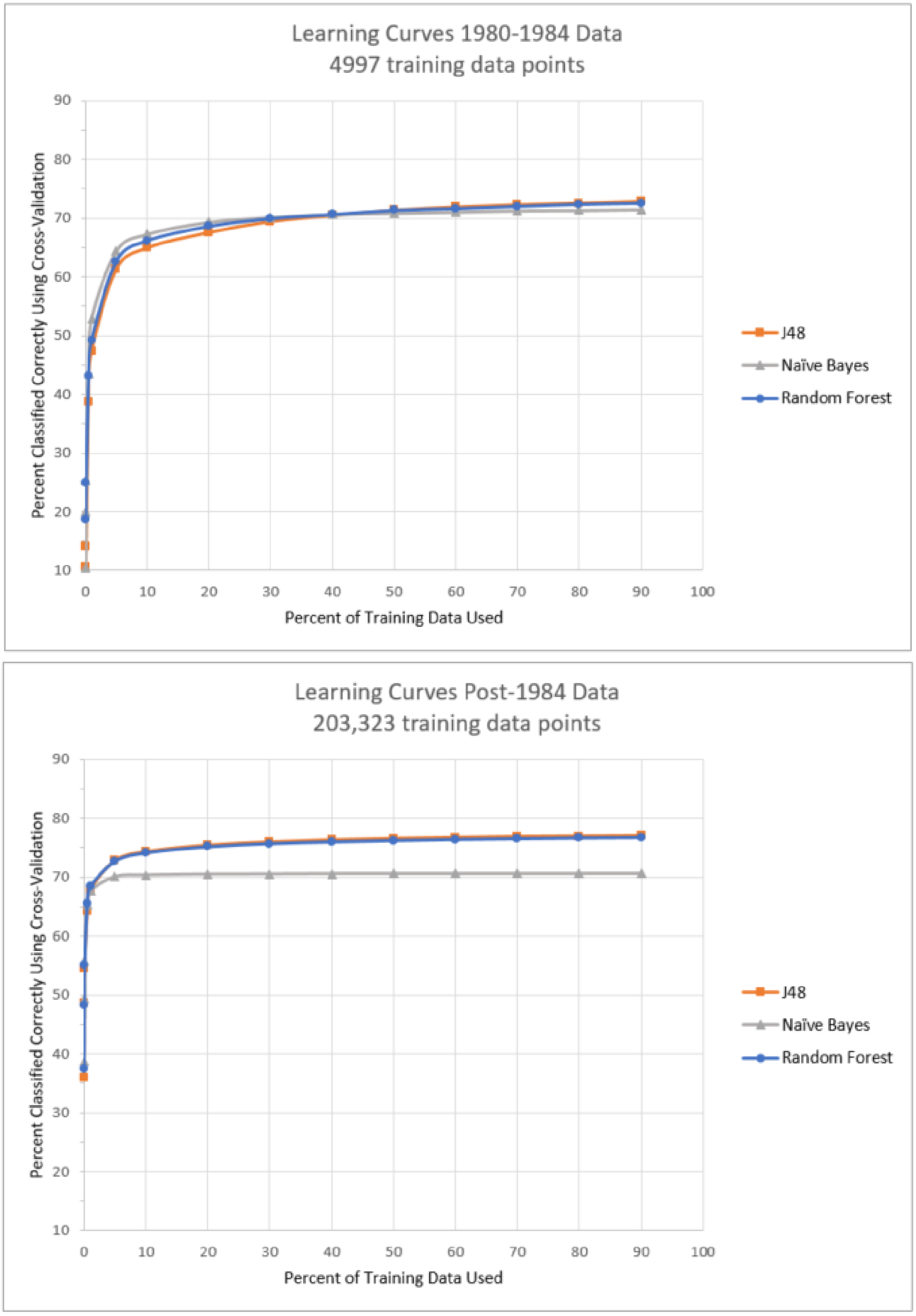
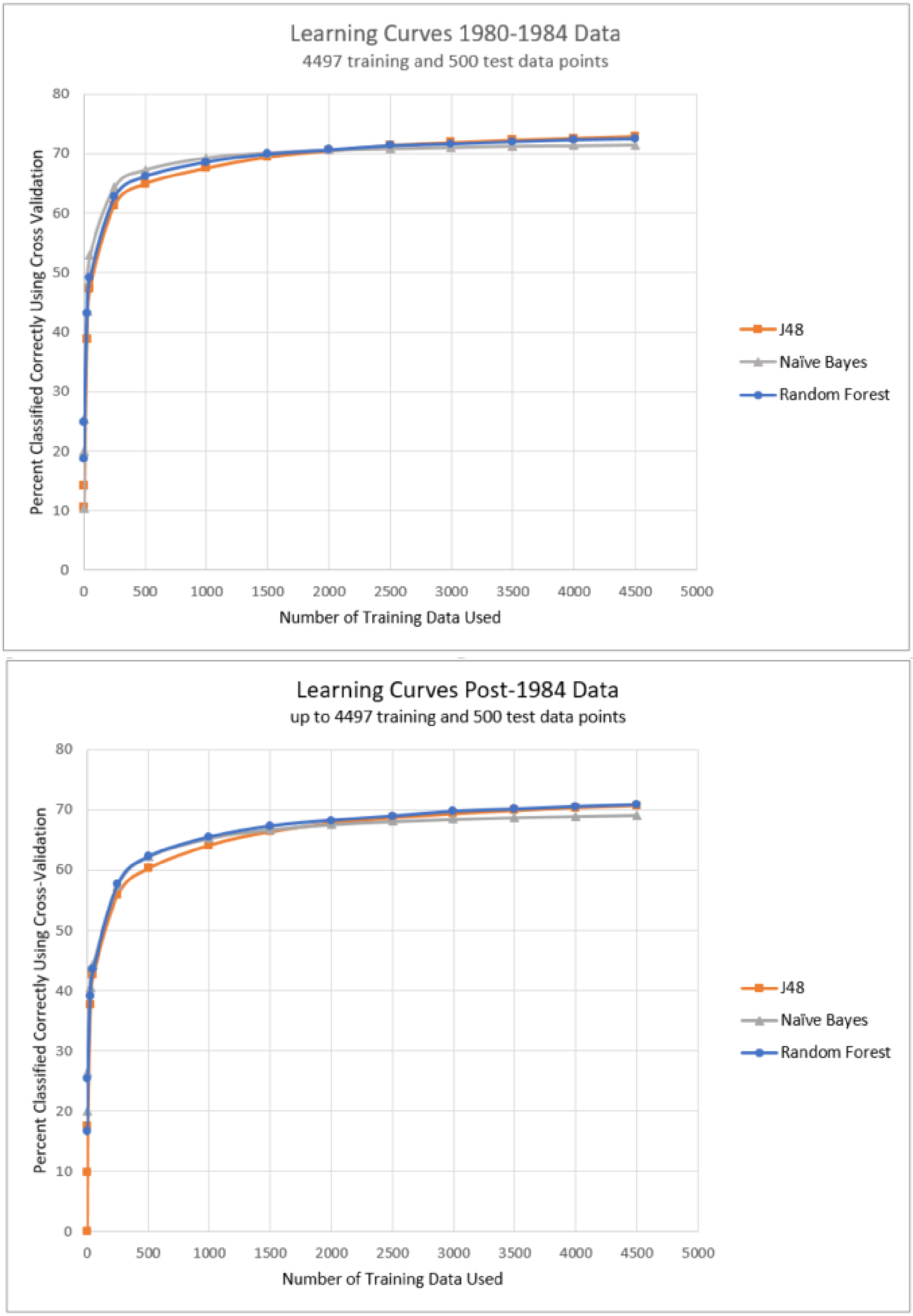


Fig. 4.1 Learning Curves Plotted by Percentages



Learning Curves 1980-1984 Data

4497 training and 500 test data points

Learning Curves Post-1984 Data

up to 4497 training and 500 test data points

Fig. 4.2 Learning Curves Focused on First 4500 Points of Data

The J48 classifier produces the best performance when all data is considered (left columns of results in Table 4.2). The 1980-1984 set producing 73.10% classified accurately and the post-1984 set producing 77.13%.

This experiment and its data design will be used as the null hypothesis which the multi-feature ablation experiment (Section 4.3.4) will attempt to reject. The basic data design resulting from the outcome regularization also forms the basis on which the ablation experiments will be built. They will use both the "All Data Used" or the "Most Common Feature Value Removal" sets of data.

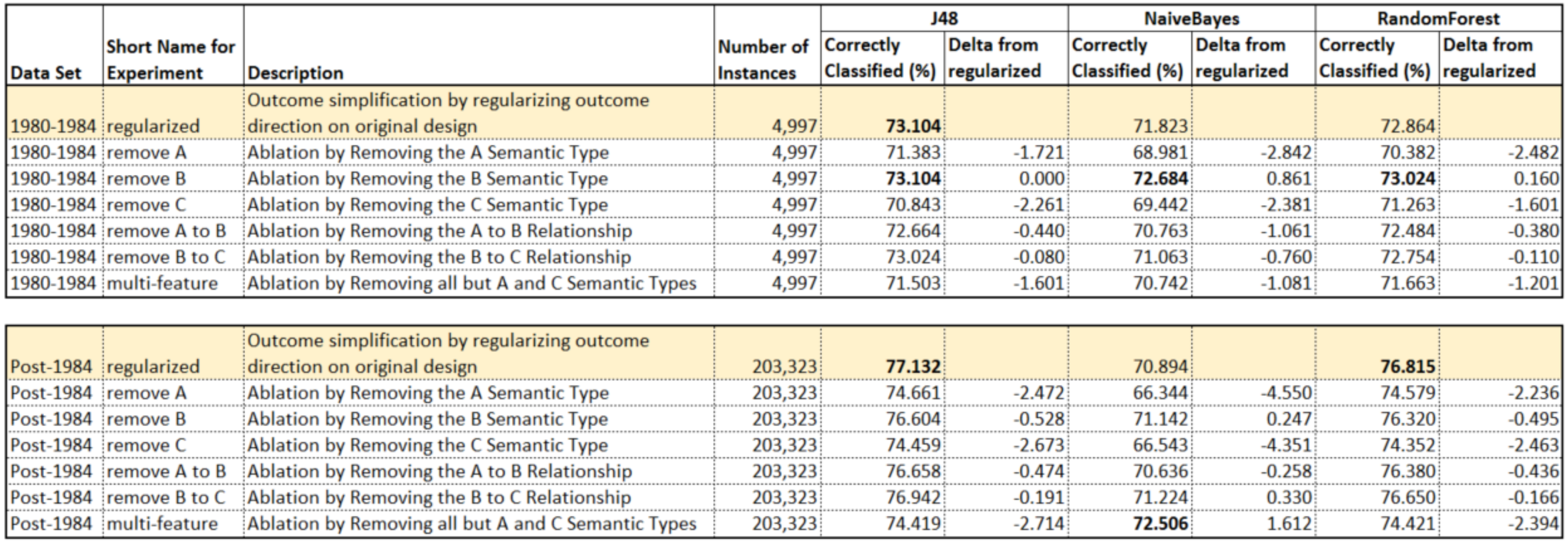
4.3.4 Experiments with Ablation

This experiment looks at single and multi-feature ablation (described in Section 4.1). Ablation results will be evaluated using simple deltas from the outcome regularization results (Section 4.3.3), and then by using significance testing against the outcome regularization experiment results for the corresponding data sets ("All Data Used" or "Most Common Feature Value Removal"). Note that the single and multi-feature ablation experiments were performed on the set of data that removed instances containing the most commonly occurring feature value of *podg*. The results for these are summarized in Table 4.2 in the right two columns. However these results are not discussed in this section, other than to say that very similar results were seen as were seen for the "All Data Used" ablation experiments.

Single Feature Ablation

This section will discuss the systematic removal of each feature and analysis of classifier performance after this ablation is applied to the training data. The purpose is to identify those features that contribute to or hurt the performance of the classifiers. Table 4.8 shows the results of systematically removing each feature from the outcome regularization experiment described in Section 4.3.3.

The regularization of outcomes experiment is the baseline to which each ablation experiment compares. Simple deltas are being used to compare each ablation with the baseline. A negative number in the delta columns indicates that the classifier was not able to perform as well when this feature was removed. This also indicates a significant contribution from that feature in the data designs – especially if the delta was "big" when compared to other ablation results.



Data Set	Short Name for Experiment	Description	Number of Instances	J48 Correctly Classified (%)	J48 Delta from regularized	NaiveBayes Correctly Classified (%)	NaiveBayes Delta from regularized	RandomForest Correctly Classified (%)	RandomForest Delta from regularized
1980-1984	regularized	Outcome simplification by regularizing outcome direction on original design	4,997	**73.104**		71.823		72.864	
1980-1984	remove A	Ablation by Removing the A Semantic Type	4,997	71.383	-1.721	68.981	-2.842	70.382	-2.482
1980-1984	remove B	Ablation by Removing the B Semantic Type	4,997	**73.104**	0.000	**72.684**	0.861	**73.024**	0.160
1980-1984	remove C	Ablation by Removing the C Semantic Type	4,997	70.843	-2.261	69.442	-2.381	71.263	-1.601
1980-1984	remove A to B	Ablation by Removing the A to B Relationship	4,997	72.664	-0.440	70.763	-1.061	72.484	-0.380
1980-1984	remove B to C	Ablation by Removing the B to C Relationship	4,997	73.024	-0.080	71.063	-0.760	72.754	-0.110
1980-1984	multi-feature	Ablation by Removing all but A and C Semantic Types	4,997	71.503	-1.601	70.742	-1.081	71.663	-1.201
Post-1984	regularized	Outcome simplification by regularizing outcome direction on original design	203,323	**77.132**		70.894		**76.815**	
Post-1984	remove A	Ablation by Removing the A Semantic Type	203,323	74.661	-2.472	66.344	-4.550	74.579	-2.236
Post-1984	remove B	Ablation by Removing the B Semantic Type	203,323	76.604	-0.528	71.142	0.247	76.320	-0.495
Post-1984	remove C	Ablation by Removing the C Semantic Type	203,323	74.459	-2.673	66.543	-4.351	74.352	-2.463
Post-1984	remove A to B	Ablation by Removing the A to B Relationship	203,323	76.658	-0.474	70.636	-0.258	76.380	-0.436
Post-1984	remove B to C	Ablation by Removing the B to C Relationship	203,323	76.942	-0.191	71.224	0.330	76.650	-0.166
Post-1984	multi-feature	Ablation by Removing all but A and C Semantic Types	203,323	74.419	-2.714	**72.506**	1.612	74.421	-2.394

Table 4.8 Summary of Feature Ablation Experiments (highlighted rows indicate the results from regularized outcomes experiment)

The results in Table 4.8 show that A and C semantic types consistently produce bigger negative deltas when compared to the other features. For example, in the post-1984 data for the J48 classifier, the A and C semantic type ablation produced -2.472 and -2.673 deltas when compared against the baseline. The other single feature ablations produced results much closer to zero. This indicates that A and C have greater contributions to the success of the classifiers' ability to predict outcomes, accurately.

Naïve Bayes classifier shows the same trend where A and C semantic types seem to be significant contributors to the success of the classifier in predicting the outcome. In the post-1984 data, the numbers show less contribution from the other features based on the positive deltas they show. This may indicate that there are feature dependencies in the data designs which would cause the poorer performance using naïve Bayes classifiers compared to the decision tree-based classifiers.

Multi-Feature Ablation

Based on the results of removing each feature, one at a time, the A and C semantic types stood out as contributing most to the outcomes. Another logical experiment is to try to train classifiers based on a feature design that only included the A and C semantic types for training. The last row in each of the two sections of Table 4.8 show the results of this multi-feature ablation test that was applied to the outcome regularization design.

The ablation that removed all but the A and C semantic type was able to classify with reasonable success. The 1980-1984 data sets indicate that this would be all that is needed to classify and predict the outcome, because all three classifiers performed better using only the A and C semantic types for training. However, the post-1984 data sets show lesser performance with the decision tree classifiers. Naïve Bayes classifier shows an interesting result with the multi-feature ablation – it actually performs better with only considering the A and C semantic types as features and ignoring all of the other features. This supports the observation in the previous subsections on single feature ablation (first part of this Section 4.3.4). The features may not be totally independent and, therefore, a prerequisite for naïve Bayes classification is not met.

Multi-Feature Ablation - Evaluation Using Significance

This experiment was also studied using hypothesis testing just as the outcome regularization experiment was. The purpose was to see how the multi-feature ablation results compare against the outcome regularization results. Table 4.2 presents the high level results using *, o, and - to indicate significantly better, unable to reject null hypothesis, and significantly worse, respectively. The experiments that used the original data designs and the data that removed *podg* were summarized for the three primary classifiers. All trials performed worse except the naïve Bayes results using the post-1984 data set. The naïve Bayes classifier performed significantly better. As with the simple ablation experiment comparisons presented in the previous parts of this section, there may be a lack of independence of the features. Naïve Bayes classification assumes independence of the features being classified. The ablation studies are able to indicate that there is some interaction between the features.

Summary of Ablation Experiments

The single feature ablation experiments shows that the A and C semantic types contribute most to the successful classification results. These results led to the "Ablated Leaving only A and C" experiment set that is presented in Table 4.2. Using tree-based classifiers like the decision tree of J48 classifier and the random forest classifiers produces less interesting results when applying multi-feature ablation to the training data. The naïve Bayes classifier, however, produced better results using the multi-feature ablation of just the A and C semantic types when applied to the much larger post-1984 data set. The other features in the current data designs may not be independent of each other and, thus, may be causing the improved performance of naïve Bayes classification.

4.4 Chapter Summary

To summarize, this chapter has presented feature designs based on training data, or gold standard data, derived from SemMedDB. After a brief examination of multiple classifiers, three primary classifiers were selected to provide a cross-section types (J48, Naïve Bayes and Random Forest) and the baseline classifier which selects most commonly occurring outcome was used to baseline classifier performance. Removal of the frequently occurring *podg* semantic type improved the accuracy when compared with the baselines but produced lower accurately classified results than regularizing the outcomes. The experiment to regularize the outcomes turned all of the A to C and C to A relationships to be normalized to always be in the A to C direction and appropriately rotated the triples. With the outcome regularization, the number of outcomes to predict was reduced by almost half in the post-1984 data set and by two thirds in the 1980-1984 set. The classifiers were able to perform significantly better using the outcome regularization design when compared with the original data designs.

Ablation experiments helped to indicate that the A and C semantic types were different and contributed the most to predicting outcomes. Ablation studies using the naïve Bayes classifiers also indicate that the features in the data designs are not independent. Results from other classifiers beyond the primary three indicate that the accuracy provided by them was similar to that of the primary three classifiers.

The most important conclusion from this chapter is that explaining the A to C relationship using classification on training data as presented in this chapter produces predicted results that are significantly better than the chosen baseline. Also, improvements to the classifier designs that reduced the number of outcomes by normalizing to only the A to C direction of that relationship also significantly improved the ability for the classifiers to predict the nature of the relationships. The J48 classifier performed best on both the 1980-1984 and the post-1984 data sets using the regularized outcomes experiment design. The hypothesis is that any of the primary three classifiers studied with the regularized outcomes data design including all data (not the data design that removed *podg*) should be able to accurately predict the LBD relationships over 70% of the time.

Chapter 5

Identifying and Explaining Hidden Knowledge

This chapter presents experiments that apply the classifiers developed in Chapter 4 to the task of explaining LBD relationships. To do this, candidate LBD pairs must first be identified along with linking B terms and the nature of the A-B and B-C relationships. That is, partially qualified triples (Section 2.1.3) must be identified for candidate LBD pairs. It will be these partial triples that will be presented to the classifiers in an attempt to explain the nature of the hidden knowledge between the A and C related concepts identified with LBD techniques. Once classifier predictions of the nature of the A to C relationships are available, further validation is performed to determine if there is any supporting evidence that the predictions are accurate (Section 2.3).

The classifiers, developed in Chapter 4, were based on the medical domain using concepts identified from MEDLINE sources. The LBD will also be based on the medical domain and will use MEDLINE abstracts as the corpus from which hidden knowledge will be extracted. The approach uses Semantic Vectors for identifying the LBD candidate pairs and uses SemMedDB for explaining the A-B and B-C relationships of the partial triple. Semantic Vectors has been used by Cohen et al. (2012) to associate concepts indirectly. In their work, they point to LBD as a result of indirect inference. Cameron et al. (2015) suggested using MEDLINE and MeSH terms to provide links to candidate LBD pairs. However, based on a survey of available literature, the combination presented here of using MEDLINE abstracts, Semantic Vectors software and SemMedDB data to identify candidate LBD pairs and ultimately form complete partial triples is novel. Additionally, using SemMedDB as a source of expert knowledge to validate the predicted A to C relationships is also novel.

The general steps presented in this chapter are as follows:

1. Identify a data for LBD, for submission to classifiers and for validation of predicted LBD explanations.
2. Generate LBD candidate pairs using statistical methods on older time-slice of the corpus
3. Prepare the candidate LBD pairs for classification by building them into partially qualified triples
4. Predict the explanation of the LBD discoveries using classification
5. Validate the predicted explanations using explanations found in newer data
6. Refine and improve on the prediction accuracy by applying ensemble learning

Section 5.1 discusses the data used for LBD, classification and validation of results. The corpus used for LBD is comprised of MEDLINE abstracts which are derived from medical journal articles and other publications (Section 2.4.1), and the concepts used are derived from lists found in MEDLINE/PubMed Baseline Repository (Section 3.1). Section 5.2 discusses the process of LBD performed using the Semantic Vectors package. Section 5.3 disscusses the preparation of the pairs for LBD by using SemMedDB (introduced in Section 2.4.2) to provide the additional information needed for the completion of the partial triples. That section also describes the use classifiers to provide possible explanations of the candidate LBD pairs. Classifiers used are those trained on the regularized output design (design presented in Section 3.5.2 and results in Section 4.3.3). Section 5.4 discusses applying the time slice methodology (introduced in Section 2.3) as a way to validate the classifier predictions. This is accomplished by comparing the classifier explanations for older candidate LBD candidate to actual relationships found in newer publications. That is, explanations for candidate LBD pairs from the 1980-1984 time slice are checked against newer references found in SemMedDB and are either confirmed or contradicted. Section 5.5 discusses the last experiments presented in this chapter that try to improve on the results by applying ensemble learning techniques (introduced in Section 2.8) to more accurately predict the LBD explanations.

Ultimately, this chapter focuses on 312,426 partial triples that were identified from an initial set of 17,799,335 candidate LBD pairs and then 6,119,954 pairs that had linking B terms. The partial triples are those that have linking B terms with identified relationships between A and B and B and C nodes. This set of 312,426 partial triples is passed to the classifiers so they may suggest explanations of the nature of the LBD relationship in the set. Then the predicted results are compared against data found in newer data to see if any predictions may be validated or contradicted. Ensemble learning provides a way to possibly increase classification accuracy at explaining LBD candidate pairs.

5.1 Data for LBD, Classification and Validation

This section presents the data used in the experiments presented in the rest of this chapter. There are three parts to these data – data used to perform LBD (Section 5.1.1), data presented to classifiers (Section 5.1.2), and data used to validate the explained LBD relationships (Section 5.1.3). The data for LBD consists of a corpus built from MEDLINE abstracts and a set of normalized concepts providing candidate A, B and C nodes of LBD. The data presented to classifiers includes the LBD candidate pairs and linking B terms. The data used for validation is derived from a source of facts (SemMedDB) and is focused on the candidate LBD pairs. The next sections provide more details about these data.

5.1.1 Data for LBD

This section presents the data used to perform LBD – it includes the selection and preparation of the corpus. LBD requires a set of concepts to initiate open or closed LBD (Section 2.1.2) and requires a set of documents (the corpus). For this research, the MEDLINE corpus was selected because it is freely available for research, and because it is has been used in other LBD research (see Section 2.4). There are two primary steps to prepare the corpus for LBD and for the experiments presented here: the time slice selections and the preparation of the documents for LBD. Preparation of the corpus involves normalizing both the concepts and the corpus so LBD may be performed using statistical methods.

Slicing the Corpus

Time slicing the corpus, as introduced in Section 2.3, allows for validation of automatically suggested relationships without the need for judgments by experts (Yetisgen-Yildiz and Pratt, 2009). The corpus is split into two sets, an older set and a newer set. The newer set of documents is set aside for use in validating the LBD discoveries and, more importantly, in validating the actual relationship type suggested by the classifier models that are generated for this research. More on how suggested relationships may be validated using the time slice approach is presented in Section 5.4.

Swanson's initial studies on Raynaud's phenomenon focused on a corpus from 1975 through mid 1985 (Swanson, 1986a), while Kostoff's studies used various date ranges depending on the disease being studied (Kostoff et al., 2007). Swanson focused on other date ranges in subsequent works (Swanson, 1988; Swanson and Smalheiser, 1996; Swanson, 2011). To use the time slice method for validating results, the only constraint is that LBD is performed on older data and that validation uses newer data. In this work, abstracts published between 1980 and 1984, inclusive, are used as the older time slice which include some of the dates used by Swanson's original work on Raynaud's disease.

Based on the concepts used in the studies presented in Kostoff et al. (2007), additional concepts beyond those from the Swanson studies of Raynaud's disease and migraines were included. The additional concepts include Parkinson's disease, multiple sclerosis and cataracts. This makes the dates of 1980-1984 less significant for reproducing Swanson's discoveries, but still allows a time slice evaluation methodology to be used. A total of 692,399 abstracts from 1980-1984 are used and were retrieved from the MEDLINE 2002 baseline (introduced and the older 2002 baseline justified in Section 3.1). LBD will be performed on the older date range and explanations of LBD pairs will be validated against newer data.

Concept and Corpus Normalization

LBD requires knowledge of terms or concepts as described in Section 2.1.2 and the set of concepts considered are introduced in Section 3.1. Concepts may present themselves in the text in many different ways. For example, Raynaud's phenomenon, Raynaud's disease or simply Raynaud's may all refer to the same disease or syndrome – they are synonyms for the same basic concept. In order to use statistical co-occurrence techniques, these synonyms must be normalized so that a single common term is used throughout the corpus to refer to the concept and all of its synonyms. This way, a document referring to Raynaud's *phenomenon* will be found to be referring to the same concept as a document referring to Raynaud's *disease*. This single common term may be a common word or phrase like Raynaud's disease or may be some unique identifier for each concept set like a GUID or a database key value. This research uses the unique identifier approach. When using this approach, the system maintains a mapping back to the concept to which the identifier relates, in order to allow for humanly readable results.

The UMLS Metathesaurus and the MetaMapAPI, introduced in Section 2.4.1, are used to find matching MEDLINE unique identifiers (CUIs) in the MEDLINE data sets along with synonyms for the concepts (MEDLINE, 2002). For example, "multiple sclerosis" has CUI C0026769 but is listed in the Metathesaurus with the synonym of "MS gene" (with corresponding CUI, C1417326). And, in another example, "Parkinson's disease" has CUI C0030567 with two synonyms of "Parkinson's disease pathway" (C1521736) and "Parkinson disease (allelic variant)" (C2681933). Since the statistical methods for performing LBD use a bag of words approach when identifying relatedness, all synonym CUIs are used as representations of the concepts. That is, when "multiple sclerosis" is found in the text, it is replaced with both CUIs: "C0026769 and C1417326"; and "Parkinson's disease" is replaced with its CUI plus two synonyms: "C0030567, C1521736, C2681933". One approach could have used only the preferred CUI presented in the Metathesaurus but this may have omitted some possible discoveries. Having all synonyms in the document will allow links to be made to documents mentioning, for example, either multiple sclerosis or MS gene. The corpus is normalized by replacing all occurrences of the concepts in the corpus with the CUI or the set of CUIs representing the concept. Including all synonyms of a concept also helps avoid any inaccuracies that MetaMapAPI may introduce when it doesn't link all synonyms to all variations of a concept, for example, if MS gene doesn't mention multiple sclerosis in its Metathesaurus entry, links to multiple sclerosis related documents may be lost.

To summarize, the normalization of the corpus involves replacing the actual concept name with the CUI or CUIs representing the concept. More than one CUI is used if synonyms were identified via the Metathesaurus. In order to be able to reference human readable names for concepts, a master list is maintained that maps the concept name to its related CUIs. This is how the results ultimately presented, here, rarely reference the CUIs.

5.1.2 Data for Classification

This section presents the data used for preparing LBD candidate pairs for classification to explain the A to C relationships. Recall that the classifiers developed in Chapter 4 required the semantic types of the A, B and C concepts along with the nature and direction of the A to B, the B to C and the C to A relationships. The C to A relationship is the outcome of the classifiers – the predicted class. In this chapter, using classification to explain LBD relationships requires partially complete triples for the candidate LBD pairs along with their linking B terms. Therefore, the natures of the relationships of the A to B and B to C nodes of the LBD triples must be identified (Section 2.1.3).

```
Require: consider the candidate LBD pairs
for each A concept in a candidate LBD pair do
    query SemMedDB for relationships between A and any B
    for each B concept with A to B relationship do
        query SemMedDB for relationships between B and any C

        if relationship found between B and C then
            if all relationships are from 1980-1984 then
                save as a partial triple
            else
                ignore
            end if
        end if

    end for
end for
```

Fig. 5.1 Completing LBD partial triples from SemMedDB

As was done in Section 3.2, SemMedDB is used to provide candidate B linking terms and to provide the facts about the A to B and B to C relationships. Figure 5.1 shows the logic used to query the SemMedDB and collect completed partial triples. The logic only queries SemMedDB for the A to B relationships and the B to C relationships – not the C to A as was done in the previous chapter. The queries also are constrained to ensure that explained A-B and B-C relationships are from literature in the 1980-1984 date range.

The natures of the relationships of the candidate LBD pairs are predicted by the best classifier design developed in Chapter 4. That design is the one that regularized outcomes to always be in the A to C direction (Section 4.3.3). This also means that the predicted relationship will always be in the direction from A to C.

5.1.3 Data for Validation

This section presents the data used for validation of the relationship explanations provided by the classifiers for the LBD candidates pairs. Section 2.3 presented four ways in which explanations to LBD discoveries may be evaluated and validated. One of those methodologies is expert validation which is also what the time slicing approach simulates. To simulate validation by experts, a set of gold standard data is generated using newer data than that used in the identification of the candidate LBD pairs. The gold standard, being sought to simulate the expert knowledge, is any documented explanation found in post-1984 data, and the explanation is for the literature based discoveries identified in the older 1980-1984 time slice. The newer time slice explained relationships are then used to determine if the suggestions from classifiers can be validated. Yetisgen-Yildiz and Pratt (2009) presented a technique (discussed in Section 5.1.1) that is similar to the approach used here. SemMedDB is used as the source of newer facts. These facts provide explanations of the pairs that LBD identified and are used to try to validate or disprove the predicted explanations suggested by classifiers to explain the candidate LBD pairs.

Queries to SemMedDB are performed that search for relationships between the LBD candidate pair – the A to C relationship. Identifying SemMedDB relationships from documents published after 1984 represent the experts' explanations for the A to C relationships. A total of 70,795 A to C relationships from the 312,426 partial LBD triples were found in newer literature in SemMedDB.

5.1.4 Summary of Data

This section provides a summary of the data that will be used in work presented in this chapter for performing LBD, for classification that will explain LBD relationships and for validation of the explanation predictions. A total of 692,399 abstracts in the 1980-1984 date range will be studied for LBD. 15,427 concepts will be considered in performing LBD. After normalization of the corpus and applying statistical relatedness (Section 5.2) a total of 18,848,172 statistically related concepts will be identified from the corpus using the concept list. After LBD candidates are separated from co-occurring pairs, 17,799,335 LBD candidates were identified. Of these, 6,119,945 had candidate linking B terms (Section 5.3). Queries to SemMedDB were used to prepare the candidate LBD pairs for classification by providing a source for explaining the known relationships between the A-B and B-C node pairs. This resulted in completion of partial triples for 312,426 of the LBD candidate pairs (also in Section 5.3). This data source, SemMedDB, was also queried to provide 70,795 validation data points from the newer time slice of post 1984 documents (Section 5.4).

Here are the data summaries:

- Documents (Abstracts) - 692,399 from 1980-1984
- Concepts - 15,427 unique concepts
- Related concept pairs identified using Semantic Vectors - 18,848,172
- Same document mention pairs (not LBD) - 1,048,837

- LBD candidate pairs - 17,799,335
- LBD pairs with linking B terms - 6,119,954
- Partial LBD triples with explained A-B and B-C relationships - 312,426
- Relationships found in newer literature used to validate LBD explanations - 70,795

5.2 LBD Using Statistical Methods

LBD is performed here using reflective random indexing, a variation of LSA, as a statistical co-occurrence technique and the general relatedness of documents is based on words found in them (see Section 2.2.3). Sometimes the related concepts are found to be mentioned in the same document, sometimes not. Those never mentioned together in the same documents provide candidate LBD pairs. Those found in the same document provide a source of possible linking B terms. For example, in Swanson's example of Raynaud's disease and fish oil being a literature based discovery, these two concepts would *not* be found in the same documents. However, statistical co-occurrence would find them to be related. Statistical methods may also identify a relatedness between Raynaud's disease and platelet aggregation and between fish oil and platelet aggregation. These two pairs *would* be found to be mentioned in the same documents, so would *not* be candidate LBD pairs, but instead, would provide platelet aggregation as a candidate B linking term. The rest of this section describes how Semantic Vectors, a reflective random indexing implementation, is able to perform LBD (Section 5.2.1) and then presents the LBD results identified by using Semantic Vectors (Section 5.2.2).

Note that there are other ways to perform LBD – for example, SemMedDB could have been used, not only to provide training data (Section 3.2), but it also could have been used to discover candidate A and C concepts related with B terms but never linked together directly. Applying statistical methods using Semantic Vectors was simply a choice made for an approach to identifying LBD pairs.

5.2.1 Semantic Vectors for LBD

The Semantic Vectors approach (see Section 2.2.3) provides a statistical method based on LSA for finding related concepts and was used because it provided improvements in performance over other implementations of LSA by including random projections (Widdows and Cohen, 2010). Once a corpus is indexed using Semantic Vectors, concepts of interest are studied to see which pairs are related more than other pairs based on their scores. Additionally, the pairs are examined to identify which are candidate LBD pairs by never being mentioned together and which are possible B linking terms because they do co-occur in documents. That is, Semantic Vectors or any statistical approach will relate concepts without regard to their co-occurrence in a document, or not. Using Semantic Vectors as a means of performing LBD, or indirect inference for discovery, is discussed in Cohen et al. (2012).

In general terms, Semantic Vectors requires a pointer to the documents (e.g., a reference to the directory location of the MEDLINE abstracts) and some parametric data to guide the process of LSA and random indexing. The two parameters explicitly set in this work were the number of dimensions for the random indexing (Section 2.2.3) and the choice to have vectors of real numbers as opposed to binary values. The Semantic Vectors software recommends dimension numbers in the hundreds when vectors contain real values, not binary (Widdows and Ferraro, 2008; Kanerva et al., 2000).[1] The significance of using random indexing in LSA is that instead of a complete term (*m*) by document (*n*) matrix, a matrix of only *m* x 200, for example, is required. In this work, 692,399 documents (Section 5.1.1) were analyzed by Semantic Vectors with real number vectors and the dimension of size 200. Increasing the dimension causes degradation in performance because the matrix being computed gets larger. Reducing the dimension used by random indexing reduces accuracy of results so some caution is applied and the recommendation of staying in the hundreds for this parameter is honored.

Semantic Vectors assigns a score between minus one and one to pairs of concepts when identifying the relatedness of terms found during the indexing of the documents. The higher the score, the more confident the algorithm is that the concepts are related. When concept pairs have a score very close to or equal to one, the concept pairs are usually very strong synonyms of each other and, in some cases, were the same concept with a different presentation like "urinary incontinence" and "Incontinence, urinary". Other times the concepts were, when presented together, actually separate concepts like "loss" and "heterozygosity" – each separately is a concept unto itself, but "loss heterozygosity" is also a concept, so it is logical that a statistical approach would find them to appear as very highly related (1.0 score). In this work, concept pairs with a score greater than 0.001 were considered. This number was chosen to provide a rather large set of related concept pairs and to provide significant numbers of pairs that included the concepts of interest (Raynaud's disease, Parkinson's disease, multiple sclerosis, migraines and cataracts). A total of 21,075,988 related pairs were identified with scores greater than the threshold of 0.001.

[1] https://github.com/semanticvectors/semanticvectors/wiki and the APIs referenced at this site discuss the parameters that may be configured when using Semantic Vectors

Results showed that interesting LBD candidates that were able to be validated were found in the full range from 0.001 through 1.0.

5.2.2 Identifying LBD Candidate Pairs

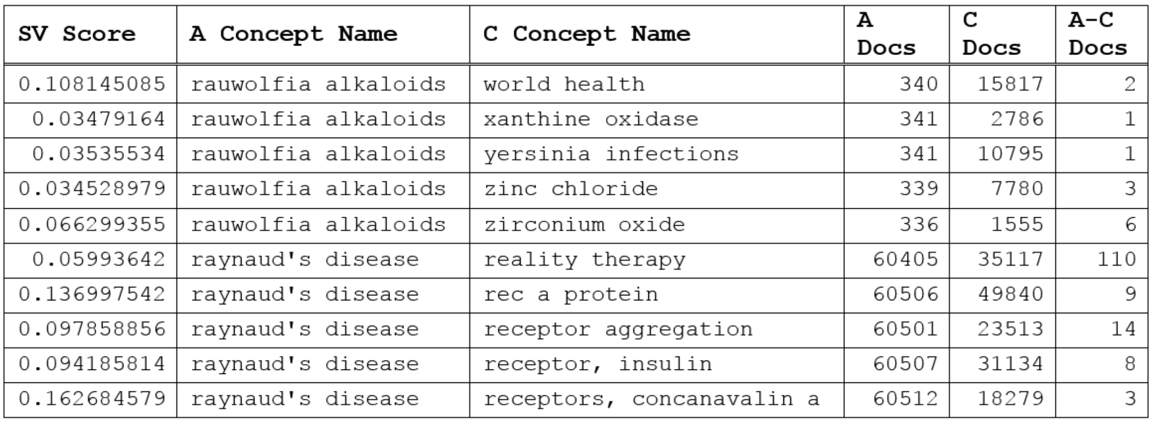
From the set of related pairs identified with Semantic Vectors, candidate LBD pairs must next be identified from those pairs whose score was above the threshold. LBD pairs will be those where the concepts are never mentioned together in the same document. Part of the Semantic Vectors processing includes the creation of indexes that allow efficient retrieval of links to documents containing specified words or concepts. These are the same types of indexes as introduced in Section 2.2.1 that allow efficient retrieval of text data based on search criteria. Candidate LBD pairs are identified by performing a search in the term by document index for each concept of the pair returning a set of documents for each concept. Then the list of documents are compared looking for intersections. If no documents contain both the A and the C concept, then an LBD pair has been identified.

Samples of same document mention pairs are shown in Table 5.1. Samples of LBD Pairs are shown in Table 5.2. (Recall, from Section 5.1.1, that the CUIs were used in the processing that identified related concepts, but for readability, only concept names are shown.) The Semantic Vector score is in the first column followed by the names of the A and the C concepts in the next two columns. The "A Docs" and "C Docs" columns indicate the number of documents that mentioned the respective concept. The last column is the number of documents that mentioned both the A and the C concept.

The data in Table 5.1 suggests, for example, that Raynaud's disease may be related to "rec a protein"[2] because they are mentioned together in the same 9 documents in the 1980-1984 date range. However, this is simply co-occurrence of these two concepts – further investigation is required to determine if there is any real relationship between these concepts that appear in some of the same documents. This is just as would be required if this pair was candidate LBD pair. The data in Table 5.2 suggests, for example, that Raynaud's disease my have a relationship with the hormone, relaxin, but these concepts never appear together in the same document.

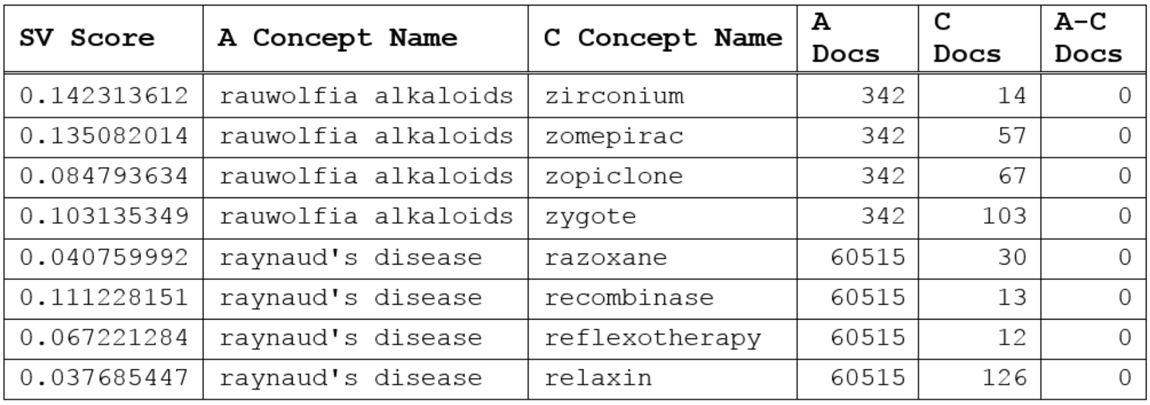
In summary, a total of 18,848,172 pairs of the 21 million related pairs identified in Section 5.2.1 were found using searches into the indexes – this included both LBD and same document mentions. Further study showed that only 1,048,837 of the related pairs contained same document mentions. This left a total of 17,799,335 unique candidate LBD pairs.

[2] "rec a protein" is identified in MEDLINE as a synonym of "Rec A Recombinases", a family of recombinases.



SV Score	A Concept Name	C Concept Name	A Docs	C Docs	A-C Docs
0.108145085	rauwolfia alkaloids	world health	340	15817	2
0.03479164	rauwolfia alkaloids	xanthine oxidase	341	2786	1
0.03535534	rauwolfia alkaloids	yersinia infections	341	10795	1
0.034528979	rauwolfia alkaloids	zinc chloride	339	7780	3
0.066299355	rauwolfia alkaloids	zirconium oxide	336	1555	6
0.05993642	raynaud's disease	reality therapy	60405	35117	110
0.136997542	raynaud's disease	rec a protein	60506	49840	9
0.097858856	raynaud's disease	receptor aggregation	60501	23513	14
0.094185814	raynaud's disease	receptor, insulin	60507	31134	8
0.162684579	raynaud's disease	receptors, concanavalin a	60512	18279	3

Table 5.1 Sample of same document mention pairs where one or more document was found to be in common (identified because A-C docs column has numbers greater than zero)



SV Score	A Concept Name	C Concept Name	A Docs	C Docs	A-C Docs
0.142313612	rauwolfia alkaloids	zirconium	342	14	0
0.135082014	rauwolfia alkaloids	zomepirac	342	57	0
0.084793634	rauwolfia alkaloids	zopiclone	342	67	0
0.103135349	rauwolfia alkaloids	zygote	342	103	0
0.040759992	raynaud's disease	razoxane	60515	30	0
0.111228151	raynaud's disease	recombinase	60515	13	0
0.067221284	raynaud's disease	reflexotherapy	60515	12	0
0.037685447	raynaud's disease	relaxin	60515	126	0

Table 5.2 Sample of LBD Pairs with zero common document mentions (identified because A-C docs column has numbers equal to zero)

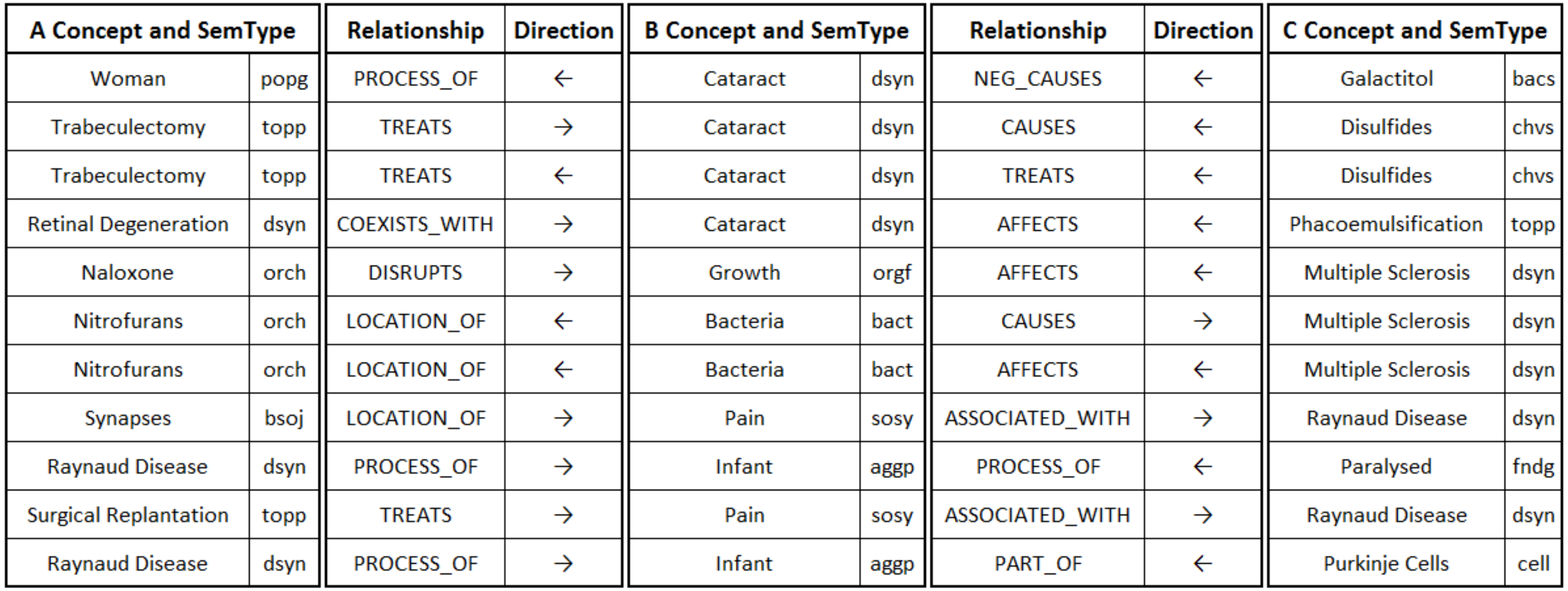
5.3 Explaining Candidate LBD Pairs

This section applies the classifiers designed and developed in Chapter 4 to the task of explaining the LBD A to C relationships. To do this, additional information about the linking B terms and corresponding relationships must be identified. Then the trained classifier models may be applied to explain the LBD pairs.

The goal of identifying linking B terms is to generate partial triples for the candidate LBD pairs so that this data may be passed to the classifiers and the A to C relationship explanations may be predicted. Some B linking terms may be found in the set of same document mentions documents described above (Section 5.2). SemMedDB (see Section 2.4.2 and Section 3.2) was found to be a good source for identifying candidate B linking terms. An advantage to using SemMedDB is that, in addition to the nature of the relationship between the A-B and B-C concepts, the actual sentence from which the relationship was identified is also provided. SemMedDB provided a better approach than brute force searching in documents for linking B terms and then determining the natures of the B term relationships with As and Cs.

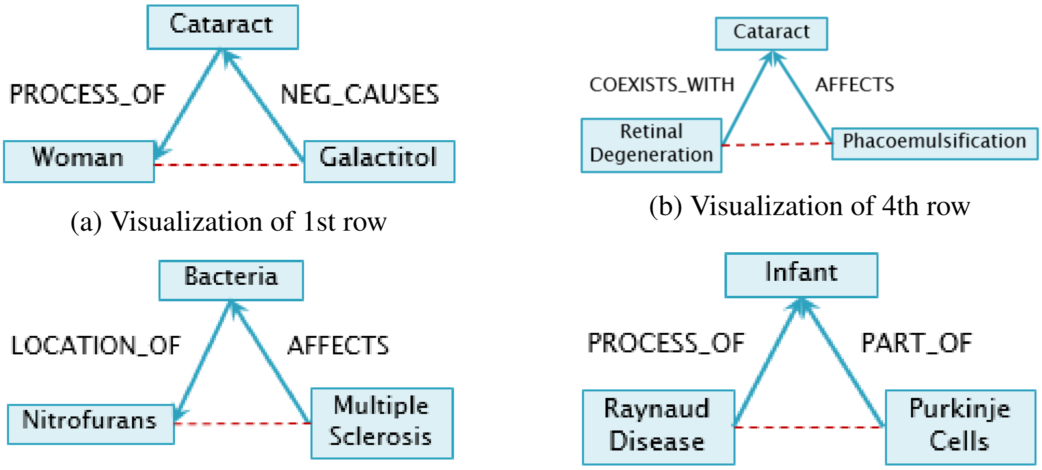
Queries to SemMedDB search for linking B terms between an A and a B and between the same B and a C concept. This approach was presented in Section 5.1.2 and uses the same basic process described in Section 3.2 to query SemMedDB. Once a linking B term is identified, the relationships from the A to B and B to C are retrieved from SemMedDB. They are also checked to make sure the dates of the relationship mentions are before 1985. This is to make sure the relationships were not known during or before the 1980-1984 test range. A list of partially qualified triples (defined in Section 2.1.3) is generated from data meeting the criteria where the relationship between A-B and B-C are those identified from queries to SemMedDB and the date range of the supporting sentence mentions is within 1980-1984.

A total of 6,119,954 LBD pairs with linking B terms were identified. At this point these candidates are not complete partial triples, yet, because the natures of the A-B and B-C relationships has not been identified. This information is retrieved from SemMedDB and includes the semantic types of the concepts and the relationships of A-B and B-C node pairs of the partial triples. A total of 312,426 partial triples were able to be generated. Examples are shown in Table 5.3 where the concepts semantic type (SemType) is the type defined in SemMedDB and MEDLINE and the direction is either to the right or to the left – for example, A to B relationships are shown as an arrow to the right (→) and B to A relationships are shown with an arrow to the left (←). Visualizations of the partial triples listed in Table 5.3 are shown in Figure 5.2. The dashed line represent the A-C LBD relationship that is not



A Concept and SemType		Relationship	Direction	B Concept and SemType		Relationship	Direction	C Concept and SemType	
Woman	popg	PROCESS_OF	←	Cataract	dsyn	NEG_CAUSES	←	Galactitol	bacs
Trabeculectomy	topp	TREATS	→	Cataract	dsyn	CAUSES	←	Disulfides	chvs
Trabeculectomy	topp	TREATS	←	Cataract	dsyn	TREATS	←	Disulfides	chvs
Retinal Degeneration	dsyn	COEXISTS_WITH	→	Cataract	dsyn	AFFECTS	←	Phacoemulsification	topp
Naloxone	orch	DISRUPTS	→	Growth	orgf	AFFECTS	←	Multiple Sclerosis	dsyn
Nitrofurans	orch	LOCATION_OF	←	Bacteria	bact	CAUSES	→	Multiple Sclerosis	dsyn
Nitrofurans	orch	LOCATION_OF	←	Bacteria	bact	AFFECTS	←	Multiple Sclerosis	dsyn
Synapses	bsoj	LOCATION_OF	→	Pain	sosy	ASSOCIATED_WITH	→	Raynaud Disease	dsyn
Raynaud Disease	dsyn	PROCESS_OF	→	Infant	aggp	PROCESS_OF	←	Paralysed	fndg
Surgical Replantation	topp	TREATS	→	Pain	sosy	ASSOCIATED_WITH	→	Raynaud Disease	dsyn
Raynaud Disease	dsyn	PROCESS_OF	→	Infant	aggp	PART_OF	←	Purkinje Cells	cell

Table 5.3 Sample of Partial Triple Data



(a) Visualization of 1st row

(b) Visualization of 4th row

(c) Visualization of 7th row

(d) Visualization of 11th row

Fig. 5.2 Partial triple visualizations of four rows of data from Table 5.3

known in a partial triple. The arrows help to show the direction of the relationship – for example, bacteria is the LOCATION_OF nitrofurans in Figure 5.2c.

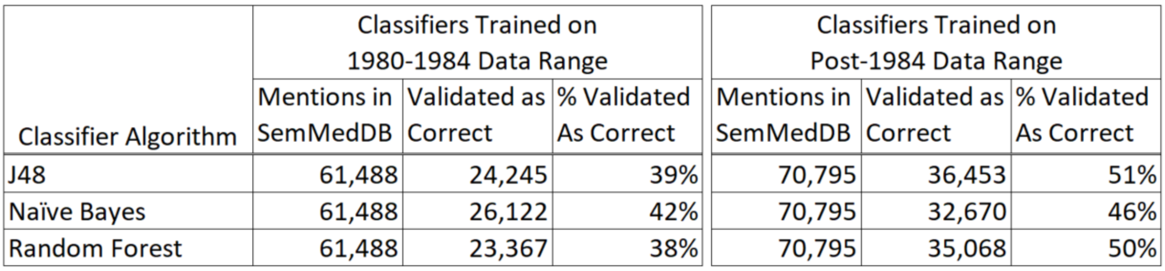
The last step in explaining the candidate LBD pairs is to use the classifiers trained in Chapter 4 to predict the LBD explanations. The classifiers are those based on the regularized outcome data design presented in Sections 3.5.2 and 4.3.3 and the J48, naïve Bayes and random forest classifiers are used in this chapter. This is the practical application of the trained classifiers to automatically explain the literature based discoveries which is the primary purpose of this thesis. The full set of 312,426 partial triples are presented to the classifiers to predict the A to C relationship.

5.4 Validating Suggested Explanations

This section presents the methods applied to validate the predictions of the A to C relationships from Section 5.3. As noted in Section 5.1.3, validation will be done by comparing the prediction to relationships found in SemMedDB for the same concept pairs. Section 5.4.1 presents validation results and Section 5.4.2 explores some of the predictions that were not able to be validated.

5.4.1 Validation of Classifier Predictions

Data for validation was introduced in Section 5.1.3. Recall that SemMedDB was queried for the each of the 312,426 candidate LBD pairs to identify facts for validation. The query is



Classifier Algorithm	Classifiers Trained on 1980-1984 Data Range			Classifiers Trained on Post-1984 Data Range		
	Mentions in SemMedDB	Validated as Correct	% Validated As Correct	Mentions in SemMedDB	Validated as Correct	% Validated As Correct
J48	61,488	24,245	39%	70,795	36,453	51%
Naïve Bayes	61,488	26,122	42%	70,795	32,670	46%
Random Forest	61,488	23,367	38%	70,795	35,068	50%

Table 5.4 Validation of LBD Explanations from Single Classifiers

constrained to allow only 1985 and newer data and to include both A and C concepts. Then the relationship between concepts is captured and used for validation. In total, 70,795 of the 312,426 candidate LBD pairs were found to be mentioned in newer literature. These 70,795 are the only candidates that may now be validated. That is, these are the only candidates with explanations in newer literature. The remaining LBD candidates may still be valid discoveries but cannot be confirmed using the time slice technique using SemMedDB. To validate them would require traditional steps like seeking validation from experts or by clinical trials as discussed in Section 2.3.

For an explained LBD relationship to be considered accurate, it must be mentioned in the newer documents and must match one or more mentions. An LBD relationship may appear in the newer literature five or ten times, for example. The criteria used here is that one or more of the relationships identified must match the predicted LBD explanation to be considered a validated match.

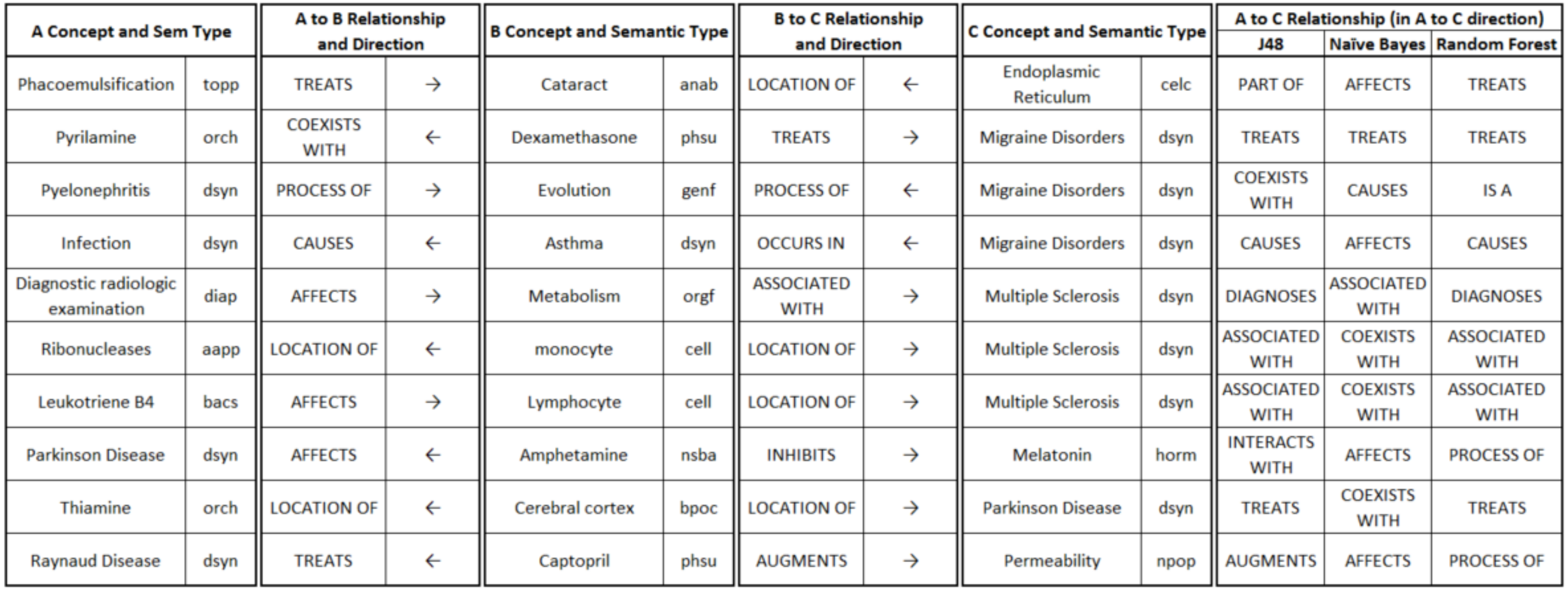
Table 5.4 shows the validation results for regularized outcomes classifiers trained in Section 4.3.3. As noted in Section 4.3, classifiers trained on newer data were also included in summarized results since they provide classifiers that were trained on much larger sets of data with more possible semantic types and relationships. The data used to train classifiers in the newer date range had no mentions of the diseases of interest. They were considered in the results because they provide a larger data set and, thus, may provide more meaningful results (See end of Section 3.2). The results in Table 5.4 show that the classifiers are able to predict, with some success, the natures of the A to C LBD relationships. This was the goal of this research.

The classifiers trained on the smaller set of data (4997 in the 1980-1984 data range as noted in Table 4.2) produced validated results approximately 40% of the time (left side of Table 5.4) while the classifiers trained on the larger data set (203,323 in the post 1984 data range, also in Table 4.2) were able to perform a bit better at approximately 50% validated accurately (right side of Table 5.4). The learning curves presented in Section 4.3.3 showed, also, that the post-1984 data set achieved higher percent classified accurately but took more training data to get to those higher percentages. Another important note is that the larger post-1984 training data set also contained many more possible semantic types and relationships than the smaller 1980-1984 data set (see Table 3.13). The classifiers trained on the smaller data set contained a subset of the feature values compared with the larger set. This is why the number of mentions in SemMedDB is smaller in the classifiers trained on 1980-1984 data range (61,488 instead of the full 70,795).

Chapter 4 presented classification performance results in Table 4.2. As noted in Section 5.3, based on cross-validation, over 60% of the predictions using the trained classifiers should be accurate with J48 possibly being able to predict in the mid 70% range. However, this is not the case for single classifier validation using the 1980-1984 data set. The validation of the 61,488 LBD candidates using the regularized outcomes experiment design was lower for this training set (J48 was 39%, Naïve Bayes – 42% and Random Forests – 38%), but these results are are still higher than the baselines from those initial classifier training sessions of 24% to 30% presented in Table 4.2. Using the post-1984 training sets produced higher results (J48 was 51%, Naïve Bayes – 46% and Random Forests – 50%). Nevertheless, these results are encouraging given that the baseline results were less than 30% for both the classifiers trained with 1980-1984 and the post-1984 training sets. The results are still less than those achieved using cross-validation in training the classifiers. Remember that this does not mean that the explained discoveries are necessarily inaccurate – it only means that they cannot be validated using the time slice approach with the data currently available. Also of note is that Naïve Bayes still performs consistently worse than other classifiers when using the larger training set (post-1984). The lower validation numbers and the mixed predicted results shown in many of the samples like those shown in Table 5.5 (to be discussed in Section 5.4.2) are what led to the idea of combining the classifier results into ensembles as will be presented in Section 5.5.

5.4.2 Candidates for Expert Validation

Of the 312,426 LBD partially qualified triples, approximately 70,000 of them had data against which validation could be performed. This section looks at a few of the discoveries that presented but that were not able to be validated using the newer literature. These candidates provide discoveries with predicted explanations that may be worth pursuing further with either experts' opinions or with clinical trials.



A Concept and Sem Type		A to B Relationship and Direction		B Concept and Semantic Type		B to C Relationship and Direction		C Concept and Semantic Type		A to C Relationship (in A to C direction)		
										J48	Naïve Bayes	Random Forest
Phacoemulsification	topp	TREATS	→	Cataract	anab	LOCATION OF	←	Endoplasmic Reticulum	celc	PART OF	AFFECTS	TREATS
Pyrilamine	orch	COEXISTS WITH	←	Dexamethasone	phsu	TREATS	→	Migraine Disorders	dsyn	TREATS	TREATS	TREATS
Pyelonephritis	dsyn	PROCESS OF	→	Evolution	genf	PROCESS OF	←	Migraine Disorders	dsyn	COEXISTS WITH	CAUSES	IS A
Infection	dsyn	CAUSES	←	Asthma	dsyn	OCCURS IN	←	Migraine Disorders	dsyn	CAUSES	AFFECTS	CAUSES
Diagnostic radiologic examination	diap	AFFECTS	→	Metabolism	orgf	ASSOCIATED WITH	→	Multiple Sclerosis	dsyn	DIAGNOSES	ASSOCIATED WITH	DIAGNOSES
Ribonucleases	aapp	LOCATION OF	←	monocyte	cell	LOCATION OF	→	Multiple Sclerosis	dsyn	ASSOCIATED WITH	COEXISTS WITH	ASSOCIATED WITH
Leukotriene B4	bacs	AFFECTS	→	Lymphocyte	cell	LOCATION OF	→	Multiple Sclerosis	dsyn	ASSOCIATED WITH	COEXISTS WITH	ASSOCIATED WITH
Parkinson Disease	dsyn	AFFECTS	←	Amphetamine	nsba	INHIBITS	→	Melatonin	horm	INTERACTS WITH	AFFECTS	PROCESS OF
Thiamine	orch	LOCATION OF	←	Cerebral cortex	bpoc	LOCATION OF	→	Parkinson Disease	dsyn	TREATS	COEXISTS WITH	TREATS
Raynaud Disease	dsyn	TREATS	←	Captopril	phsu	AUGMENTS	→	Permeability	npop	AUGMENTS	AFFECTS	PROCESS OF

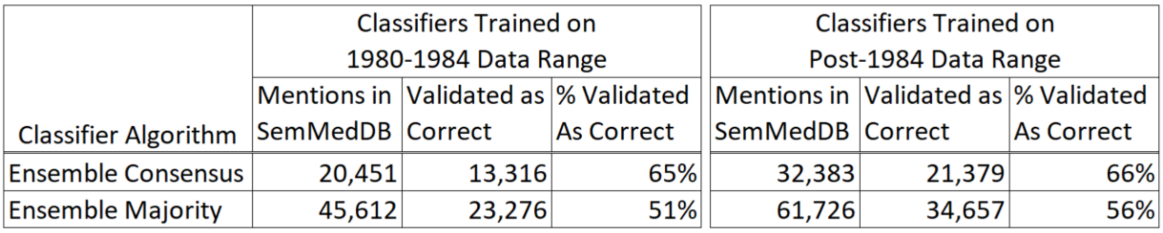
Table 5.5 Samples of Explained LBD Results - these are from classifiers trained on 1980-1984 data using regularized outcome design

A sample of results are shown in Table 5.5. These are predictions from classifiers trained on 1980-1984 data using the regularized outcomes data design. A similar set of results was generated by passing the same LBD partial triples to the classifiers trained on post-1984 data. The data in Table 5.5 includes the possible discovery shown in the second row that pyrilamine, an organic chemical (semantic type of orch), may treat migraine disorders, a disease or syndrome (dsyn). Having a drug that treats a disease is logical. It is also encouraging, that this TREATS explained discovery is predicted by all three classifiers. The third row in Table 5.5 shows a relationship between two diseases or syndromes – pyelonephritis and migraine disorders. Here each classifier came up with a different prediction – COEXISTS WITH, CAUSES and IS A. Pyelonephritis is a bacterial infection of the kidneys and is often accompanied with flu-like symptoms including headaches. Migraines are a kind of headache, so logical that these two conditions are related with the COEXISTS WITH and CAUSES relationships being plausible. The last row in the table indicates a relationship between Raynaud's disease and permeability. In the medical sense, permeability relates to the ability of gases or liquids to pass through matter. Raynaud's is where capillaries close down preventing the flow of blood and causing fingers and toes to turn white. In a fundamental way, this is a permeability problem. In the predictions by the classifiers, again, there is confusion among the three classifiers – AUGMENTS, AFFECTS and PROCESS OF – for the explained relationship between these two concepts. Here the relationships are similar – an augments relationship is similar to one that affects.

The purpose of this section was to present some explained LBD relationships that are not able, at this time, to be validated. The results show that sometimes there is consensus between the three classifiers applied, but, also, that sometimes there is none. The idea to look at ensembles to try to better predict LBD relationships comes from examination of these non-validated results. Ensembles are presented in the next section.

5.5 Additional Experiments Using Ensemble Learning

The percentage of explanations, shown in Table 5.4, that were able to be validated as accurate is lower than the best possible percentage of accurately classified results that cross-validation tests showed in Chapter 4. This fact, along with observations from Section 5.4.2, are what led to further investigation of applying some sort of ensemble classification to the task of explaining LBD candidate pairs. This set of experiments applies a variation of ensemble learning that combines results of heterogeneous classifiers in ways that may improve the accuracy of LBD relationship predictions. The approach uses a stacking ensemble learning



	Classifiers Trained on 1980-1984 Data Range			Classifiers Trained on Post-1984 Data Range		
Classifier Algorithm	Mentions in SemMedDB	Validated as Correct	% Validated As Correct	Mentions in SemMedDB	Validated as Correct	% Validated As Correct
Ensemble Consensus	20,451	13,316	65%	32,383	21,379	66%
Ensemble Majority	45,612	23,276	51%	61,726	34,657	56%

Table 5.6 Validation of LBD Explanations from Ensemble Classifiers

variation. Stacking usually takes the results from base learners and provides their outcomes as the inputs to a different, higher level classifier called a metalearner (See Section 2.8). A voting approach, which is normally used in bagging approaches to ensemble learning, is used with stacking (See Sections 2.8.1 and 2.8.3). This approach takes results from different classifiers and combines them with majority or consensus vote to determine final outcomes of the stacked ensemble learner.

The results of ensemble learning applied to validating results are shown in Table 5.6. The ensembles are created by combining primary classifiers' results with either consensus and majority tallying techniques. The next sections discuss the details of consensus ensembles (Section 5.5.1), majority ensembles (Section 5.5.2) and further study of actual explained discoveries (Section 5.5.3). This sections ends with a summary (Section 5.5.4).

5.5.1 Consensus Ensembles (All Match)

The predictions from all of the classifiers considered must match at least one sentence from SemMedDB for the consensus ensemble to declare a match. Sometimes more than one sentence with different relationships are found in SemMedDB. For there to be consensus in this work, one or more sentence, from each classifier prediction, must agree. Two sets of consensus ensembles were studied – those where the J48, Naive Bayes and Random Forest classifiers agreed using the classifiers trained on 1980-1984 data (left side of Table 5.6) and those where the three classifiers agreed using the classifiers trained on post-1984 data (right side of Table 5.6). The ensembles were tallied using the classifiers that regularized the outcomes to the A to C direction only (recall that regularized outcomes data design was described in Section 3.5.2 and the classifier training results for regularized outcomes in 4.3.3). Results for consensus are labeled "Ensemble Consensus" in Table 5.6.

The number of mentions of candidate pairs in newer SemMedDB that all agreed was 20,451 from the 1980-1984 data range and 32,383 from the post-1984 data range. Out of these mentions, 13,316 (65%) and 21,379 (66%), respectively, matched the predictions provided by all three of the classifiers. The consensus ensembles produce more accurate results than the individual classifiers (Table 5.4). Additionally, the consensus results of approximately 65% are closer to the mid-70% range that the cross-fold evaluation predicted should be able to be achieved.

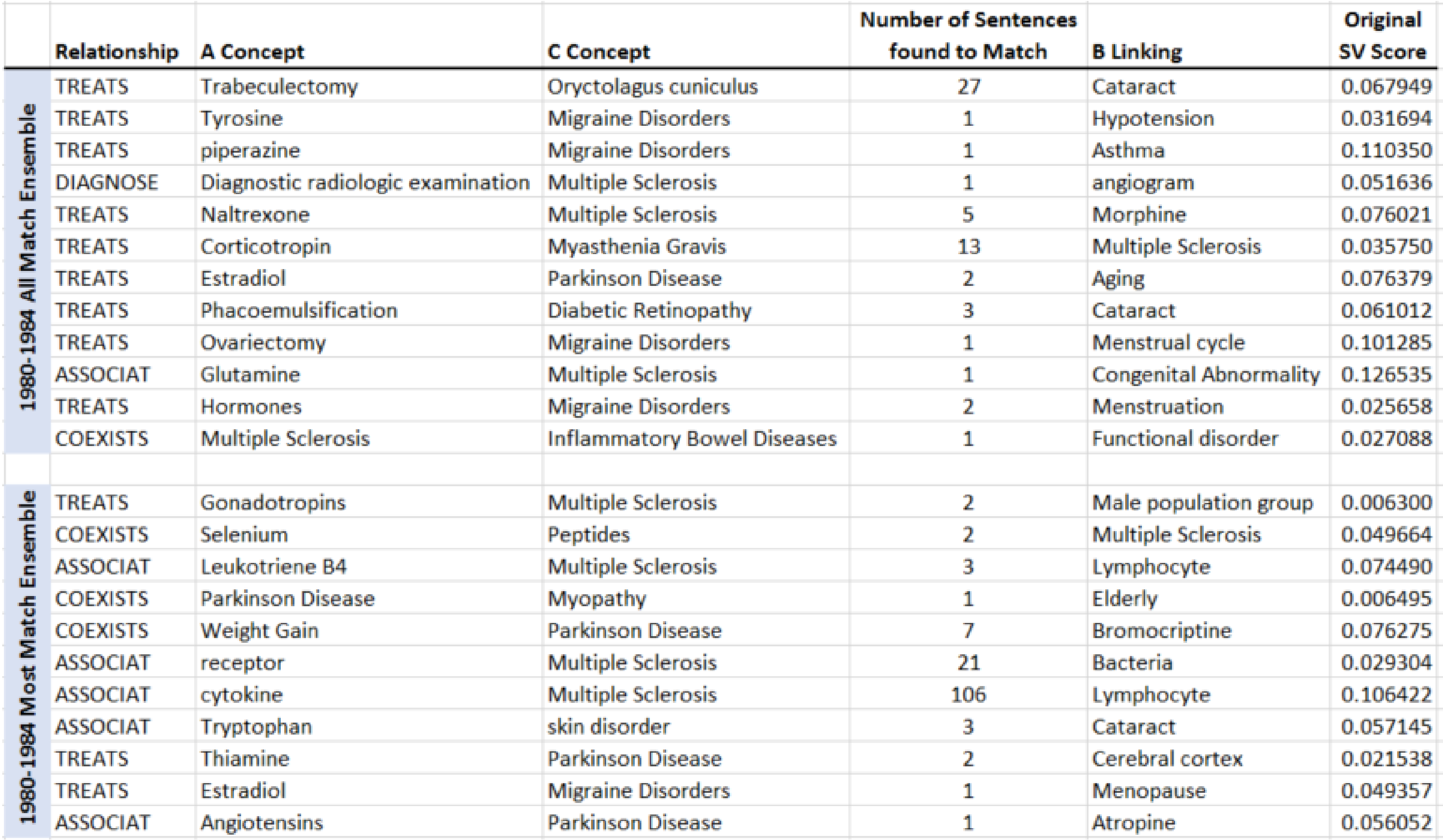
5.5.2 Majority Ensembles (Most Match)

The predictions from most of the classifiers considered must match for at least one sentence from SemMedDB for majority ensembles to declare a match. As noted with consensus ensembles, sometimes more than one sentence with different relationships are found in SemMedDB. For there to be majority agreement in this work, one or more sentence must agree and there must be agreement in two or three out of the three classifier predictions. Two sets of majority ensembles were studied – those where most of the J48, naive Bayes and random forest classifiers agreed using the classifiers trained on 1980-1984 data (shown on left side, last row of Table 5.6) and those where most of the classifiers agreed using the classifiers trained on post-1984 data (shown on right side, last row of Table 5.6). The results for majority are labeled "Ensemble Majority" in Table 5.6.

The number of mentions of candidate pairs in newer SemMedDB where a majority agreed was 45,612 from the 1980-1984 data range and 61,726 from the post-1984 data range. Out of these mentions, 23,276 (51%) and 34,657 (56%), respectively, matched the predictions provided by most of the classifiers. The majority ensembles produce more results classified and validated as being accurate than the individual classifiers (Table 5.4) but were lower than the consensus results. A difference is that approximately double number of validated candidates were identified with majorities than were identified with consensus and, likewise, more of these candidates were validated as being accurate. The most interesting point noted in these results is that the ensembles are necessary to increase the chance of accurate predictions which is a primary goal of this research – assist researchers in focusing on the most promising LBD candidates for future research and clinical trials. Ensembles reduce the number of results that were able to be validated, but allowed more accurate predictions.

5.5.3 Examples of Explained Discoveries

Table 5.7 shows samples of the validated results from the ensemble studies from the regularized outcomes design. The relationship from A to C is always in the A to C direction. All of the discoveries that appear in the all match ensembles also appear (by definition) in the most match ensembles. They are left off of the lower part of the table so that other majority



	Relationship	A Concept	C Concept	Number of Sentences found to Match	B Linking	Original SV Score
1980-1984 All Match Ensemble	TREATS	Trabeculectomy	Oryctolagus cuniculus	27	Cataract	0.067949
	TREATS	Tyrosine	Migraine Disorders	1	Hypotension	0.031694
	TREATS	piperazine	Migraine Disorders	1	Asthma	0.110350
	DIAGNOSE	Diagnostic radiologic examination	Multiple Sclerosis	1	angiogram	0.051636
	TREATS	Naltrexone	Multiple Sclerosis	5	Morphine	0.076021
	TREATS	Corticotropin	Myasthenia Gravis	13	Multiple Sclerosis	0.035750
	TREATS	Estradiol	Parkinson Disease	2	Aging	0.076379
	TREATS	Phacoemulsification	Diabetic Retinopathy	3	Cataract	0.061012
	TREATS	Ovariectomy	Migraine Disorders	1	Menstrual cycle	0.101285
	ASSOCIAT	Glutamine	Multiple Sclerosis	1	Congenital Abnormality	0.126535
	TREATS	Hormones	Migraine Disorders	2	Menstruation	0.025658
	COEXISTS	Multiple Sclerosis	Inflammatory Bowel Diseases	1	Functional disorder	0.027088
1980-1984 Most Match Ensemble	TREATS	Gonadotropins	Multiple Sclerosis	2	Male population group	0.006300
	COEXISTS	Selenium	Peptides	2	Multiple Sclerosis	0.049664
	ASSOCIAT	Leukotriene B4	Multiple Sclerosis	3	Lymphocyte	0.074490
	COEXISTS	Parkinson Disease	Myopathy	1	Elderly	0.006495
	COEXISTS	Weight Gain	Parkinson Disease	7	Bromocriptine	0.076275
	ASSOCIAT	receptor	Multiple Sclerosis	21	Bacteria	0.029304
	ASSOCIAT	cytokine	Multiple Sclerosis	106	Lymphocyte	0.106422
	ASSOCIAT	Tryptophan	skin disorder	3	Cataract	0.057145
	TREATS	Thiamine	Parkinson Disease	2	Cerebral cortex	0.021538
	TREATS	Estradiol	Migraine Disorders	1	Menopause	0.049357
	ASSOCIAT	Angiotensins	Parkinson Disease	1	Atropine	0.056052

Table 5.7 Samples of validated matches from regularization results (Sample drawn from 22 Consensus and 70 Majority predictions that were validated as being accurate. All shown that matched in the "1980-1984 All Match" set are also present in the "1980-1984 Most Match" results. They are not shown here for brevity.)

matches may be shown. After the A and C concepts is the number of sentences that mention the discovery. The last column is the Semantic Vectors scores for the original candidate LBD pair. The set of examples shown provides examples that mention some of the diseases of interest.

The actual sentences describing the discoveries is available because the SemMedDB data contains the sentence references for every stored relationship between concepts. As an example, the five sentences for cytokine being a associated with multiple sclerosis which were identified and validated in the most match ensemble with an ASSOCIATED WITH relationship as the explanation of the LBD candidate pair. They are:

- "Substances that down-regulate cytokines such as TNF-alpha or promote IL-10 or TGF-beta can be anticipated to affect MS beneficially."
- "This result raised the possibility that these cytokines played an important role in the demyelinating process in SCID-hu-MS."
- "The role of cytokines in multiple sclerosis."
- "Cytokines in multiple sclerosis: methodological aspects and pathogenic implications."
- "Cytokines have a central role in multiple sclerosis (MS) pathogenesis and may contribute to the aetiology of MS."

These sentences found in newer literature support the explained LBD relationship that cytokine is associated with multiple sclerosis which was never mentioned in the 1980-1984 abstracts used for LBD. Further research into cytokine and multiple sclerosis shows that there were mentions in papers but either synonyms were used that were not indicated in the UMLS Metathesaurus or the mention was not in the abstracts which were used for the LBD in this work.

Another example are these sentences for naltrexone being identified by the consensus ensemble as a treatment for multiple sclerosis:

- "There is overwhelming anecdotal evidence, that in low doses naltrexone not only prevents relapses in MS but also reduces the progression of the disease."
- "Low-dose naltrexone for multiple sclerosis and autism: does its benefit reveal a common cause?"
- "Low-dose naltrexone for treatment of multiple sclerosis: clinical trials are needed."

- "Pilot trial of low-dose naltrexone and quality of life in multiple sclerosis."
- "Prevention and diminished expression of experimental autoimmune encephalomyelitis by low dose naltrexone (LDN) or opioid growth factor (OGF) for an extended period: Therapeutic implications for multiple sclerosis."

These sentences found in newer literature support the explained LBD relationship that naltrexone may be beneficial to those with multiple sclerosis that was never mentioned in the 1980-1984 literature. However, some caution is also indicated in these references – for example, the third mention says "clinical trials are needed".

5.5.4 Summary of Ensemble Learning

This section has presented two approaches, consensus and majority, to ensemble learning using various combinations of the trained classifiers previously presented. The results of these ensemble experiment prove to be more significant and are novel for identifying the most interesting and likely to be accurately explained LBD candidate relationships. The most accurate predictions of LBD explanations were obtained using the consensus ensembles using the classifiers trained on the post-1984 data using the regularized outcome design (66% with 21,379 relationships accurately validated). The ensembles achieve higher percentages validated as being accurate compared with individual classifiers.

However, the drawback of the consensus ensembles is that fewer discoveries are available after this processing. Majority ensembles reduced the accuracy of the predictions but provided many more explained discoveries that may provide real discoveries. Larger training sets were also important in providing more accuracy in both the consensus and the majority ensembles.

Additionally, all ensembles performed better than individual classifiers when trying to validate the results. Ensemble learners are able to achieve higher precision with reduced recall, but, when the goal is to focus researches to the discoveries most likely to prove fruitful, this is a reasonable trade-off. The approach of using classifiers to explain the natures of LBD relationships works best with some form of ensemble. Using classification to explain LBD relationships should be viable in practice based on the results presented in this chapter of the ability to use time slice approaches to validate the predictions presented for the 1980-1984 LBD candidate pairs.

5.6 Chapter Summary

This chapter has presented the data and experiments performed to automatically identify and explain the nature of the hidden knowledge relationships. This section summarizes the experiments performed and presented in this chapter. First a corpus obtained from MEDLINE of 692,399 abstracts was prepared for a statistically based approach to LBD. The preparation included identifying a set of 15,427 medical terms on which to focus and normalizing these terms to handle occurrences of their synonyms. With a normalized corpus, a statistical approach to LBD is applied which is based on co-occurrence of the same or similar concepts in documents to determine the relatedness of documents. In addition to being statistically related in the corpus, term pairs must not appear in the same document to be retained as candidate LBD pairs. This produced 17,799,335 candidate LBD pairs.

The next steps in explaining candidate LBD pairs is to complete the partial triples required as inputs to the classifiers. This involves finding linking B terms and explaining the A-B and B-C relationships. A gold standard is provided by a database (SemMedDB) that was compiled by other researchers and organizations. This database assists in two ways – it provides the data required to create partial triples out of the candidate LBD pairs, and it provides a set of newer data of that is used to validate any results produced by the classifiers.

From SemMedDB, a total of 6,119,954 linking B terms were identified for the set of 17,799,335 candidate LBD pairs. Then this set was further studied using SemMedDB to identify the natures of the A-B and the B-C triples. This step is to create the data required by the classifiers – the partial triples. A total of 312,426 partial triples were completed and used as inputs to the classifiers.

Explaining the candidate LBD pairs from 1980-1984 was performed by using the classifiers trained in Chapter 4 that used the regularized outcomes design. The classifiers predicted explanations of the A-C relationships identified by LBD techniques. Validation of the predictions was performed using the time slice methodology using newer data to validate or invalidate the explained relationships. SemMedDB was searched for each of the 312,426 LBD candidate pairs, and this provides the validation data. The validation data examined were those based off of the regularized outcomes design and were the data mentioned in SemMedDB articles dated after 1984 (a total of 70,795 records were found to mention the candidate LBD pairs). In this chapter, the individual classifiers applied to the candidate LBD pairs produced validated results that were accurate approximately 40 to 50% of the time. While not as good as classifier training percentages presented in Chapter 4, these results are encouraging. Recall that results that cannot, yet, be validated may not be inaccurate – they just cannot be validated with relationships found in newer literature.

Ensemble learning was the last set of experiments explored. The goal was to determine if ensembles learners were able to be even more accurate in their predictions. Two types of ensembles were created using combinations of the primary individual three classifiers (J48, random forest and naïve Bayes). Consensus and majority are the two ensemble types. The results were more accurate when validated against newer facts. The consensus classifiers produced the highest percentage of validated predictions. While the majority ensembles produced more accurately validated instances, they produced smaller percentages of accurately classified predictions.

The trade-offs produced with ensemble learners is that the most accurate produced smaller sets of results. For example, the consensus voting of the three primary classifiers on the post-1984 data set, 66% accuracy was achieved, but only 13,316 discoveries resulted out of 20,451 possible that could have been validated (Table 5.6) which is also much less than the 70,795 possible discoveries examined with the single classifiers (Table 5.4) . However, improvements using consensus ensembles allows accurate validation to achieve percentages closer to the cross-validation accurately classified percentages found during training (66%, here, versus 70%, or more, during classifier training as shown in Table 4.2).

Chapter 6

Conclusions

The primary objective of this thesis was to determine if the natures of the relationship could be automatically explained using supervised machine learning classification. To satisfy this objective, this work first explored data representations suitable for applying classification techniques to explain the relationships. Then, this work applied traditional classification evaluation methods on both classifier outcomes and data designs. Classifiers applied to the training data successfully predicted the A to C relationships over 70% of the time, while the chosen baselines only achieved approximately 30% accurately predicted relationships. The classifiers were then used on real LBD candidate pairs from an older set of MEDLINE abstracts found using statistical LBD. The predicted LBD explanations were validated against more recent literature, which is the time-slice validation approach. The validations of predicted relationship explanations using ensemble classifiers achieved 66% accuracy.

This thesis researched approaches to automatically explaining the nature of LBD relationships. The results are exciting in that this work has proven that this problem can be solved with classification and that the results can be validated. The design of the classifiers did not focus on the exact concepts being considered, but, instead, used the semantic types of the concepts. This combined with the direction and nature of the relationship provided enough information to obtain promising results. Additionally, the time slice approach was able to be applied for validation of the explained relationships. This allowed validation without requiring experts from the medical domain to provide the validation. The time-slice validation showed that many of predictions from ensemble learners from the 1980-1984 data were corroborated by finding the same explanations cited in newer literature.

The rest of this chapter discusses how this thesis answers the questions proposed in the introduction (Chapter 1) and how the limitations presented in the literature review (Chapter 2) were addressed. This chapter also presents areas of future work that could be performed to further explore the problem of explaining literature based discoveries. The end of this chapter provides some final words about this work.

6.1 How this Work Addresses the Research Questions

Section 1.2 presented a set of research questions that form the focus of this thesis. This section summarizes the contributions this thesis has provided in addressing these questions. The primary question of this thesis was:

> *Can the nature of the LBD relationships be automatically explained?*

Yes, the nature of LBD relationships may be explained using text classification. There are limitations in that the numbers of accurately explained results becomes small compared with the number of discoveries offered by LBD. However, this is to be expected since there is consensus that LBD produces many more unimportant and unrelated concepts than real and truly important discoveries (Swanson et al., 2006; Kostoff et al., 2007; Preiss, 2014; Cameron et al., 2015). An additional encouraging part of this research is that refinement to the classification results using ensemble learning is able to further filter the explained relationships to those that have better probability of being accurately explained (improved accuracy of results). Accuracy of the suggested explanations is important since a goal of this work is to assist researchers in narrowing in on the most likely and most promising discoveries presented by LBD.

Additional research questions, and discussions of how this thesis addressed them, are:

1. Can this problem be modeled as a classification problem?

   Yes, it can. Classifier design in this work was built around the semantic types of the A, B or C terms and the natures of known relationships between A and B and between B and C terms. The relationships also include an indication of the direction of the relationship. For example, for one of the candidate LBD pairs of nitrofurans being related to multiple sclerosis shown in Figure 5.2, nitrofurans (A) and multiple sclerosis (C) are related with the B term, bacteria. The MEDLINE semantic types of nitrofurans, multiple sclerosis and bacteria are orch, dsyn and bact, respectively. The identified relationship between the A and B is LOCATION_OF with a direction of B to A and the relationship between B and C is PART_OF with a direction of C to B. The five classes are modeled as the A, B and C semantic types along with the A-B and B-C relationship and direction. The prediction or outcome of the classifier is the nature and direction of the A to C relationship.

2. Is there a source of training data for classifier training?

   The Semantic MEDLINE Database or SemMedDB is a source of training data. The SemMedDB includes the semantic types and nature of relationships between concept pairs and includes references to the sentences in MEDLINE supporting the noted relationships. This source allows generation of training data sets and may be constrained to specific time slices.

3. Once trained, do classifiers produce measurably better results than baselines?

   Yes, in all cases, the training data performance of the classifiers was better than a baseline that chose the most frequently occurring outcome. The best classifier and feature designs achieved over 70% accuracy while the baselines with those same feature designs were approximately 30% accurate.

4. Do classifiers produce results that can be validated against known facts?

   The time slice approach, described by Yetisgen-Yildiz and Pratt (2009), was successful at validating explained LBD results. After the small set of validated results emerged from the ensemble learners, the supporting sentences were able to be matched with the same predictions 66% of the time while baselines from classifier training were accurate approximately 30% of the time. SemMedDB provided a source of facts for newer relationships that did not occur in the older time slice. From this source, the correlated sentences allowed validation of the explanations predicted by the classifiers and ensembles of this research.

5. Are ensemble learners better than single classifiers in predicting LBD explanations?

   Yes, they are – ensemble learners reduce the number of explained results, but when compared against known facts, they produce a more accurate set of predictions than single classifiers. Ensemble learners were able to accurately predict the natures of the LBD relationships 66% of the time while single classifiers were only accurate 40-50% of the time. Recall that baseline classification performance using training data was approximately 30%.

6.2 Contributions

This section examines how this thesis has improved on known limitations initially discussed in Section 2.5

6.2.1 Explanation of LBD Candidate Relationships

Section 2.5.1 discussed that a limitation of LBD systems is that the nature of the relationship is not explained by LBD. Explaining LBD relationships is the primary contribution of this research. Explaining the LBD relationships has been turned into a classification problem (Chapter 4), and this supervised machine learning has successfully explained LBD relationships automatically including using a time slice approach to validate the explanation predictions (Sections 5.3 and 5.4). Additionally, ensemble learning used in this research provides a more accurate set of explained LBD relationships (Section 5.5).

6.2.2 Lack of Accuracy and Excess Quantity Using Co-Occurrence

Section 2.5.2 introduced some of the problems raised by using co-occurrence techniques for performing LBD. Wren (2008b) suggested that limits be placed on the vocabulary for the A and C terms used in the closed LBD. This work used limited vocabularies (concepts) as suggested (Section 3.1) and also normalized the corpus in a way that handles a problem of synonyms (Section 5.1.1). When a concept has a synonym, it is important when using co-occurrence techniques for LBD that all the synonym occurrences are presented as the same concept. In this work, synonyms were identified using MetaMap and the CUIs for the concept and the CUIs for all synonyms of that concept replaced the actual concept in the corpus.

Additionally, the problem of too many false positives or noise was addressed in this work by applying ensemble learning techniques. The ensemble learning (Section 5.5) assisted greatly in refining the explained relationships down to those most likely to be correct. The results of the ensemble classifiers were validated using time slice techniques and, although the numbers of results from the ensembles were smaller, they were more accurate than single classifier applications.

6.2.3 Using LSA Does Not Identify Candidate B Terms

Section 2.5.3 introduced the problem of using LSA for LBD that linking B terms are not identified during LSA. The approach used in this work was to identify candidate B terms using another source of concepts and their relationships (SemMedDB) to identify candidate linking B terms. SemMedDB represents the knowledge of experts. It was searched for concepts related to A and concepts related to C and the intersection became the set of candidate B terms.

6.3 Future Work

This section discusses some areas for future enhancement research that might be applied to the task of explaining literature based discovery. While many questions and alternative experiments were considered and explored in this research, many more questions and ideas presented themselves during the course of this work than were able to be explored during this research. This section discusses some of these ideas for future investigation.

Explaining LBD from a non-medical domain This work focused on the medical domain with data provided by MEDLINE and SemMedDB. An exciting extension of this work would be to develop the same classification solution on a totally different domain. For example, networks of individuals, groups and events where the goal of the discoveries would be to link previously unrelated entities (LBD) and suggest explanations of how they may be related.

Studying more concepts In this work, all available concepts were not included in classifier training (Section 3.1) and in LBD (Section 5.1.1). Further studies could try to include more concepts. Also, more sophisticated normalization of terms could have been employed including more complete list of concepts and application of stemming to allow plural and singular forms of concepts to be handled. This work depended on MetaThesauraus to provide normalization of very closely related or actually the same concepts. There may be other ways to identify the same concepts that could be explored further.

Applying relationship extraction ReVerb is an OpenIE tool (Section 2.6.4) that was briefly tested as a means to explain the A-B and B-C relationships in the partial triples. While it presented some relationship explanations, it did not line up with the relationships found in SemMedDB. Ultimately, SemMedDB was used to identify gold standard results against which explained candidate LBD pairs were validated. SemMedDB has specific relationship generalizations like TREATS and CAUSES whereas a relationship extraction capability would find the literal relationship and then a secondary normalization into the SemMedDB relationship generalizations would need to be performed. For this reason ReVerb was not used. Using other methods to explain the identified relationships found in same document mentions (without actually reading all the journals) would be a good topic of future study.

Exploring SV configuration In this work, 0.001 was used as the threshold of the SV related results to include in studied results and what to leave out. Table 5.7 presents samples of validated LBD explanations and indicates that scores less than 0.01 may less likely to produce good results. Additional experiments would include raising the threshold of which results to include and which to leave out.

Another SV parameter is the vector dimension for the random indexing portion of SV – this work used the default value of 200. Kanerva et al. (2000) suggests using 300. Additional experiments would involve altering the threshold score of results considered and altering the vector dimension.

Testing with larger corpus LBD was performed on documents collected from the 2002 MEDLINE abstracts corpus. This decision was made to provide a smaller set of data on which to perform LBD. The time slice approach was used to validate results. Therefore, LBD was performed on the 1980-1984 time and validation was performed on newer literature. Future work could include performing experiments using more recent MEDLINE collections and using larger time slices on which to perform LBD. Additionally, only the abstracts were used in LBD. A corpus that included the full texts of the literature would possibly produce even more interesting LBD candidates.

Applying different B term selection methodologies The SemMedDB provided a source for identifying candidate B terms in this work. An alternative methodology for B term selection would be to extract directly from the documents using NLP. This introduces the problem of not knowing how the B term may be related to the A or C term. Additional processing would consider word proximity, first, to provide only B terms that occur near the A or C term. Then further NLP would be required to identify the nature of the A to B or the B to C relationship.

Applying enhancements to ensemble learning Ensemble learning introduces many options and varieties of how to develop the ensemble. Boosting algorithms could be applied – these include simply applying more weight or confidence to the results of some of the underlying classifiers than others. More than three classifiers could also have been used and combined to produce the ensemble results. The Weka suite of tools also includes some ensemble tools that could have been considered (page 372 of Witten et al. (2011)).

Modifying and adjusting the classifier data design The class designs did not consider combining similar relationship types. For example, PART_OF from B to A is similar to LOCATION_OF from A to B. This was identified by examining the confusion matrix results. Additional pre-processing of the data input to the classifiers could have combined some of the relationship synonyms in order to reduce the numbers of training instances and possibly increase the performance of the classifiers and ultimately increase the confidence of identifying valid results explaining the LBD relationships.

References

Albright, R. (2004). Taming text with the SVD. *SAS Institute Inc.* Cary, NC.

Banko, M. and Etzioni, O. (2008). The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, Columbus, Ohio. Association for Computational Linguistics.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):D267–D270.

Bradford, R. B. (2006). Relationship discovery in large text collections using latent semantic indexing. In *In Proceedings of SDM 06 (SIAM)*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bruckner, G., Webb, P., Greenwell, L., Chow, C., and Richardson, D. (1987). Fish oil increases peripheral capillary blood cell velocity in humans. *Atherosclerosis*, 66(3):237–245.

Bruza, P. and Weeber, M. (2008). *Literature-based Discovery*. Springer, 1st edition.

Cameron, D., Kavuluru, R., Rindflesch, T. C., Sheth, A. P., Thirunarayan, K., and Bodenreider, O. (2015). Context-driven automatic subgraph creation for literature-based discovery. *Journal of Biomedical Informatics*, 54:141 – 157.

Carreras, X. and Màrquez, L. (2005). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164. Association for Computational Linguistics.

Cios, K. J., Pedrycz, W., Swiniarski, R. W., and Kurgan, L. A. (2007). *Data Mining: A Knowledge Discovery Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Cohen, A. M. and Hersh, W. R. (2005). A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71.

Cohen, T., Schvaneveldt, R., and Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240 – 256.

Cohen, T., Widdows, D., Vine, L. D., Schvaneveldt, R., and Rindflesch, T. (2012). Many paths lead to discovery: Analogical retrieval of cancer therapies. In *International Symposium on Quantum Interaction*, Paris School of Economics, Paris. Springer.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Culotta, A., McCallum, A., and Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 296–303, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*. University of Sheffield Department of Computer Science.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop*, pages 1–15, Berlin, Heidelberg. Springer.

DiGiacomo, R. A., Kremer, J. M., and Shah, D. M. (1989). Fish-oil dietary supplementation in patients with Raynaud's phenomenon: a double-blind, controlled, prospective study. *The American journal of medicine*, 86(2):158–164.

Downey, D., Etzioni, O., and Soderland, S. (2005). A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 1034–1041, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam (2011). Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 11, pages 3–10. IJCAI/AAAI.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 1535–1545, Edinburgh, Scotland, UK.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181.

Ferrucci, D. and Lally, A. (2004). UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization. Technical report, University of Waikato, Department of Computer Science. Working Paper, 98/2.

Ganiz, M. C., Pottenger, W. M., and Janneck, C. D. (2005). Recent advances in literature based discovery. Technical Report LU-CSE-05-027, Lehigh University.

Gordon, M., Lindsay, R. K., and Fan, W. (2002). Literature-based discovery on the world wide web. *ACM Transactions on Internet Technology (TOIT)*, 2(4):261–275.

Gordon, M. D. and Dumais, S. (1998). Using latent semantic indexing for literature based discovery. *Journal of the American Society for Information Science*, 49(8):674–685.

Gordon, M. D. and Lindsay, R. K. (1996). Toward discovery support systems: A replication, re-examination, and extension of Swanson's work on literature-based discovery of a connection between Raynaud's and fish oil. *Journal of the Association for Information Science and Technology*, 47(2):116–128.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hall, M. A. and Frank, E. (2008). Combining naive Bayes and decision tables. In *FLAIRS Conference*, volume 2118, pages 318–319.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, New York, 2nd edition.

Henry, S. and McInnes, B. T. (2017). Literature based discovery: Models, methods, and trends. *Journal of biomedical informatics*, 74:20–32.

Hristovski, D., Dinevski, D., Kastrin, A., and Rindflesch, T. C. (2015). Biomedical question answering using semantic relations. *BMC Bioinformatics*, 16(1):1–14.

Hristovski, D., Friedman, C., Rindflesch, T. C., and Peterlin, B. (2006). Exploiting semantic relations for literature-based discovery. *AMIA Annual Symposium Proceedings*, 2006:349–353.

Hristovski, D., Friedman, C., Rindflesch, T. C., and Peterlin, B. (2008). Literature-based knowledge discovery using natural language processing. In Bruza, P. and Weeber, M., editors, *Literature-based discovery*, volume 15 of *Information Science and Knowledge Management*, pages 133–152. Springer-Verlag Berlin Heidelberg.

Hung, C. and Chen, J.-H. (2009). A selective ensemble based on expected probabilities for bankruptcy prediction. *Expert Systems with Applications*, 36(3):5297 – 5303.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing*. Prentice Hall, 2nd edition.

Kalman, D. (1996). A singularly valuable decomposition: The SVD of a matrix. *The College Mathematics Journal*, 27(1):2–23.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159.

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the Annual Conference of the Cognitive Science Society*, volume 22, pages 103–6.

Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., and Rindflesch, T. C. (2012). SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics*, 28(23):3158–3160.

Kohavi, R. (1995). The power of decision tables. In *European conference on machine learning*, pages 174–189. Springer.

Kostoff, R. N. (2008). Literature-related discovery (LRD): Introduction and background. *Technological Forecasting and Social Change*, 75(2):165 – 185.

Kostoff, R. N., Block, J. A., Solka, J. L., Briggs, M. B., Rushenberg, R. L., Stump, J. A., Johnson, D., Lyons, T. J., and Wyatt, J. R. (2007). Literature-related discovery (LRD). Technical Report ADA473438, Office of Naval Research.

Kostoff, R. N., Block, J. A., Solka, J. L., Briggs, M. B., Rushenberg, R. L., Stump, J. A., Johnson, D., Lyons, T. J., and Wyatt, J. R. (2008a). Literature-related discovery (LRD): Lessons learned, and future research directions. *Technological Forecasting and Social Change*, 75(2):276 – 299.

Kostoff, R. N., Solka, J. L., Rushenberg, R. L., and Wyatt, J. A. (2008b). Literature-related discovery (LRD): Water purification. *Technological Forecasting and Social Change*, 75(2):256 – 275.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Li, J., Zhang, Z., Li, X., and Chen, H. (2008). Kernel-based learning for biomedical relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5):756–769.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Lindsay, R. K. and Gordon, M. D. (1999). Literature-based discovery by lexical statistics. *Journal of the Association for Information Science and Technology*, 50(7):574–587.

Livingston, E. H. (2004). Who was student and why do we care so much about his t-test? *Journal of Surgical Research*, 118(1):58 – 65.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 4(2):R1.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press, Cambridge, UK.

Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC press.

McCallum, A., Freitag, D., and Pereira, F. C. N. (2000). Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

MEDLINE (2002). Baseline Repository, PubMed Data Files. U.S. National Library of Medicine. https://mbr.nlm.nih.gov/Download/Baselines/2002/, http://www.ncbi.nlm.nih.gov/pubmed/, or http://www.PubMed.gov/.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Microsoft Research.

Preiss, J. (2014). Seeking informativeness in literature based discovery. In *Proceedings of the 2014 Workshop on Biomedical Natural Language Processing*, pages 112–117.

Preiss, J., Stevenson, M., and Gaizauskas, R. (2015). Exploring relation types for literature-based discovery. *Journal of the American Medical Informatics Association*, 22(5):987–992.

Preiss, J., Stevenson, M., and McClure, M. H. (2012). Towards semantic literature based discovery. In *2012 AAAI Fall Symposium Series: Information Retrieval and Knowledge Discovery in Biomedical Text*, volume 30, pages 7–18.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Rindflesch, T. C. and Fiszman, M. (2003). The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477.

Rindflesch, T. C., Kilicoglu, H., Fiszman, M., Rosemblat, G., and Shin, D. (2011). Semantic MEDLINE: An advanced information management application for biomedicine. *Information Services & Use*, 31(1-2):15–21.

Rodriguez, J., Kuncheva, L., and Alonso, C. (2006). Rotation forest: A new classifier ensemble method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1619–1630.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Schapire, R. E. (2013). Explaining AdaBoost. In *Empirical Inference*, pages 37–52. Springer.

Sebastian, Y., Siew, E.-G., and Orimaye, S. O. (2017). Emerging approaches in literature-based discovery: techniques and performance review. *The Knowledge Engineering Review*, 32:e12.

Sehgal, A. K., Qiu, X. Y., and Srinivasan, P. (2008). Analyzing LBD methods using a general framework. In Bruza, P. and Weeber, M., editors, *Literature-based Discovery*, volume 15 of *Information Science and Knowledge Management*, pages 75–100. Springer-Verlag Berlin Heidelberg.

Seki, K. and Uehara, K. (2013). Supervised hypothesis discovery using syllogistic patterns in the biomedical literature. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1663–1669. AAAI Press.

Smalheiser, N. R. (2017). Rediscovering don swanson: The past, present and future of literature-based discovery. *Journal of Data and Information Science*, 2(4):43–64.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427 – 437.

Strang, G. (1980). *Linear algebra and its applications*. Academic Press.

Sutton, C. and McCallum, A. (2007). An introduction to conditional random fields for relational learning. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA.

Swanson, D. R. (1986a). Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18.

Swanson, D. R. (1986b). Undiscovered public knowledge. *The Library Quarterly: Information, Community, Policy*, 56(2):103–118.

Swanson, D. R. (1988). Migraine and magnesium: eleven neglected connections. *Perspectives in biology and medicine*, 31(4):526–557.

Swanson, D. R. (2011). Literature-based resurrection of neglected medical discoveries. *Journal of biomedical discovery and collaboration*, 6:34–47.

Swanson, D. R. and Smalheiser, N. R. (1996). Undiscovered public knowledge: A ten-year update. In *KDD*, pages 295–298. AAAI.

Swanson, D. R. and Smalheiser, N. R. (1997). An interactive system for finding complementary literatures: A stimulus to scientific discovery. *Artificial Intellelligence*, 91(2):183–203.

Swanson, D. R., Smalheiser, N. R., and Torvik, V. I. (2006). Ranking indirect connections in literature-based discovery: The role of medical subject headings. *Journal of the American Society for Information Science and Technology*, 57(11):1427–1439.

Tan, A. C. and Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification. In *Proceedings of New Zealand Bioinformatics Conference*. University of Glasgow. Te Papa, Wellington, New Zealand,13-14 February 2003.

Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., and Ananiadou, S. (2011). Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics*, 27(13):i111–i119.

UMLS (2012). Unified Medical Language System, National Institute of Health, U.S. National Library of Medicine, Version 2012AA. http://www.nlm.nih.gov/research/umls/.

Wang, G., Hao, J., Ma, J., and Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1):223 – 230.

Webb, G. I. (1999). Decision tree grafting from the all-tests-but-one partition. In *IJCAI*, volume 2, pages 702–707.

Weeber, M., Klein, H., de Jong-van den Berg, L. T., and Vos, R. (2001). Using concepts in literature-based discovery: Simulating Swanson's Raynaud-fish oil and migraine-magnesium discoveries. *Journal of the American Society for Information Science and Technology*, 52(7):548–557.

Widdows, D. and Cohen, T. (2010). The semantic vectors package: New algorithms and public tools for distributional semantics. In *2010 IEEE Fourth International Conference on Semantic Computing (ICSC)*, pages 9–15.

Widdows, D. and Ferraro, K. (2008). Semantic vectors: A scalable open source package and online technology management application. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Williams, N., Zander, S., and Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 36(5):5–16.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Burlington, MA, 3rd edition.

Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., and Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations. Working paper, University of Waikato.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Wren, J. D. (2008a). The "Open Discovery" challenge. In Bruza, P. and Weeber, M., editors, *Literature-based Discovery*, volume 15 of *Information Science and Knowledge Management*, pages 39–55. Springer-Verlag Berlin Heidelberg.

Wren, J. D. (2008b). Where is the discovery in literature based discovery. In Bruza, P. and Weeber, M., editors, *Literature-based Discovery*, volume 15 of *Information Science and Knowledge Management*, pages 57–72. Springer-Verlag Berlin Heidelberg.

Yetisgen-Yildiz, M. and Pratt, W. (2006). Using statistical and knowledge based approaches for literature-based discovery. *Journal of Biomedical Informatics*, 39(6):600–611.

Yetisgen-Yildiz, M. and Pratt, W. (2009). A new evaluation methodology for literature-based discovery systems. *Journal of Biomedical Informatics*, 42(4):633 – 643.