2019

Indirect Relatedness, Evaluation, and Visualization for Literature Based Discovery

Sam Henry
Virginia Commonwealth University

INDIRECT RELATEDNESS, EVALUATION, AND VISUALIZATION FOR LITERATURE BASED DISCOVERY

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

SAM HENRY

M.Eng., Cornell University, USA - August 2009 - May 2010

B.S., Virginia Commonwealth University, USA - August 2006 - May 2009

Director: Bridget McInnes, Ph.D.

Assistant Professor, Department of Computer Science

Virginia Commonwealth University

Richmond, Virginia

May, 2019

Acknowledgements

Thank you to my advisor, Dr. Bridget McInnes for her guidance and insight into this research. Thank you to the colleagues who contributed to this research: Alex McQuilken's help with association scores, Amy Olex's Hadoop implementation of CUI Collector, Clint Cuffy's word2vec expertise and package, Jasmine Norman's clustering work, Megan Charity's tensor flow work, Samhita Pendyal's clustering and visualization work, and Robert (Mack) Stump's server and linux help. Thank you to Dr. Halil Kilicoglu for his reading suggestions and SemRep expertise, Dr. Dayanjan S. Wijesinghe for his insight into visualization and pharmacological substances, Dr. Alberto Cano for his advice on MySQL queries to make concept expansion much faster, and Dr. Vojo Kecman for his advice on SVD and squaring sparse matrices. Thank you to my committee members for taking their time to read this, Dr. Alberto Cano, Dr. Bridget McInnes, Dr. Dayanjan S. Wijesinghe, Dr. Halil Kilicoglu, and Dr. Thang Dinh.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

INDIRECT RELATEDNESS, EVALUATION, AND VISUALIZATION FOR LITERATURE BASED DISCOVERY

By Sam Henry

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2019.

Director: Bridget McInnes, Ph.D.,
Assistant Professor, Department of Computer Science

The exponential growth of scientific literature is creating an increased need for systems to process and assimilate knowledge contained within text. Literature Based Discovery (LBD) is a well established field that seeks to synthesize new knowledge from existing literature, but it has remained primarily in the theoretical realm rather than in real-world application. This lack of real-world adoption is due in part to the difficulty of LBD, but also due to several solvable problems present in LBD today. Of these problems, the ones in most critical need of improvement are: (1) the over-generation of knowledge by LBD systems, (2) a lack of meaningful evaluation standards, and (3) the difficulty interpreting LBD output. We address each of these problems by: (1) developing indirect relatedness measures for ranking and filtering LBD hypotheses; (2) developing a representative evaluation dataset and applying meaningful evaluation methods to individual components of LBD; (3) developing an interactive visualization system that allows a user to explore LBD output in its

entirety. In addressing these problems, we make several contributions, most importantly: (1) state of the art results for estimating direct semantic relatedness, (2) development of set association measures, (3) development of indirect association measures, (4) development of a standard LBD evaluation dataset, (5) division of LBD into discrete components with well defined evaluation methods, (6) development of automatic functional group discovery, and (7) integration of indirect relatedness measures and automatic functional group discovery into a comprehensive LBD visualization system. Our results inform future development of LBD systems, and contribute to creating more effective LBD systems.

CHAPTER 1

INTRODUCTION

Scientific publications are the primary means for disseminating academic research. Databases contain millions of documents, and thousands are added each day. Scientific literature is growing at an exponential rate [1], and it's estimated that scientific output doubles every nine years [2]. This overwhelming amount of information makes it difficult for researchers to stay current, even in their own disciplines, leading to an increase in specialization and a fracturing of information. Literature Based Discovery (LBD) synthesizes new knowledge from fragments of information, to bridge disciplines, and generate hypotheses. As the scientific literature grows, LBD is becoming an increasingly necessary tool for facilitating research.

LBD [3] seeks to find information that is implicit in text, but never explicitly stated. New knowledge can be formed by piecing together fragments of information found across multiple documents. For example, one document may state that "A implies B" and another that "B implies C"; new knowledge is generated by hypothesizing that therefore "A implies C". In its simplest form, a hypothesis is an assertion that a relationship exists between two terms that never directly co-occur, and can be represented as a term-term pair. In modern LBD systems, hypothesis generation is often more complex and varied than the simple ABC paradigm we used as an example, but the goal of generating new knowledge by finding latent relationships is common to all systems.

LBD shows great promise, and several discoveries have been attributed to its use [3, 4, 5, 6, 7, 8, 9, 10, 11], but it has failed to achieve widespread adoption and

use in laboratory environments. The reasons for this lack of adoption are many, but we identify three problems that are in critical need of being addressed.

Problem 1: Over-generation of knowledge - LBD systems tend to create too many hypotheses, causing promising and meaningful hypotheses to be buried within false, uninteresting, or too obvious ones.

Problem 2: Lack of meaningful evaluation methods - LBD is difficult to evaluate, and evaluation methods are often ad-hoc and system specific. Without standard evaluation methods, LBD systems and components cannot be quantitatively compared or objectively improved.

Problem 3: Difficulty interpreting output - LBD systems often output hypotheses as a simple list of terms. This combined with their over-generation of knowledge means a user is presented with a list of hundreds or thousands of often unrelated terms, which are meaningless without extensive manual review.

In this dissertation, we address these problems, and present our work towards developing more effective LBD systems. One of our core assumptions is that an LBD hypothesis is an assertion that a relationship exists between two terms that never directly co-occur, and the likelihood of that hypothesis being true can be estimated by the strength of their relatedness. If we can predict the future direct relatedness between two indirectly related terms, then we can predict their likelihood of being a future discovery. Estimating the relatedness between two directly co-occurring terms is an established and well studied field, which we use as a starting point in our study of estimating indirect relatedness, for which little work has been done. We first address Problem 1, the over-generation of knowledge by LBD systems using indirect association measures to rank and filter hypotheses. We first apply association

measures and vector-based measures to direct relatedness, then we extend them to quantifying indirect relatedness.

Specific contributions to the fields of estimating direct semantic relatedness are presented in the Chapter 4, and include:

1. Concept Association and Expansion - we use association measures between concepts to estimate direct relatedness, and introduce concept expansion, which with concept associations is a novel methodology that accounts for lexical variation at both the synonymous and hyponymous levels.

2. Set Associations - we extend association measures to quantify relatedness between sets of terms rather than individual term pairs.

3. An analysis of association measures for direct relatedness, for which we achieve state of the art results.

4. An analysis of vector measures for direct relatedness, for which we achieve state of the art results.

Chapter 5 focuses on our contributions to estimating indirect relatedness, which include:

1. Indirect Association Measures - we develop four indirect association measures, including linking term association (LTA), minimum weight association (MWA), shared B to C association (SBC), and linking set association (LSA).

2. The development of a dataset and method for evaluating the ability to estimate future relatedness.

3. An analysis of indirect association measures and vector-based measures on their ability to estimate direct and future relatedness.

In Chapter 6, we address Problem 2, the lack of evaluation standards for LBD. Here, we develop an evaluation framework for LBD and test how well estimating future relatedness translates to term filtering and ranking for LBD. Specific contributions of this chapter are:

1. Development of a standard evaluation dataset.

2. Assignment of evaluation methods for LBD components in isolation and in combination.

3. Evaluation of indirect relatedness measures for term filtering and term rankings steps of LBD.

Our evaluation dataset better models the difficulty of LBD compared to other evaluation datasets, and can be used, along with our evaluation framework to analyze individual or sets of components of LBD. Furthermore our dataset and evaluation framework can be used by most LBD systems, and provides an evaluation standard for LBD.

In Chapter 7, we address Problem 3, difficulty interpreting output by developing an interactive visualization environment to explore LBD output in its entirety. This visual environment incorporates our indirect relatedness measures and several novel contributions, including:

1. Automatic Functional Group Discovery - we apply hierarchical clustering algorithms to automatically find functional groups (sets of related terms) in the LBD output.

2. Functional Group Ranking - we estimate the interestingness of the functional groups using indirect relatedness measures.

3. Interactive Visualization - we use the clustering hierarchy and cluster rankings to create an interactive visualization of LBD output. We use visual cues to aid a user in exploring LBD output interactively.

Using the visualization, we replicate the historic Raynaud's Disease - Fish Oil Discovery made by Swanson [3]. Our visualization makes finding eicosapentaenoic acid, the active ingredient in fish oil and present day treatments for Raynaud's disease easy, by providing an understanding of the output as a whole and visual cues to lead to promising terms.

This dissertation begins with Chapter 2, which provides the necessary background information. Next, in Chapter 3 we break LBD into a set of discrete components with well defined inputs and outputs. Using this framework, we provide a literature review for each component, but focus on those most relevant to our work. We also create a set of ideal evaluation criteria, and compare and contrast existing evaluation methodologies with respect to this criteria. The next four chapters cover our primary contributions towards direct relatedness (Chapter 4), indirect relatedness (Chapter 5), LBD evaluation (Chapter 6), and LBD output summarization and visualization (Chapter 7). We end with Chapter 8, where we present our final conclusions and future work.

CHAPTER 2

BACKGROUND

In this chapter, we present relevant background information, including the corpora, tools, and an overview of the topics of distributional semantics, association measures, and semantic similarity and relatedness. Figure 1 shows an overview of the tools and datasets used, and how they interact.

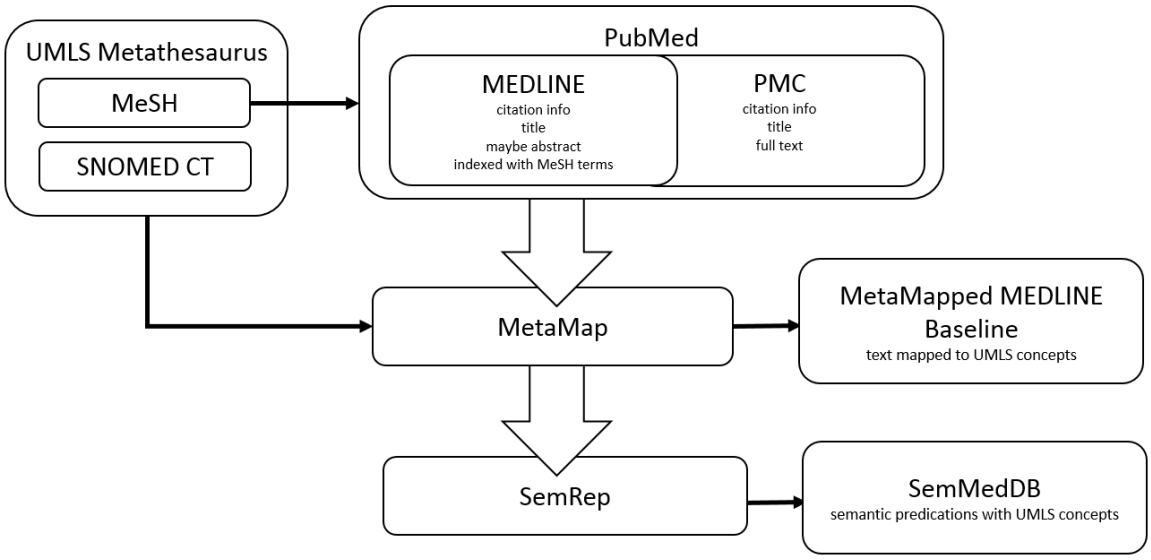

Fig. 1. An overview of the tools and datasets used

2.1 Corpora

A corpus is a collection of documents and a corpora is more than one corpus. In this section, we describe the corpora relevant to this work.

2.1.1 MEDLINE

MEDLINE is a database combining over 22 million biomedical and life sciences journal references[1] and is maintained by the National Institutes of Health, National Library of Medicine. The references include a title, author information, and 65% of the references contain abstracts, as of the 2015 baseline. The 2015 MEDLINE baseline encompasses approximately 5,600 journals, and contains 22,775,609 citations, of which 13,835,206 contain abstracts. In this work, we use MEDLINE abstracts and titles from 1975 to present day. Prior to 1975, only 2% of the citations contained an abstract. Figure 2 shows the percentage of MEDLINE citations over time. The percentage of publications that include abstracts steadily increases over time, with the exception of 2015, for which more abstracts will likely be included in the coming years.

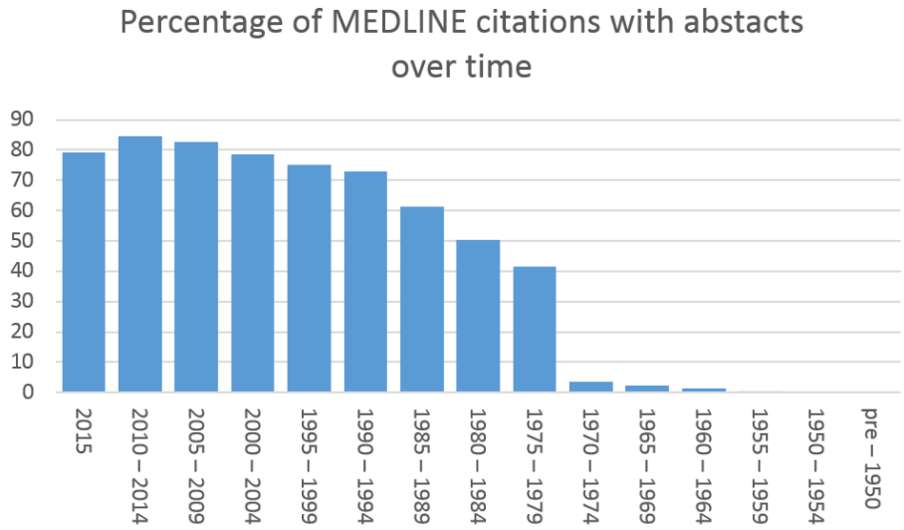

Fig. 2. The percentage of abstracts included with MEDLINE citations in five year intervals (as of the 2015 MEDLINE baseline)

[1] `https://www.nlm.nih.gov/bsd/medline.html`

2.1.2 PubMed

PubMed encapsulates MEDLINE, PubMed Central (PMC), and other article and reference repositories. MEDLINE is the largest subset of PubMed. PMC is a set of full-text biomedical and life sciences articles. Some PMC references overlap MEDLINE references, and all PMC references contain full text documents. Articles within PubMed can easily be searched manually via the web[2], and copies of the entire repository (yearly baselines) may be downloaded for processing[3].

2.2 Tools

2.2.1 Concept Hierarchies

A concept hierarchy or taxonomy is an organization of semantic concepts into a hierarchical or semi-hierarchical structure. They act as both dictionaries and thesauri, and play an important role in text normalization and the integration of human semantic knowledge into NLP tasks. Concepts are the basic units of a hierarchy. They encapsulate term meaning, and account for the syntactic and lexical variations of a term. Figure 3 shows an example of a concept hierarchy in which the words *feline, felines, cat*, and *cats* all map to the same concept of *Cat*. Concepts in the hierarchy are linked via primarily isA relationships. Concepts with the most narrow meanings are at the leaf nodes, and concepts with the most broad meanings are at the root. The example in Figure 3 shows that a *Cat* isA *Mammal* isA *Animal* isA *Organism* isA *Entity*. In this work we use several hierarchies from the biomedical domain, which are described in the next sections.

---

[2] `https://www.ncbi.nlm.nih.gov/pubmed`

[3] `https://www.nlm.nih.gov/databases/download/pubmed_medline.html`

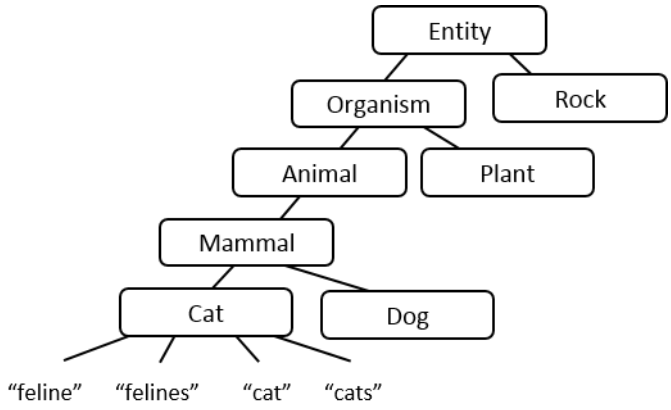

Fig. 3. An example concept hierarchy (taxonomy).

2.2.1.1 Unified Medical Language System (UMLS)

The Unified Medical Language System (UMLS) contains a Metathesaurus, SPECIALIST lexicon, and Semantic Network.

The Metathesaurus consists of over 3.1 million biomedical concepts. Nearly 200 vocabularies and taxonomies have been semi-automatically combined to form the Metathesaurus. This includes the Medical Subject Headings (MeSH) taxonomy, Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT), Online Mendelian Inheritance in Man (OMIM), Gene Ontology, and HUGO Gene Nomenclature Committee (HGNC). A complete list of combined vocabularies is available online[4]. Within the Metathesaurus, synonymous terms (including spelling variation) are mapped to a single concept, which is indicated by a unique code, a Concept Unique Identifier (CUI). Concepts are arranged in a semi-hierarchical structure, and concepts are linked via primarily parent/child and narrower/broader relationships. Concepts are categorized into one of 133 different semantic types (although the semantic types change over time). Semantic types are grouped together into 15 Semantic Groups to

[4]`https://www.nlm.nih.gov/research/umls/sourcereleasedocs/`

provide an even broader categorization of concepts[12].

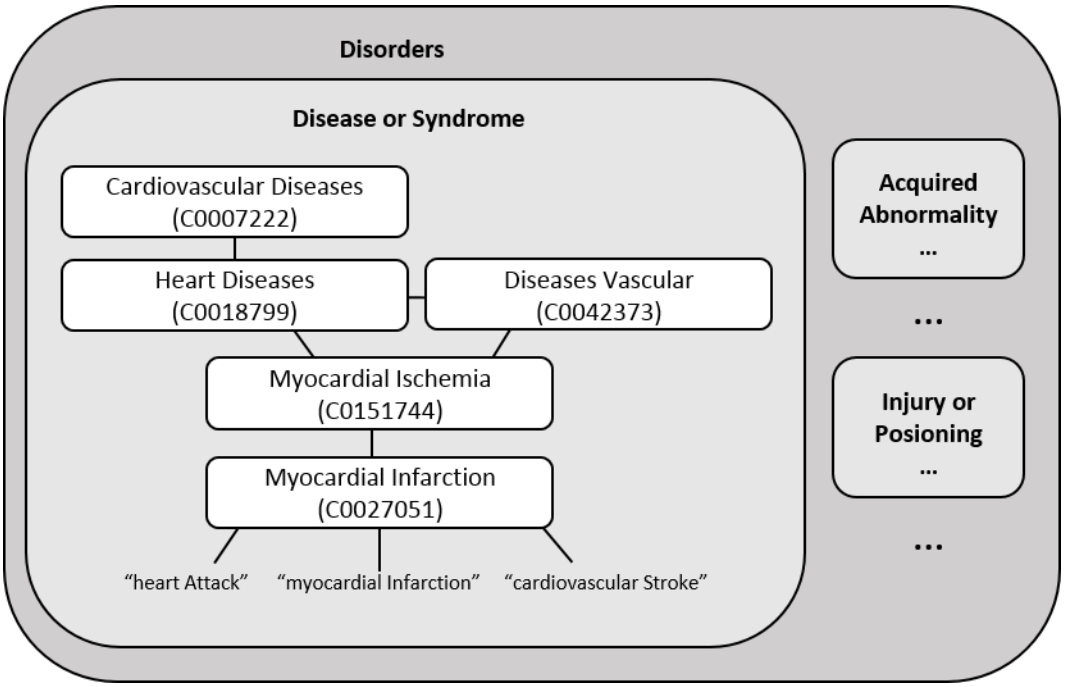

Fig. 4. A partial hierarchy of the UMLS. Multiple terms (*Heart Attack, Myocardial Infarction, and Cardiovascular Stroke*) are mapped to a single concept, *Myocardial Infarction* with a CUI of `C0027051`. *Myocardial Infarction* is linked to other concepts within the UMLS (*Myochardial Ischemia, Heart Diseases, etc.*) all of which have been assigned the semantic type, *Disease or Syndrome*. The semantic type, *Disease or Syndrome* is grouped with the other semantic types, *Acquired Abnormality* and *Injury or Poisoning* to form the semantic group, *Disorders*.

The UMLS semantic network contains semantic types arranged in a hierarchical structure that includes causal, and compositional relationships. The semantic network models high level relationships among semantic types such as *disrupts, part of,* or *contains*, but these relations do not always translate to individual concepts that are a part of the semantic type. For example, the semantic network indicates that *Clinical Drugs* treat *Diseases or Syndromes*, and that *aspirin* is a *Clinical Drug*, and *Cancer* is a *Disease or Syndrome*, but *aspirin* does not treat *cancer*.

The SPECIALIST Lexicon contains lexical information for commonly occurring English and biomedical words. Each entry in the lexicon contains lexical variations of a term, and their syntactic, morphological, and orthographic information.

2.2.1.2 Systematized Nomenclature of Medicine - Clinical Terms

The Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) is a comprehensive clinical terminology and taxonomy. In version 2016AA of the UMLS (the version we use), SNOMED CT contains 280,695 concepts with a PAR/CHD relation; 219,160 leaf nodes and 91,737 non-leaf nodes. The average depth of the taxonomy is 10, the average depth of a non-leaf node is 9.2 and the average depth of a leaf node is 10.9. The average branching factor of a non-leaf node is 5.1 and on average each node (concept) has 5.1 distinct paths to the root.

2.2.2 Text Processing Tools

In this work, raw text may be processed using one of several tools, MetaMap, SemRep, and Compoundify. Figure 5 shows examples of the output of each tool at a conceptual level (the actual output is more complex).

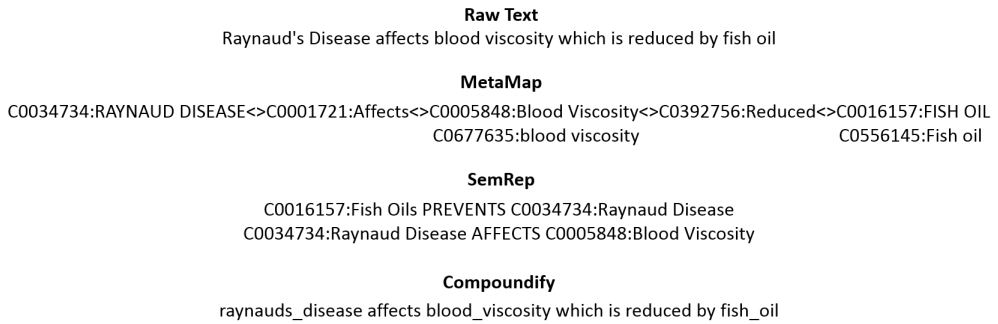

Fig. 5. The output of each of the text processors on the same raw text

2.2.2.1 MetaMap

MetaMap [13] is a tool developed by the National Library of Medicine, National Institutes of Health to map raw text to CUIs. From the raw text input, it produces a set of ordered CUI mappings. This has the effect of performing stop word removal and text normalization. MetaMap[5] is freely available and can be used through an interactive web interface, or downloaded for local use. MetaMap can be run on any text, but since MEDLINE is a popular dataset, MetaMap is run on MEDLINE to produce the MetaMapped MEDLINE baseline[6], which contains the text of all titles and abstracts of MEDLINE mapped to CUIs.

2.2.2.2 SemRep and SemMedDB

SemRep [14] extracts relationships from biomedical text as semantic predications in the form of subject-predicate-object triples. For example, Figure 5 shows that two relationships are extracted: `fish oils prevent Raynaud's Disease` and `Raynaud's Disease affects blood viscosity`. SemRep precision rates are between 73-96%, and recall rates between 55-70% [15] depending on the relation type. The accuracy of the extracted predicates was found to be 84% [16]. SemRep is designed for the biomedical domain, but efforts have been made to extend SemRep to the other domains [17]. SemRep is freely available[7], and can be run on any text. SemRep has been run on the entire MEDLINE baseline for which greater than 96 million semantic predications (as of June 30th, 2018) have been extracted to produce SemMedDB, a database of freely available predications[8].

[5] `https://metamap.nlm.nih.gov/MetaMap.shtml`

[6] `https://ii.nlm.nih.gov/MMBaseline/index.shtml`

[7] `https://semrep.nlm.nih.gov/`

[8] `https://skr3.nlm.nih.gov/SemMedDB/`

2.2.2.3 Compoundify

Compoundify combines multi-word terms in text using the UMLS SPECIALIST Lexicon as a glossary. The result is a corpus in which all compound terms have been identified (component words joined by an underscore). Compoundify is a standalone component in the word2vec-interface package[9].

2.2.3 Text::NSP

The Text::NSP package developed by Pedersen, et al. [18] is a freely available open-source software package that identifies n-grams, collocations, and word associations in text. It contains implementations of the association measures used throughout this work. It is implemented in Perl and takes advantage of regular expressions to provide flexible tokenization.

2.3 Distributional Semantics

Distributional semantics is the study of modeling word meaning through these contexts in which words appear. The distributional hypothesis states that words used in similar contexts have similar meanings. As stated famously by Firth [19], "You shall know a word by the company it keeps". The contexts of words are captured using training corpora, and are based on co-occurrences between words in text.

A problem fundamental to distributional semantics is the sparsity of co-occurrence data in text, which obscures term meaning. With direct (first order) co-occurrence vectors, each element in the vector corresponds to the count of co-occurrences between a term and another term. Most terms never co-occur with one another which produces a sparse co-occurrence vector containing many zeros. Several techniques

[9] `https://metacpan.org/pod/Word2vec::Interface`

have been developed to deal with the sparsity problem, and are explained in the next few sections.

2.3.1 Direct Co-occurrence Vectors

Direct (first order) co-occurrence vectors are the simplest distributional context vector to construct. They do not deal with the problem of sparsity, and serve as a baseline. Direct co-occurrence vectors have dimensionality of the vocabulary size, that is, each element of the vector corresponds to the count of co-occurrences between the term and the term at that index. Each term in the vocabulary has a single index in the vector. These vectors are constructed over a training corpus and record all co-occurrences of terms within a predefined window size. For example, consider the phrase in the previous sentence: "record all co-occurrences of". The vector for the word *record* would increment values at the vector indexes for the term *all* using a window size of 1, *all* and *co-occurrences* using a window size of 2, and *all*, *co-occurrences*, and *of* using a window size of 3.

2.3.2 Singular Value Decomposition

Singular Value Decomposition (SVD) is a factor analysis technique to decompose a matrix, M into a product of three simpler matrices, such that $M = U \cdot \Sigma \cdot V^T$. The matrices U and V are orthonormal and Σ is a diagonal matrix of eigenvalues in decreasing order. Limiting the eigenvalues to d, we can reduce the dimensionality of our matrix to $M_d = U_d \cdot \Sigma_d \cdot V_d^T$. The columns of U_d correspond to the eigenvectors of M_d. Typically this decomposition is achieved without any loss of information. Here though, SVD reduces a word-by-word co-occurrence matrix from thousands of dimensions to hundreds, and therefore the original matrix cannot be perfectly reconstructed from the three decomposed matrices. The intuition is that any information lost is

noise, the removal of which causes the similarity and non-similarity between words to be more discernible [20]. Singular Value Decomposition (SVD) is the factor analysis technique used within the commonly used method, Latent Semantic Index (LSI) [21].

2.3.3 Word Embeddings

Word embeddings are another, increasingly popular [22] dimensionality reduction method. Word embeddings construct reduced dimensionality distributional context vectors directly from a training corpus by iterating over it and learning word representations. The word embeddings method, *word2vec*, proposed by Mikolov, et al. [23], is a neural network based approach that learns a series of weights (the hidden layer within the neural network) that either maximizes the probability of a word given the surrounding context, referred to as the continuous bag of words (CBOW) approach (Equation 2.1), or to maximize the probability of the context given a word, referred to as the skip-gram approach (Equation 2.2). For either approach, the resulting hidden layer consists of a matrix where each row represents a word in the vocabulary and columns a word embedding. The basic intuition behind this method is that words closer in meaning will have vectors closer to each other in this reduced space.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{0<j\leq c;} log(p(w_t|w_{t-j})) \tag{2.1}$$

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c; j\neq 0} log(p(w_{t+j}|w_t)) \tag{2.2}$$

Equations 2.1 and 2.2 define the optimization functions for CBOW and skip-gram respectively. In these equations T is the number of words in the corpus, w_t indicates the word for which the window is located, w_i indicates the word at location i (words in the window), c indicates the window size, and p indicates probability.

Equation 2.1 can therefore be interpreted as: sum of the log probabilities of the word at location t given the previous $t - j$ words summed over all words in the corpus. Equation 2.1 can be interpreted similarly.

2.4 Semantic Similarity and Relatedness

Measuring the degree to which two terms are similar (e.g., *liver-organ*) or related (e.g., *headache-aspirin*) is a fundamental natural language processing task. We use these in LBD to estimate the relatedness between start and target term pairs. There are many ways to quantify the relatedness or similarity between terms [24], and measures of similarity often perform well for relatedness quantification tasks, and vice versa.

Semantic similarity is a subset of semantic relatedness, two terms may be related, but not similar (e.g., *headache-aspirin*), or both similar and related (e.g., *liver-organ*). Semantic similarity is based primarily on isA relationships, and as such, semantic similarity measures often exploit concept hierarchies or taxonomies such as the UMLS. From the simplest perspective, these measures are based on the path length between two concepts in the hierarchy (path [25]), but more complex measurements include weighting the edges with information content (lin [26], resnick [27]), incorporating the depth of the least common subsumer (lowest node the two concepts have in common) (wup [28]), or measuring path overlap (cmatch [29], batet [30]).

Semantic relatedness measures quantify a greater variety of relationships among terms than semantic similarity measures. They typically don't exploit taxonomic information, but instead are often based on the distributional semantics, and use vector representations of terms (Section 2.3). For this, a context vector is created for each concept and the similarity between the concepts is calculated by taking the cosine between their individual context vectors. Estimating semantic similarity and relat-

edness is a well established field with standard evaluation datasets and procedures. In this section we present these evaluation procedures and some previous work.

2.4.1 Evaluation Data

Similarity or relatedness measures are typically evaluated on several standard evaluation datasets. These datasets consist of term pairs for which a similarity or relatedness score has been assigned by human graders. The score indicates the degree to which the two terms are semantically related. We evaluate using the MiniMayoSRS and UMNSRS reference standard datasets. We use a version of the datasets for which each term is mapped to UMLS concepts in the SNOMED CT vocabulary.

MiniMayoSRS consists of 30 term pairs whose relatedness was determined by nine medical coders and three physicians from the Mayo Clinic. The relatedness of each term pair was assessed based on a four point scale: (4.0) practically synonymous, (3.0) related, (2.0) marginally related and (1.0) unrelated. The average correlation between physicians is 0.68. The average correlation between medical coders is 0.78.

UMNSRS, developed by Pakhomov, et al. [31], consists of 725 clinical term pairs whose semantic similarity and relatedness was determined independently by four medical residents from the University of Minnesota Medical School. The similarity and relatedness of each term pair was annotated based on a continuous scale by having the resident touch a bar on a touch sensitive computer screen to indicate the degree of similarity or relatedness. As suggested by Pakhomov and colleagues, we use a subset of ratings with higher Intraclass Correlation Coefficients (ICCs). This subset has an ICC of 0.73, and consists of 401 pairs for the similarity set (UMNSRS Sim), and 430 pairs for the relatedness set (UMNSRS Rel).

2.4.2 Correlation and statistical significance

We report the Spearman's rank correlation (ρ) between the automatically assigned relatedness scores and the gold standard scores, and use Fisher's R-to-Z transformation [32] to compute statistical significance between the correlation results.

Spearman's measures the statistical dependence between two variables to assess how well the relationship between the rankings of the variables can be described using a monotonic function. Spearman's correlation coefficient is based on the rank of terms rather than the scores assigned and can therefore be used with any scoring method, regardless of the range of scores applied. Using Spearman's correlation means a single number can be computed for the performance of each method, making them easily and quantitatively comparable across systems. Fisher's R-to-Z transformation [32] computes the statistical significance (p-value) between two correlation coefficients. We report ρ between the automatically assigned relatedness scores and the gold standard scores and consider a p-value of $p \leq 0.05$ to be statistically significant.

2.5 Association Measures

Association measures are based on co-occurrence statistics in a corpus. They quantify the association between two terms, which is a measure of the likelihood the terms co-occur together more often than expected. Association measures are based on terms' individual occurrence frequencies, and their mutual co-occurrence frequencies. As such, frequency counts of terms, and term pairs must be collected from a corpus. Traditionally bigram counts are used, but this can be extended to find co-occurrences within an arbitrary sized n-gram model, or window size.

Figure 6 shows an overview of the association calculation process. In this process, co-occurrence information is collected from a corpus and used to populate a contin-

gency table of values, these values are input into an association measure equation, such as Pearson's chi squared [33] (as shown), and a scalar association score is output.

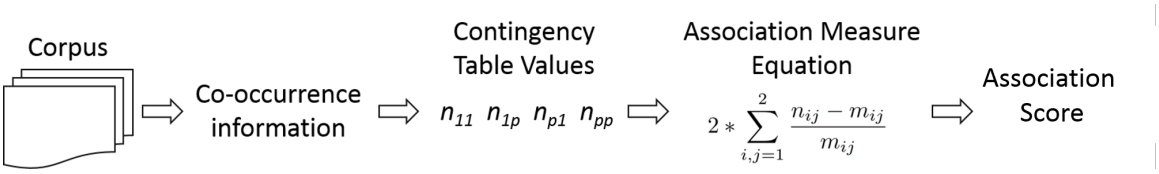

Fig. 6. An overview of how association measures are calculated. The equation shown is Pearson's chi squared, where m_{ij} values are: $m_{11} = \frac{n_{1p} * n_{p1}}{n_{pp}}$, $m_{12} = \frac{n_{1p} * n_{p2}}{n_{pp}}$, $m_{21} = \frac{n_{p1} * n_{2p}}{n_{pp}}$, $m_{22} = \frac{n_{p2} * n_{2p}}{n_{pp}}$ and $n_{2p} = n_{pp} - n_{1p}$, $n_{p2} = n_{pp} - n_{p1}$

To calculate an association score, first co-occurrence information is collected from a corpus. This co-occurrence information may be displayed in a contingency table for each term pair. Table 1 shows a contingency table for the generic term pair X and Y, and uses the standard notation, $\overline{X}$ and $\overline{Y}$ to indicate any token except X or Y respectively. $*$ indicates any single token. The cell n_{11} is the joint frequency of the term pair X, Y, the number of times tokens X and Y are seen together. The cell n_{12} is the frequency in which X occurs in the first position but Y does not occur in the second position, and the cell n_{21} is the frequency in which Y occurs in the second position, but X does not occur in the first position. The cell n_{22} is the frequency in which neither X nor Y occur in their respective positions. The cells, n_{1p}, n_{p1}, n_{2p} and n_{p2} represent the marginal totals which are the number of times a term does not occur in the first or second position of the term pair. Lastly, the cell n_{pp} is the total number of term pairs found in the corpus. It is important to note that all contingency table values can be calculated as sums and differences between just four values, n_{11}, n_{1p}, n_{p1}, and n_{pp}.

To illustrate the calculation of association scores, we use the tokens *stop* and *smoking* as example X and Y values. The value n_{11} indicates the count of all instances

	Y	$\overline{Y}$	totals
X	$n_{11} = XY$	$n_{12} = X\overline{Y}$	$n_{1p} = X*$
$\overline{X}$	$n_{21} = \overline{X}Y$	$n_{22} = \overline{XY}$	$n_{2p} = \overline{X}*$
totals	$n_{p1} = *Y$	$n_{p2} = *\overline{Y}$	$n_{pp} =**$

Table 1. A contingency table showing how the counts, n_{xy} are calculated for the generic term pair XY. $\overline{X}$ and $\overline{Y}$ indicate any token except X or Y respectively. $*$ indicates any single token.

of the term pair *stop smoking*. n_{12} indicates the count of all term pairs with *stop* as the leading token excluding any counts where *smoking* is the trailing token, and n_{1p} is the count of all term pairs beginning with *stop*. Table 2 is a contingency table with sample values for *stop smoking*.

	stop	¬ stop	totals
smoking	2955	75020	77975
¬ smoking	308792	2712312165	2712620957
totals	311747	2712387185	2712698932

Table 2. An contingency table of observed counts of the term pair *stop smoking*.

Using the observed co-occurrence counts in the contingency table, we can calculate the expected values which estimate the values one would expect to see based on the independence model. The independence model is the hypothesis that the words in the term pair happen to co-occur purely by chance. Using this model, the expected values are calculated as the product of the marginal totals divided by the total number of term pairs in the text. Table 3 explicitly defines the calculations, where m_{ij} is the expected value, and Table 4 shows the results of the calculations.

	Y	$\overline{Y}$	totals
X	$m_{11} = \frac{n_{1p} * n_{p1}}{n_{pp}}$	$m_{12} = \frac{n_{1p} * n_{p2}}{n_{pp}}$	n_{1p}
$\overline{X}$	$m_{21} = \frac{n_{p1} * n_{2p}}{n_{pp}}$	$m_{22} = \frac{n_{p2} * n_{2p}}{n_{pp}}$	n_{2p}
totals	n_{p1}	n_{p2}	n_{pp}

Table 3. A contingency table of expected value equations. n_{xy} values are retrieved from the contingency table of observed values (Tables 1 and 2)

	stop	¬ stop	totals
smoking	9	77966	77975
¬ smoking	311738	2712309219	2712620957
totals	311747	2712387185	2712698932

Table 4. An contingency table of expected values of the term pair, *Stop Smoking*.

Looking at Table 4, the term pair *stop smoking* would be expected to occur 9 times if the token pair occurred together by chance, but is observed 2,955 times (Table 2). Intuitively this indicates a strong association, and to quantify it, we use an association measure equation. A combination of expected and observed values are input into an association measure equation (e.g. Pearson's Chi-Squared [33]) to produce a single number that quantifies the association between two terms.

In this dissertation, we use association measure equations as implemented in Text::NSP [18]. Text::NSP includes a wide range of association measures that are grouped into five categories: Mutual Information, Fisher's exact Tests, Chi-squared, Dice, and Odds. Table 5 shows a listing of the association measures implemented in Text::NSP.

Although all association measure equations use expected and observed contin-

Class	Measure
Mutual Information Measures	Log Likelihood Ratio [33] *True* Mutual Information [34] Pointwise Mutual Information [34] Poisson-Stirling [35]
Fisher's Exact Test	Left Tailed Fisher [36] Right Tailed Fisher [36] Two Tailed Fisher [36]
Chi-squared	Phi Coefficient [37] T-score[38] Pearson's Chi-Squared [33]
Dice	Dice Coefficient [39] Jaccard Measure
Odds	Odds Ratio [40]

Table 5. Association Measures implemented in the Ngram Statistics Package of Text::NSP

gency table values as input, they quantify association in different ways. We give an overview of an association measure from each group in Table 5 below:

- The Log Likelihood Ratio [33] (G^2), shown in Equation 2.3, reflects the degree to which the observed and expected values diverge. A G^2 score of zero implies that the data fits perfectly into the hypothesized model, and the observed values are equal to the expected. A higher score indicates that the term pair is less likely to have appeared together by chance.

$$G^2 = 2 * \sum_{i,j=1}^{2} n_{ij} * log(\frac{n_{ij}}{m_{ij}}) \tag{2.3}$$

- The Fisher's Exact Tests [36] ($fisher$), shown in Equation 2.4, calculates the significance of observing a set of contingency table values. This is done by fixing the marginal totals, and exhaustively computing the significances of observing each possible set of table values that would lead to the observed marginal totals. This allows the tests to estimate any probability distribution.

$$fisher = \frac{1}{n_{11}!n_{12}!n_{21}!n_{22}!} * \frac{n_{1p}!n_{2p}!n_{p1}!n_{p2}!}{n_{pp}!} \tag{2.4}$$

- The Pearson's Chi-squared test (χ^2) [33], shown in Equation 2.5, G^2, measures the deviation between the observed data and expected data. The computed value compares the observed value to the predicted value under a chi-squared distribution. The greater the difference between the two values, the more likely the two terms are dependent.

$$\chi^2 = 2 * \sum_{i,j=1}^{2} \frac{n_{ij} - m_{ij}}{m_{ij}} \tag{2.5}$$

- The Dice Coefficient [39] shown in Equation 2.6, is a ratio between a term pairs co-occurrence frequency and the sum of their individual occurrences.

$$dice = 2 * \frac{2 * n_{11}}{n_{p1} + n_{1p}} \tag{2.6}$$

- The Odds Ratio [40] shown in Equation 2.7, computes the ratio of how often the term pair co-occurs together over the number of times they do not occur together.

$$odds = \frac{n_{11} * n_{22}}{n_{21} * n_{12}} \tag{2.7}$$

CHAPTER 3

LITERATURE BASED DISCOVERY

This chapter provides relevant background information related to the focus of this research, literature based discovery (LBD). First, we provide a motivation as to why LBD is important by covering its main application areas of drug discovery, drug repurposing, and adverse drug event prediction. Next, to better understand how our contributions fit into the LBD process, we define an LBD pipeline. We break LBD into a series of generalized components, and define the inputs, outputs, and goals of each. We give examples of each component to illustrate different implementations and ideologies. Next, we present literature reviews on our main areas of contribution. Since we apply indirect relatedness measures to ranking and filtering hypotheses (Chapter 5), we present a literature review on hypothesis ranking and filtering methods. Next, since we develop new evaluation methods (Chapter 6), we present a review of existing evaluation methods. We describe different evaluation methods, provide examples, and contrast them by defining a set of desirable characteristics of evaluation methods. Lastly, we present a literature review of visualization methods and of functional groups, relevant to our work in developing an interactive visual environment to explore LBD output (Chapter 7).

3.1 Motivation

LBD has led to countless discovery proposals ranging from treatments for cataracts [41], multiple sclerosis [42], and Parkinsons Disease [43], to understanding and discovering new health benefits of curcumin [44], finding potential treatments for cancer [45], find-

ing a link between Hypogonadism and Diminished Sleep Quality [46], and discovering obesity mechanisms [47]. Although not all of these proposals have been verified, drug development [8, 48, 9], drug repurposing [45, 49, 5, 48, 50, 51, 52, 53], and adverse drug event (ADE) prediction [49, 54, 55, 16, 56] are perhaps the most promising and prevalent application areas of LBD. These applications can save millions of dollars, and save lives by bringing new drugs to the market faster, and preventing fatal ADEs. Since new drug development, drug repurposing, and ADE prediction are common application areas, we briefly cover them here to provide real-world motivation that goes beyond the intrinsic scientific value of LBD.

Drug Discovery: A new drug costs between 500 million and 2 billion dollars to develop, and can take between 10 to 15 years [51] to come to market. The success rate is less than 10% [51], and the number of new drugs approved by the FDA is declining [49]. LBD can provide a deeper understanding of drug mechanisms, interactions, and side effects. This can lead to a better understanding of physiological functions, leading to, and reducing both the cost and time for new drug development. Much of the work with LBD and drug discovery has focused on incorporating genetic microarray information into LBD systems [8, 48, 9]. Incorporating microarray data is promising because it adds empirical evidence to support hypotheses generated from literature. Hu, et al. [9] correlate microarray analysis of genes and diseases with the strength of those relationships in literature. Hristovski, et al. [8] incorporate microarray correlations into discovery patterns found in literature, and Zhang, et al. [48] identify potential prostate cancer drugs using a combination of semantic relations found in literature and microarray data. These systems explain links between drugs, genes, and effects, but may also be used to explain protein interactions as well [53]. Both proteins and genes affect disease development and progression, and can be targeted by drugs and chemicals. Understanding this process can lead to the

next generation of drugs, saving lives and reducing cost.

Drug Repurposing: Drug repurposing, which is the process of finding new applications for existing drugs. Drug repurposing is on the rise, accounting for "approximately 30% of the new US Food and Drug Administration approved drugs and vaccines in recent years" [51]. Classic examples of drug repurposing include Viagra, which was developed as a treatment for angina, and was repurposed to treat erectile dysfunction; Rogaine, originally developed for high blood pressure, found success as a baldness treatment [57]; Topiramate, an anti-epileptic drug was developed to treat obesity, and Prozac, an anti-depressant was developed to treat premenstrual dysphoria [49]. Although LBD did not play a role in these repurposings, LBD is increasingly being used towards that goal [45, 49, 5, 48, 50, 51, 52, 53]. Currently there are about 4,000 drugs approved for human use, and about 5,000 more drugs registered for investigational use [5]. Many of the investigational drugs have been extensively studied and satisfy basic regulatory requirements. By applying LBD to drug repurposing, drug development costs may be reduced by up to 50%, and bring drugs to market much more quickly [49]. LBD is useful for drug repurposing because it yields a better understanding of the biological effects of a drug, and may be used to evaluate a drug's benefit/risk profile. This may allow one to arrive at novel discoveries [49]. As of 2011, drugs developed using LBD are in the preclinical stage [49].

Adverse Drug Event Prediction: LBD provides a better understanding of drug mechanisms and side effects, and in a similar way that this knowledge can be applied for drug repurposing, it can also be applied to adverse drug event (ADE) prediction [49, 54, 55, 16, 56]. Adverse events can be caused by normal use, misuse, or sudden discontinuation of medications. ADEs often lead to hospitalization, and account for an estimated 12% of all emergency room visits [54]. Furthermore, the number of serious or life-threatening ADEs is increasing [49]. ADEs pose significant

health and financial problems worldwide [16].

Since LBD can explain drug mechanisms and side effects it makes ADEs more easily predicted and avoided. A recent study by the Food and Drug Administration [58], found that ADE prediction systems were able to predict many life-threatening cardiac-related ADEs, and anticipated that development of similar technologies are in line with their initiatives, and will be helpful tools in the future. Unforeseen ADEs may occur after drugs are released to the market, and LBD allows for early detection of these ADEs through automated analysis of literature and clinical notes. By quickly identifying ADEs both safety and quality of patient health care increase [54].

3.2 Components

Modern LBD systems consist of many components, and there is a wide variety of LBD systems. Although two systems may appear very different on the surface, they share many underlying components, strategies, and goals. In this section, we divide the LBD process into a series of well defined, independent, and generalized steps. We describe the inputs, outputs, and goals of each step to create a generalized LBD framework. We use this framework to identify areas of LBD that need improvement, and in Chapter 6, assign evaluation methodologies for individual steps and sets of steps. By breaking LBD into a series of smaller pieces, each step can be quantitatively evaluated and compared, developed in isolation, and components can be shared among LBD systems. Figure 7 shows our generalized LBD framework.

There are two main ways to perform LBD, open-discovery and closed discovery [59] (also called one-node search and two-node search respectively [60]). In open discovery, the user inputs a start term, and the system outputs potential hypotheses. In closed discovery, the user inputs both a start term and a target term, and the system outputs a set of linking terms, or reasons why a hypothesis may be true.

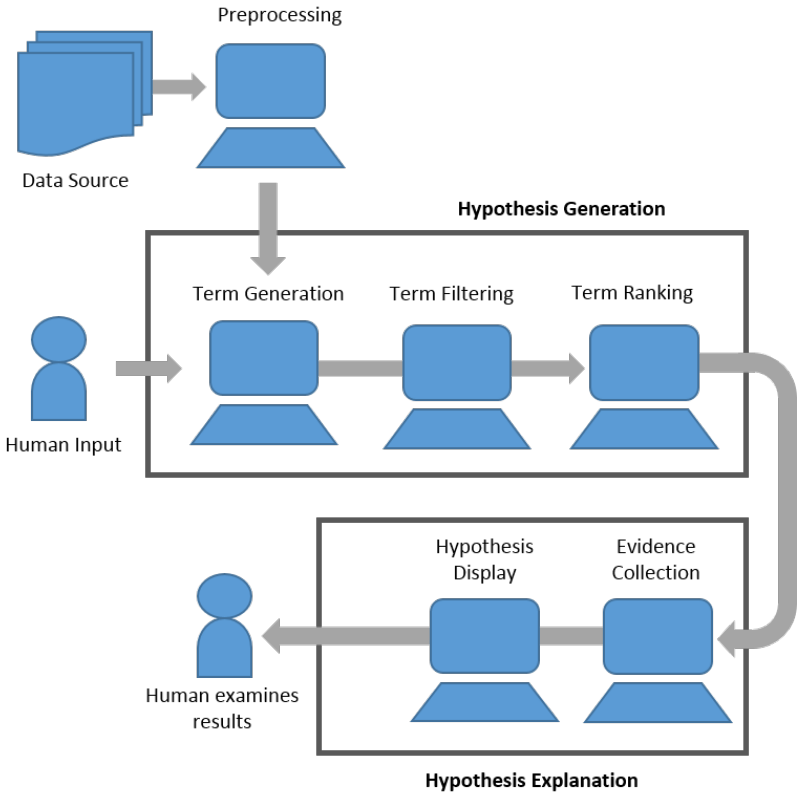

Fig. 7. The LBD process as a series of generalized steps. First data is preprocessed, then, hypothesis generation is performed, in which target terms are generated, filtered and ranked. Lastly, evidence to support the discovery is collected and the hypotheses and their evidence is displayed to the user.

Open discovery is used to generate new discoveries, where as closed discovery is used to explain hypotheses. The contributions of this dissertation focus primarily on open discovery, which takes place in the hypothesis generation step of our framework. These generated hypotheses are fed into the hypothesis explanation step, in which closed discovery may be performed. Each step of our framework is explained below.

Preprocessing: The goal of preprocessing is to convert data from its raw form to a form accepted by the hypothesis generation step of LBD. It is often tightly coupled with the data source, and includes, among other things, text normalization,

stop word removal, collecting co-occurrence information, named entity recognition, and relation extraction. Generally speaking, preprocessing methods are not unique to LBD.

Preprocessing is performed on one or more data sources, often a corpus, set of relations, or database. Popular datasets include MEDLINE, the MetaMapped MEDLINE baseline, and SemMedDB, but other sources, such as the Therapeutic Target Database (TTD) [50], Comparative Toxicogenomics Database (CTD) [50], and microarray data [8], have been included. The use of innovative datasets presents promising directions of future research [61].

Hypothesis Generation: The goal of hypothesis generation is to create hypotheses in the form of start-target term pairs. Hypothesis generation consists of three sub-steps, *target term generation*, *term filtering*, and *term ranking*. Hypothesis generation is unique to LBD, and the methods, inputs, and outputs vary system by system. At its core though, hypothesis generation has the goal of generating a set of potential hypotheses, which can be represented as a list of target terms. This generalized model of hypothesis generation is shown in Figure 8, where a start term(s) and a data source are input, and a ranked target term list is output.

We illustrate the variety of hypothesis generation methods by describing a few examples and how they fit into our generalized framework.

Yetisgen-Yildiz and Pratt [62] use co-occurrence information collected from MEDLINE, and a starting term of interest as input. Their term generation step is based on the ABC co-occurrence model of LBD. A to B co-occurrences are found, spurious B terms are removed, and B to C co-occurrences are found to produce a list of potential target terms. No target term filtering step is applied. A target term ranking step is applied to compare the ranking algorithms of average minimum weight [63], literature cohesiveness [64], and a linking term count [60]. The final output is a ranked list of

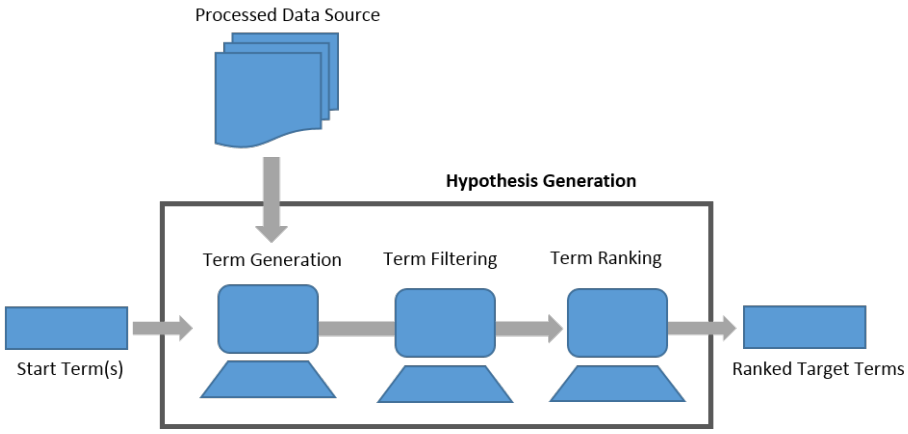

Fig. 8. A generalized model of the hypothesis generation process. Target terms are generated from a data source and start term, filtered and ranked to produce hypotheses as a ranked set of target terms

target terms.

Cohen, et al. [65] perform hypothesis generation in a vector space created from SemRep predications (Predication-based Semantic Indexing). Using the start term of interest, a term generation step consisting of a "logical leap", in which target terms are generated directly from the start term via a nearest neighbor search. Term filtering consists of eliminating terms that occur more than 100,000 times. Target terms are ranked by the cosine distance between the start-target vector mean, and its nearest neighbor (a linking term). Target terms whose start-target vector mean have the smallest cosine distance between itself and a linking term are ranked the highest. The final output is a ranked list of target terms, and their linking term.

Workman, et al.'s Spark system focuses on user interaction. For its hypothesis generation step (they call it, "initial output"). A user enters a search query (which includes a start term), from which a maximum of 10,000 semantic predications from 5000 of the most recent SemMedDB citations are retrieved. The term generation is the retrieval of predications. The term filtering step is the restriction to 10,000 of

the 5000 most recent predications. Next, the user selects the desired target term ranking method (they call it "retrieval affordance"), and the top ranked target terms (and their relationship to the start term) are displayed to the user, and the process is repeated.

Term Generation: The term generation step creates a term list of potential hypotheses related to the user's start term(s). This is the linking step, where target term are generated. Examples of term generation include the ABC model [60, 59], discovery patterns [66], vector-based nearest neighbor searches [67, 68], discovery by analogy [69], bibliometric linking [70], user interaction [71], or simply returning all terms in the vocabulary [72]. Each of these methods generate hypotheses in different ways, but all use a start term and preprocessed data as input, and all output a set of target terms. Most generate additional information, but that information is typically used to rank or explain a hypothesis. To restrict the scope of terms output by term generation, semantic type filters [59] and relationship-type filters [66] are popular. These reduce the number of target terms generated based on their semantic type (e.g. Drug, Disease, etc..) or the relationship type connecting them to the start or linking terms (e.g. reduces, affects, etc..).

Figure 9 shows a generalized model of term generation. Ideally the target term list output correctly identifies all possible discoveries between the start and target terms, and no false discoveries. This is, however unlikely, so term filtering and ranking steps are applied using the generated target term list as input.

Term Filtering: Term filtering removes false start-target term pairs from the list of target terms generated in the term generation step. Examples of filters include term occurrence rate, where too frequently [73, 74, 75] or too infrequently [76] occurring terms are removed; word sense disambiguation [77] to remove multiple meanings of a token, removal terms that are too broad [74, 78] or too similar to the

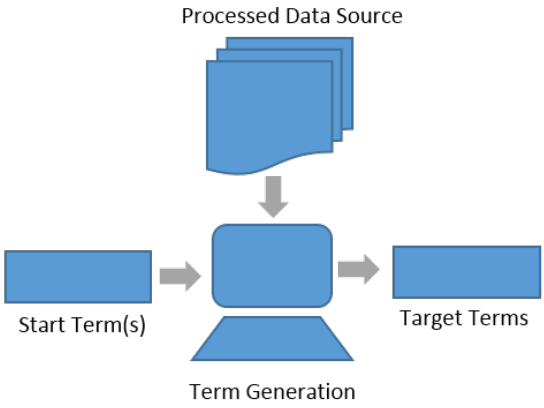

Fig. 9. The term generation step takes a start term of set of start term as input and a preprocessed data source to produce a set of target terms.

start term [79, 75]; or ranking and thresholding, using Term Frequency-Inverse Global Record Frequency [80], Term Frequency-Inverse Document Frequency (TF-IDF) [81], or other ranking and thresholding methods such as those discussed in Section 3.3.1.

Figure 10 shows the input and output of the term filtering step. We can frame term filtering as a binary classification task, which is input a set of target terms, and outputs a set of true terms, and a set of false terms, for which the start-target term pair constitutes a true or false hypothesis. The set of true terms continues as the input to the term ranking step, and the false terms are removed from further consideration.

Target Term Ranking: Target term ranking takes a list of target terms as input, ranks them based on some criteria, and outputs the ranked list. The ranking criteria may be interestingness of the start-target term pair, the likelihood that it is a true discovery, or the strength of the relationship. More generally, the goal of target term ranking is to assign a real-valued number to the start-target term pair. This is distinct from term filtering, for which a binary true/false value is assigned to each start-target term pair.

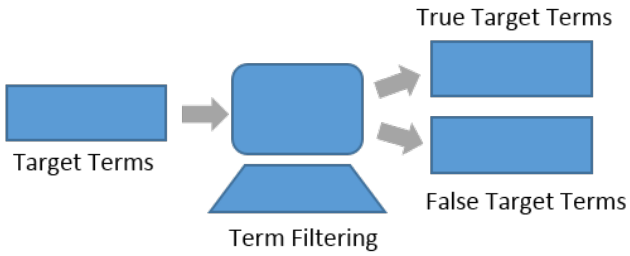

Fig. 10. The term filtering process takes as input a set of target terms representing possible discoveries, and classifies them as true discoveries or false discoveries (noise).

The need for target term ranking is motivated by computational or user-time requirements of downstream tasks. Hypothesis explanation is generally more computationally intensive than hypothesis generation, and only a small number of potential hypotheses may be analyzed by a user. It is critical that a concise list of discoveries is generated.

Examples of target term ranking algorithms include Linking Term Count (LTC) [60], Average Minimum Weight (AMW) [63, 62], X to Z support [82], Predicate Interdependence [51], vector-based ranking methods where, a term or concept vector representation is constructed, and a score is generated using cosine distance [83], Euclidean distance [84], or information flow [84] between the A and C terms, Graph-based ranking methods have been proposed, which construct co-occurrence graphs, and rank hypotheses based on the graphs' characteristics, such as degree centrality [76], or graph proximity metrics [85].

Target term ranking takes as input a list of true discoveries, and outputs a ranked list of discoveries, for which the rank and score indicates the likelihood a relationship exists, or a score of interestingness.

Hypothesis Explanation: The goal of hypothesis explanation is to collect evidence to support a hypothesis, and display that evidence to the user in an under-

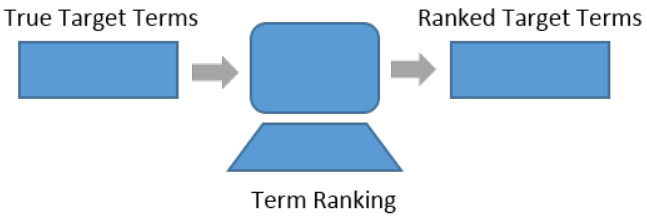

Fig. 11. An unordered set of true target terms is input into the term ranking step, which outputs a ranked list of target terms.

standable manner. In the first step of this process, *evidence collection*, any evidence to support the hypothesis is found. This may consist of finding linking, "B" terms, finding semantic predications that explain the hypothesis, finding article IDs or text snippets, or adding any other information to support and explain the hypothesis. In the next step, *hypothesis display*, that evidence is displayed to the user. How it is displayed is heavily dependent on the type of evidence collected. It could be as simple as a static ranked target term list, or as complex as an interactive explanatory graph of linking terms, semantic predications, and text snippets. Creating more informative output for LBD is critical for its adoption in real-world environments. Outputting just a list of target terms is insufficient, and more information should be provided.

3.3 Literature Review

With this understanding of LBD components, we present a literature review on the topics most relevant to the contributions of this dissertation. We include a literature of hypothesis ranking and filtering methods, a comparison between different evaluation metrics, and overviews of visualization systems and functional groups in LBD.

3.3.1 Hypothesis Ranking and Filtering

The number of hypotheses generated by LBD systems is usually too large to be manually reviewed, so ranking them is critical. This over-generation of knowledge is due, in large part, to the "small world" problem [63], which states that a start term will most likely co-occur with a highly connected term. Since a highly connected term co-occurs with many terms, within the ABC term generation paradigm, the "B implies C" step will create a set of target term hypotheses that approaches the vocabulary size. Systems that do not rely on the ABC paradigm [71, 86, 11] also over-generate knowledge, and require some sort of term ranking for thresholding and prioritizing what is displayed to the user.

There have been a variety of techniques to limit the target term list. These include: eliminating B terms via ranking and thresholding [80, 62]; applying semantic type filters to restrict B and C terms to informative semantic types (such as "drugs" or "diseases") [59]; using hierarchical filters to remove terms that are too general [74, 78]; or using relation extraction techniques and relation-type filters (such as keeping only "treats" or "increases" type relations) [66]. However, even with these techniques, too many terms are generated, and a term ranking method is required.

Target term ranking methods estimate the interestingness of the start-target term hypothesis. Ranked terms may be eliminated by applying a threshold, or ordered and displayed to the user. For LBD, it is assumed that target terms never co-occur with the start term (since they are new knowledge), meaning that traditional information retrieval ranking methods, which require direct co-occurrences cannot be applied. Instead, indirect ranking measures must be used. Methods designed specifically for LBD include: Linking Term Count (LTC) [60] which counts the number of unique B (linking) terms between A and C; Average Minimum Weight (AMW) [63, 62]

calculates the mean of minimum mutual information from A to B and B to C for all A to B to C pathways; X to Z support [82], which sums the weights of all ABC paths between A and C; and Predicate Interdependence [51] which ranks drug-disease pairs based on drug-gene and gene-disease predicate interdependence in literature.

Vector-based ranking methods have also been used in LBD. In these cases, a term or concept vector representation is constructed, and a score is generated using cosine distance [83], Euclidean distance [84], or information flow [84] between the A and C terms. The method in which vectors are created varies by LBD system. Bruza, et al. [84] construct vectors in Hyperspace Analogue to Language (HAL) space, Cohen, et al. [69] construct vectors using Predication-Based Semantic Indexing (PSI), and Sybrandt, et al. [87] construct vector representations using FastText [88] (a word2vec implementation).

Graph-based ranking methods have also been used. Graph-based methods construct co-occurrence graphs, and rank hypotheses based on the graphs' characteristics, such as degree centrality [76], or graph proximity metrics such as probability of best path, network reliability, expected reliable distance, or variations of random walks [85]. More recently, Kastrin, et al. [89] using the graph proximity metrics of Jaccard's Coefficient - a ratio of common neighbors to total neighbors, and Adamic/Adar metric - which uses weighted counts of shared neighbors, such that lower connected neighbors receive a higher weight. Sybrandt, et al. [87] propose and evaluate several ranking methods for LBD which include concept embeddings and topic network graph based metrics, and a combination of these methods.

Little work has been done in comparing these different ranking methods, but Yetisgen-Yildiz and Pratt [62] perform a comparison between a few, including LTC and AMW, and find that LTC is best performing hypothesis ranking method evaluated. For LTC, a linking term is defined as any term for which both the start

and target term co-occur with, and LTC is the number of terms they both co-occur with. Start-target term pairs with more linking terms are ranked higher than those with less. Sybrandt, et al. [87] propose and evaluate several ranking methods, including both vector and graph based measures. They find PolyMultiple is the best performing, which is a weighted average of all the methods evaluated.

Although most term ranking methods focus on finding the most probable, or highest scoring start-target term pairs, Smalhesier [61] argues that the most probable term pairs are often uninteresting, and instead it is the discoveries that, at the time, seem the least probable are the most interesting. This rarity principle is incorporated into several LBD systems including Epiphanet [90] and Spark [71] which are interactive systems with an option to find more rare or surprising terms. Other systems [91, 70, 92] focus on finding infrequently co-occurring terms rather than frequently co-occurring in the term generation and term ranking steps. RaJoLink [91] first finds rare terms in the starting term literature. Several of the rare terms are selected, and common terms within the selected rare term's literatures are found which forms a set of target terms. Later work [92] focuses on finding document outliers.

3.3.2 Evaluation

Evaluating LBD systems is difficult. Researchers disagree on many of the fundamental goals of LBD, and how to best facilitate discovery. Some points of contention include the level of automation, what constitutes a discovery, how to convey the results, and whether rarely or frequently co-occurring terms make the best discoveries. These points of contention make the overall goal of LBD difficult to define, and therefore to quantitatively evaluate. Even if these points of contention were resolved, accurately defining a gold standard dataset of all possible future discoveries is impossible, and estimating one is difficult. Additionally, the intended applications of

LBD systems vary, leading to a lack of standardized datasets. These disagreements and differences mean that many different evaluation strategies have been proposed. In the next subsections, we describe evaluation strategies that are popular, useful, or relevant this work. We describe each method, and its strengths, and weaknesses. We also define a set of criteria that LBD evaluation methods should meet, and determine which of the criteria the evaluation methods meet (summarized in Table 6). Our criteria states that an ideal evaluation method should be:

1. Automated - it is not a manual process, and should be easy to calculate in a reasonable amount of time. It is scalable.

2. Replicable - it is objective, and can be replicated with different LBD paradigms. Preferably it is based on a fixed dataset.

3. Quantifiable - it provides a numeric performance metric, preferably a single number that allows comparisons between systems.

4. Informative - it provides a deeper understanding of the behavior of the system or component.

5. Modular - it does not rely on the entire LBD pipeline, instead it evaluates individual components or sets of components in isolation.

Evaluation Methods

	Automated	Replicable	Quantitative	Informative	Modular
Discovery Replication	X	X	X		
User Studies				X	X
New Discovery Proposal					
Time-Slicing	X	X	X	X	X
ROC Curves	X	X	X	X	X
Established MI Graphs	X	X	X	X	X

Table 6. Evaluation methods and which of the ideal evaluation criteria they meet

3.3.2.1 Discovery Replication

Description: The goal of discovery replication is to reproduce a historic discovery, most commonly, Swanson's famous Raynaud's Disease - Fish Oil Discovery [3]. Discovery replication is simulated using data and knowledge available prior to the replicated discovery's publication date. If a system is capable of making the discovery, discovery replication is considered a success. This may mean simply the presence or absence of the target term of interest in the system's output, or that it is ranked sufficiently high in the output. Reporting the rank of the target term makes this method more quantitative.

Example: Swanson's Raynaud's Disease - Fish Oil Discovery was made in 1986, so data published prior to 1986 is input into a system. The start term of Raynaud's Disease is input into the system, and the system will output a list of target terms. If fish oil (the target term of interest) is present in this list, or if its rank in the output target term list is subjectively determined to be high enough, the discovery replication is considered a success.

Strengths: Discovery replication can be *automated*, is easy to compute, and is *replicable* by nearly (if not all) LBD systems. The results are easily understood, and clearly demonstrates a system's ability to perform at least one discovery. When the rank of the discovery is reported, discovery replication is *quantitative*.

Weaknesses: Discovery replication, is a narrow, constrained task. Only a single discovery is reported, and even when multiple discovery replications are made (Cameron, et al. [93] report fourteen) it is still a very constrained task. Since the task is so narrow, it is prone to overfitting, and when ranks are reported, it is often an unstable metric - the rank may vary widely even with small parameter adjustments. All this means discovery replication is *not informative*; it provides little insight into

how a system is working or how it may be improved. When reporting just the presence or absence of a discovery in the target term list, discovery replication is not quantitative. Lastly, discovery replication is *not modular*, generally speaking it evaluates the performance of the entire LBD process.

Conclusion: Although discovery replication may be interesting when used as an example or case study, it is a poor evaluation strategy, and insufficient on its own.

3.3.2.2 User Studies

Description: User studies determine what users like and dislike about a system, how a system is used, and how it can be improved. They are performed by creating questionnaires for, monitoring, or interviewing users of an LBD system. User studies are particularly important for user-centric systems, where interaction with the user plays a key role in discovery [71, 90], and for systems with complex or visual output [10].

Example: User studies were used in the development of the Arrowsmith system. In one study [94], researchers are observed using Arrowsmith, and based on the users' needs they improve the user interface, and develop and incorporate tools users' identified as critical for facilitating discovery. In another study, Smalheiser, et al. [95] perform a five year study involving a diverse group of voluntary researchers. Researchers filled out notebooks and described their use of Arrowsmith; weekly phone calls were made to monitor their progress. This, combined with "unsolicited" suggestions from web users were used to improve the web-interface, guide development of their system, and discovered novel ways the system was being used.

Strengths: User studies are critical for understanding how LBD systems are actually being used. Adoption of LBD tools in their intended domain is critical, and user studies ensure the LBD tools we develop are both usable and useful. All of this

makes user studies *extremely informative*. Additionally, they are (arguably) *modular*; they can be used for evaluating the user interface and how results are displayed in LBD systems.

Weaknesses: User studies are subjective and *not quantitative*, they consist of opinions and qualitative evidence. Their reliance on human users means they are inherently *not automated* and *not replicable*.

Conclusion: User interfaces and how the results are displayed are a critical component of LBD systems, and user studies are therefore a critical evaluation component. They are subjective by their very nature, making them not replicable, but they provide invaluable insight into how systems are used, and how they can be improved. They can reveal interesting ways LBD systems are being used to drive the development of tools that will have translational impact. User studies are a necessary evaluation metric and should be used in conjunction with other, quantitative evaluation metrics.

3.3.2.3 New Discovery Proposal

Description: New discovery proposals are the publication of discoveries made using an LBD system. New discoveries are best published in a journal of the intended domain, which lets experts validate the discovery. Often though, new discoveries and expert evaluation accompany a system description published in a journal more relevant to LBD researchers.

Example: LBD has produced many different discoveries, with varying level of empirical evaluation and expert vetting. This ranges from clinical trials [96], in vitro testing [4, 5], in vivo testing [6, 7], finding support through mircoarray analysis [8, 9], or expert testimonial validating the process [10, 11]. Specific examples include:

- Clinical trials - DiGuacomo, et al. [96] test Swanson's Raynaud's Disesase - Fish

Oil Hypothesis in a clinical trial.

- In Vitro - Fritjers, et al. [4] confirm in-vitro, their predicted associations between compounds and cell proliferation. Cohen, et al. [5] confirm their predicted therapies for prostate cancer in-vitro with cell cultures.

- In Vivo - Wren, et al. [6] perform in-vivo testing of their predictions of compounds affecting the development of cardiac hypertrophy with rodent models. Lekka, et al. [7] perform in-vivo experimentation to support their treatment for Multiple Sclerosis.

- MicroArray Support - Hu, et al. [9] use microarray and proteomic data to confirm hypothesized associations between specific genes and breast cancer. Hristovski, et al. [8] use microarray data to support their hypotheses on Parkinson's disease.

Strengths: New discovery proposal show the efficacy of an LBD system, and is critical for translational adoption, particularly when published in journals of the application domain. The use of a system to produce discoveries and endorsement by a domain expert validates the system. It exposes LBD to that community and gives LBD credibility.

Weaknesses: The reliance on domain experts for empirical evaluation means that new discovery proposal is *not automated* and *not replicable*. New discovery proposal is *not quantitative*, and relies on the entire system, so is *not modular*. Although they show a system works, they don't provide any insight into how a system works. They are *not informative* with respect to evaluation of individual system components or improving the system as a whole.

Conclusion: New discovery proposal with empirical evaluation is not really an

evaluation metric. It shows a system works, provides example applications, and most importantly, it is critical for the adoption of LBD systems in translational work, but does not provide any quantitative information on how to improve or compare LBD systems. New discovery proposal is vital and should be used in conjunction with quantitative evaluation metrics.

3.3.2.4 Time-Slicing

Description: In time-slicing evaluation [62], a cutoff date is selected to divide the data into training and test sets. The pre-cutoff (training) dataset is used to generate hypotheses, and the post-cutoff (test) dataset is used to estimate a gold standard of all future discoveries. The gold standard is estimated using co-occurrences [62] or extracted relationships [75] that occur in the post-cutoff dataset and do not occur in the pre-cutoff dataset (therefore indicating new knowledge). Using the hypotheses (the predicted discoveries) and gold standard discoveries, evaluation metrics are computed such as: precision and recall graphs over time [79], average precision at K [62], average interpolated precision graphs [62], mean average precision [62], and F-measure [75]. These metrics may be calculated using a single starting term (e.g. multiple sclerosis [97]), several starting terms (e.g. Alzheimer's disease, migraine, and schizophrenia [79]), averaged over 100 randomly selected terms [62], or using all terms in the vocabulary [75].

Example: Yetisgen-Yildiz and Pratt[62] proposed time-slicing evaluation. They use a cut-off date of January 1, 2000, meaning the pre-cutoff dataset consists of all publications before January 1, 2000, and the post-cutoff dataset consists of all publications after January 1, 2000. They use co-occurrences between MeSH descriptors to indicate relationships, and the gold standard set is created by finding all co-occurrences in the post-cutoff dataset and not in the pre-cutoff dataset. 100 randomly

selected diseases and the pre-cutoff dataset are input into their LitLinker system [74] resulting in list of target terms for each of the 100 diseases. The evaluation metrics of average precision at K, average interpolated precision graphs, and mean average precision [62] are calculated to quantify performance.

Strengths: Time-slicing evaluation is an *automated*, large scale evaluation that provides *quantitative* and *informative* metrics. It is *replicable* and is *modular* when used to evaluate individual components of an LBD system. Gold standard datasets are necessary for large scale, quantifiable evaluation of LBD systems, and time-slicing is one of the only methods of generating them.

Weaknesses: Time-slicing evaluation has been criticized because gold standard datasets are noisy; they contain false discoveries, or omit true discoveries. Gold standard datasets are difficult to estimate. Using co-occurrence information will produce many false positive relationships, and to increase the precision of gold standard datasets, Preiss, et al. [75] create use semantic predications rather than co-occurrences to constitute a relationship. They use multiple parsers, and state that if a relationship is extracted by any two, or all three parsers, then it is more likely a true relationship. Yang, et al. [50] create a highly precise gold standard using a manually curated database, the Comparative Toxicogenomics Database (CTD). Both of these methods, however, increase precision at the expense of recall - many relationships are missed.

There is also a lack of standard evaluation procedures once the gold standard has been evaluated. A variety of evaluation metrics have been proposed, including, precision at K [62], average interpolated precision graphs [62], Mean average precision (MAP) [62], and F-measure [75]. Different papers report evaluation metrics with different datasets making comparison between systems difficult to impossible.

Conclusion: Time-slicing is a valuable method for estimating gold standard datasets, but no gold standard evaluation datasets have been widely adopted, and

the method in which a gold standard is generated is not standardized. Similarly there is no standard evaluation procedure once a time-sliced gold standard has been generated. A standardized technique and dataset should be developed and adopted to make comparison across systems possible. Time-slicing type techniques are necessary for estimating gold standard datasets, but evaluation procedures must be standardized.

3.3.2.5 Receiver Operating Characteristic (ROC) curve Analysis

Description: Receiver operating characteristic (ROC) curves plot the trade-off between true and false positive rates, typically to evaluate the performance of binary classifiers. They require a dataset with true and false samples, and a method of assigning true and false labels to each sample. ROC curves are generated by varying some parameter (typically a threshold) and calculating the true and false positive rates at different values of this parameter. The area under the ROC curve (AUROC) can be calculated to provide a single number to summarize a system's performance. ROC curves have been used in LBD to evaluate the performance of target term ranking algorithms [85, 89, 87] and link prediction methods [85, 89].

Example: Eronen, et al. [85] frame the evaluation of their BIOMINE system as a link prediction task. In a time-slicing manner, they generate 500 links that occur (positive samples) in a "future" network of protein interactions and gene-pairs, and generate 500 links that never occur (negative samples) in that network. They evaluate the performance of different graph proximity measures and classifiers for link prediction. Kastrin, et al. [89] perform link prediction on a co-occurrence network of MeSH descriptors in a time-slicing manner. They use the entire co-occurrence network and evaluate both supervised and unsupervised approaches, including target term ranking measures. ROC curves and AUROC are reported. Sybrandt, et al. [87] divide a dataset of SemRep predications into pre- and post-cutoff segments. From

the post-cutoff dataset, they extract a set of published predications, and a set of highly-cited predications (published predications coming from articles that are cited more than 100 times). The published and highly-cited predications form two sets of true samples, where highly-cited predications are more confidently true but contain fewer samples. They create a set of false samples by creating "noise" predications, predications that do not occur in either the pre- or post-cutoff datasets. They pose evaluation as a noise discrimination task, and evaluate several target term ranking algorithms. ROC curves and AUROC are calculated for *noise* vs *published*, and *noise* vs *highly-cited* datasets.

Strengths: AUROC provides a single, *quantitative* number to summarize performance, and the ROC curve itself creates a very *informative* metric to understand true and false positive rate trade-offs. ROC curve analysis is *automated* and *replicable* - it can be performed by any LBD system that applies a threshold. ROC curve analysis is *modular*, it can be used to evaluate the filtering or term ranking steps of LBD. Since ROC curve analysis and link-prediction type analysis are smaller, more constrained evaluation tasks, they relax the constraint that the gold standard dataset contain all possible future discoveries, and instead evaluate based solely on the presence or absence of samples in a dataset, making them more easily assessed [98].

Weaknesses: ROC curves are a powerful evaluation method, but may be less informative than precision and recall (PR) curves for problems with higher class imbalances such as LBD [99].

Conclusion: ROC curves are an excellent evaluation metric. We use them in Chapter 5 and 6, however standard and representative datasets should be developed, and PR curves may be more appropriate for LBD. We address these problems in Chapter 6.

3.3.2.6 Established Mutual Information Graphs

Description: Wren [63] evaluates target term ranking algorithms by the rate at which they accumulate the total established mutual information (MI) in a dataset. His key hypotheses are that MI between the start and target term can be used as an estimate of relevance or irrelevance, and that term pairs with high established MI based on their direct relationship should retain a high MI based on their indirect relationship. He uses these direct MI scores as gold standard weights, and evaluates the performance of indirect MI measures on their ability to rank directly related terms.

Example: Using a single start term, *capsaicin*, the MI between it and every directly co-occurring term is found. The sum total of all MI's is calculated, and each term is assigned a score that is their percent of total MI contributed. This estimates their percentage of relevance contributed by that term to the sum total of all relevance. Next, an indirect ranking algorithm is used to rank this set of directly co-occurring terms. Using the ranks assigned by the indirect ranking algorithm, and the scores of percent total MI, a cumulative sum at each rank is calculated and plotted. That is, for rank 1, the percent total MI accumulated is the percent total contributed by the first term. For rank 2, the percent total MI accumulated is the percent total contributed by terms one and two. For rank 3, the percent total MI accumulated is the percent total contributed by the first three terms, and so on. These accumulated totals are plotted in a graph of rank versus cumulative established MI to show the rate at which target term ranking algorithms capture all relevant terms in a dataset. The area under the curve (AUC) is calculated to quantify performance with a single number, and the AUC of 50 randomly selected words is averaged to better quantify performance.

Strengths: Established MI graphs can be generated using data alone, and are therefore *automated* and *replicable*. They are *quantitative*, and the AUC provides a single number for evaluation. They are *modular*, and very *informative*, and provide an understanding of how a system performs. Using direct relationships to estimate relevance, and using that in evaluation means that established MI graphs are sensitive to the ranks of the most relevant terms. This sensitivity to rank is critical for target term ranking evaluation. Other methods, such as mean average precision (MAP) are also sensitive to rank, but they assume a perfect gold standard. Estimating a gold standard is difficult, and in methods such as MAP, an untrue sample in the gold standard would contribute as much as a true sample. Established MI graphs do not require a perfect gold standard. Instead, relevance can be used both as an estimate of truthfulness and importance. Co-occurrences that do not constitute a relationship will presumably be less relevant, and therefore contribute less to the final established MI graph. In effect, creating a perfect gold standard requires knowing the cutoff threshold of relevance, and established MI graphs avoid the need for this threshold.

Weaknesses: This is an interesting evaluation metric, but it evaluates a method's performance on known relationships only. It makes no use of future relationships, but instead assumes that a method's performance on direct, known relationships can estimate its performance on indirect, future relationships.

Conclusions: Wren's observation that indirect ranking methods should perform well for directly related terms is interesting, and we believe indirect ranking measures are estimating relatedness, and should therefore perform well on standard direct relatedness evaluation datasets. Established MI graphs are interesting and informative, and we combine the idea with time-slicing and future direct relatedness to create cumulative relatedness graphs in Section 5.2.1.

3.3.2.7 Other and System Specific

Other evaluation methods have been used, but are often specific to a particular system or application. Cameron, et al. [10] perform a statistical evaluation on the context driven sub-graphs generated by their system. They determine if a discovery was replicated in a systematic way or serendipitous manner using sums of the rarity of the intermediate links. Yang, et al. [50] apply LBD to drug repurposing, for which they develop drug similarity metrics and relation extraction algorithms. They evaluate their drug similarity metric using Spearman's and Pearson's correlation coefficients with it and a gold standard drug-similarity based on the based on the anatomical therapeutic chemical (ATC) classification system. They evaluate the performance of their relation extraction algorithm using the Comparative Toxicogenomics Database as a gold standard, and evaluate their repurposed drug candidates with mean average precision in a time-slicing approach. Baker, et al. [11] use side effect profiles extracted from ChemoText to predict chemical biological behavior, specifically, 5-HT6 binding and dopamine antagonism. They evaluate their machine learning algorithms using sensitivity, specificity, and correct classification rate. Cohen, et al. [73] use a dataset of SemRep predications from which all "treats" relationships are removed. They use vector space operations to automatically rediscover these "treats" relationships as a chain of other relations. They use statistical analysis, including average precision to evaluate their technique.

3.3.3 Visualization

The development of effective user interfaces for LBD has become an increasingly important research topic. An effective user interface is critical for promoting the adoption of LBD systems in actual laboratory work. The goal of systems that focus

on user interaction is not to automatically produce new discoveries, but rather to provide a "dynamic and interactive experience that allows scientists to both explore and validate conceptual connections" [90].

Systems that focus on user interaction place the user as a central part of the discovery process. Discovery begins with a flash of insight, followed by an effort to realize and understand that insight [90]. User interaction focused systems are designed to promote abductive reasoning, and provide tools for deductive and inductive reasoning once a hypothesis has been generated. Their focus is on user interaction, and displaying information in a manner that facilitates greater understanding. User interaction systems are based on theories of how humans assemble new information and create new connections [76]. These systems are an aid to human creativity, rather than a fully automated hypothesis generation machine.

"Abductive reasoning, as defined by the philosopher and logician, C. S. Peirce (1839-1914) is concerned with the generation of new explanatory hypotheses given a set of observations" [65]. Inductive and deductive reasoning can then be applied to confirm or disprove these hypotheses. Although theories of abductive reasoning have been applied to other models of LBD [67], it is an important theory for user interaction systems. In these systems, abductive reasoning is accomplished through the theoretical framework of distributed cognition [90] in which a machine is viewed as complementary to the human mind. Users interact with the system to produce reasoning that is greater than the sum of its parts. The goal is not to automatically produce new discoveries, but rather to provide a "dynamic and interactive experience that allows scientists to both explore and validate conceptual connections" [90].

Theories of discovery browsing [100, 101, 71] may also guide the design of systems. Discovery browsing is based on Information Foraging Theory, and was first proposed for LBD by Wilkowski, et al. [76], and later implemented by Goodwin, et al. [100] and

Workman, et al. [71]. In "discovery browsing" information is displayed to the user, and the user selects topics they find interesting or surprising. The Spark system [71] uses SemRep predications with a highly interactive graphical user interface to spark the creativity of the user. Semantic MEDLINE[1] provides a visual environment for exploring SemMedDB predications and has been used for LBD [46, 47].

3.3.4 Functional Groups

Term ranking and Filtering methods focus on removing spurious terms from the target term list, but in addition to eliminating terms, similar terms may be grouped to reduce the total number of items that must be analyzed simultaneously. Weeber, et al. [59] outlined their idea of *functional pathways*, for which similar linking terms generated by their LBD system are manually grouped and analyzed. In their paper, they replicate the three primary pathways (originally identified by Swanson) for which fish oil treats Raynaud's Disease. These pathways are fish oil's effects on: blood viscosity, platelet aggregation, and vascular reactivity. Groups of terms related to these functional pathways are identified in the linking term set generated by their system. Weeber, et al. conclude that by analyzing the data as sets of related terms, the output is more understandable. Their method is manual and limited to the linking term set, but they do note that there are several fish oil-related concepts in their target term list, including: fish oils, maxepa, fatty acids omega-3, omega-3 polyunsaturated fatty acid, eicosapentaenoic acid, epa-e, cod liver oil, salmon oil, fatty acids essential, and dietary fats.

Automated methods of grouping target terms of an LBD system has been a largely overlooked topic, but some work has been done. Baker [53] assigns a high-level

[1] `https://skr3.nlm.nih.gov/SemMed/`

classification to sets of terms by exploiting the MeSH hierarchy. She provides a broad categorization of terms by assigning a category according to the MeSH term at the third level of the term's ancestor tree. Although effective for her application, using the MeSH hierarchy alone is problematic when using multiple taxonomies of the UMLS (e.g. MeSH and SNOMED CT), as their structures differ significantly. Cameron, et al. [10] use graph-based similarity measures, and hierarchical agglomerative clustering to group similar "contexts" of SemRep predications. The system creates an easily interpretable graphical output that succinctly explains the interaction between terms of interest. Their system output is impressive, but is computationally intensive, requires some manual intervention, and is intended for closed discovery, in which both the start and target terms are pre-defined.

CHAPTER 4

DIRECT RELATEDNESS

A core assumption is that a literature based discovery (LBD) hypothesis is an assertion that a relationship exists between two terms that never directly co-occur, and the likelihood of that hypothesis being true can be estimated by the strength of their relatedness. Semantic similarity and relatedness measures quantify the degree to which two concepts are similar (e.g., *liver-organ*) or related (e.g., *headache-aspirin*). Traditionally these measures are used to estimate direct relatedness, for which two terms directly co-occur. This makes the study and development of semantic relatedness measures critical to our overall goal of developing more effective LBD systems. Since estimating direct relatedness is a well studied field with standard evaluation metrics and datasets, and estimating indirect relatedness is a less well established field, we study direct relatedness as a starting point in developing indirect relatedness measures.

In this chapter, we describe our contributions to estimating direct semantic relatedness. Broadly, these include studies on association measures and vector representations. Results are shown on several standard evaluation datasets, and compared against several state of the art techniques. These results are interesting both for the further development of indirect relatedness measures, and for the field of direct relatedness and its applications.

This chapter is organized by our contributions. First, we present our contributions to direct association measures, which include concept association and expansion, set associations, and an analysis of parameters that affect performance. Next, we

present our contributions to vector-based relatedness measures which include a parameter analysis and comparison between different vector representations. Next, we compare against other measures for estimating direct relatedness, and achieve state of the art results for both association measures and vector-based relatedness measures. These results, show the effectiveness of association measures and vector-based measures at estimating semantic relatedness, and are critical to developing indirect relatedness measures. Lastly, we draw conclusions on how these results impact our work at estimating indirect relatedness.

4.1 Association Measures

In this section, we apply association measures to estimating direct semantic similarity and relatedness. In our method, we adapt several existing association measures to fit within our framework, which introduces several novel components. These modifications greatly improve results across evaluation datasets. Novel contributions of our method include:

1. Concept Associations - Use of UMLS concept co-occurrence counts rather than word or term co-occurrence counts to account for lexical variation at the synonymous level.

2. Concept Expansion - Introduction of concept expansion, a technique that propagates frequency counts up the UMLS hierarchy to account for lexical variation at the hyponymous level (e.g. a migraine is a headache).

3. Set Associations - Extension of association measures to quantify the association between sets of terms rather than single term-term pairs.

4. UMLS::Association[1] - An open-source Perl implementation of the methods discussed in this section.

Figure 12 shows a high level view of our method. It consists of three components: *Co-occurrence Collection*, *Contingency Table Generation*, and *Quantification*. First, concept co-occurrence counts are collected over MetaMapped text. This co-occurrence information is stored and used in later calculations. Synonym level lexical variations and multi-word term identification are accounted for since MetaMapped text is used in place of plain text. Second, a contingency table is generated from the collected co-occurrence counts. The contingency table optionally includes hierarchical information from the UMLS in the case of concept expansion, or incorporates co-occurrence values of sets of terms in the case of set associations. Third, the values in the contingency table are used to quantify the association between the concept pair. Each step is explained below.

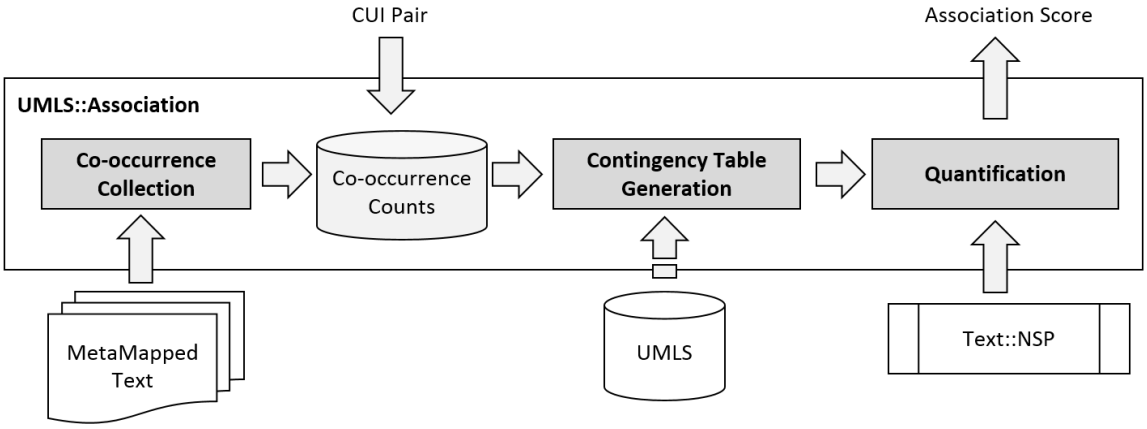

Fig. 12. The association measure calculation process. Our specific implementations are available in UMLS::Association, which uses MetaMapped Text as corpus, and Text::NSP for association equation calculations.

[1] `https://metacpan.org/pod/UMLS::Association/`

Co-occurrence Collection: We extract UMLS concept co-occurrence counts from MetaMapped text to build a database from which association measures are calculated. Although using concepts rather than terms provides stop word removal, normalization, and identification of multi-word concepts, it introduces a complication of ambiguity when the text is mapped to concepts. This ambiguity is caused by tokens for which multiple CUI mappings are valid. Consider the phrase, *Stop Smoking*. The term *Stop* has just one possible meaning, and therefore MetaMap assigns only the CUI `C1947925` to that term. *Smoking* on the other-hand has three possible interpretations, `C0037369` (*Smoking*), `C0453996` (*Tobacco Smoking*), and `C1881674` (*Smoke Emission*), so MetaMap assigns three CUIS to the token *Smoking*. We treat ambiguous mappings as an occurrence of each possible term pair. In our *Stop Smoking* example, concept co-occurrence counts are incremented for: `C1947925 C0037369`, `C1947925 C0453996`, and `C1947925 C1881674`.

Contingency Table Generation: The co-occurrence counts are used to generate a contingency table as described in Section 2.5. Optionally, the UMLS hierarchy is used for concept expansion (Section 4.1.1), or the contingency table is augmented for set associations (Section 4.1.2).

Quantification: We calculate the association between two concepts using the counts in the contingency table using the Text::NSP package (Section 2.2.3), which includes a wide range of association measures. The result is a single number to quantify the relatedness between the two concepts.

4.1.1 Concept Expansion

Here, we describe concept expansion which accounts for lexical variation at the hyponymous level. Concept expansion propagates the co-occurrence counts of a concept's descendants up the UMLS hierarchy, such that concept co-occurrence counts

include all co-occurrences of a concept pair's descendants. For example, the concept co-occurrence count of *Tobacco Consumption - Heart Diseases* could reasonably include counts of *Smoking - Heart Diseases*, or *Smoking - Coronary Heart Diseases*, since *Smoking* and *Coronary Heart Diseases* are more specific mentions of their parent term. Concept expansion captures this relationship by propagating co-occurrence counts up the UMLS hierarchy for all descendants of a concept. For a given term pair the co-occurrence counts of the concept pair and any of their children is calculated. This process is best shown through an example. Consider the simplified hierarchy in Figure 13 of the concepts *Heart Diseases* and *Tobacco Consumption* (In reality, there are more children of this pair, but it creates an ineffective example to show the entire descendant hierarchy).

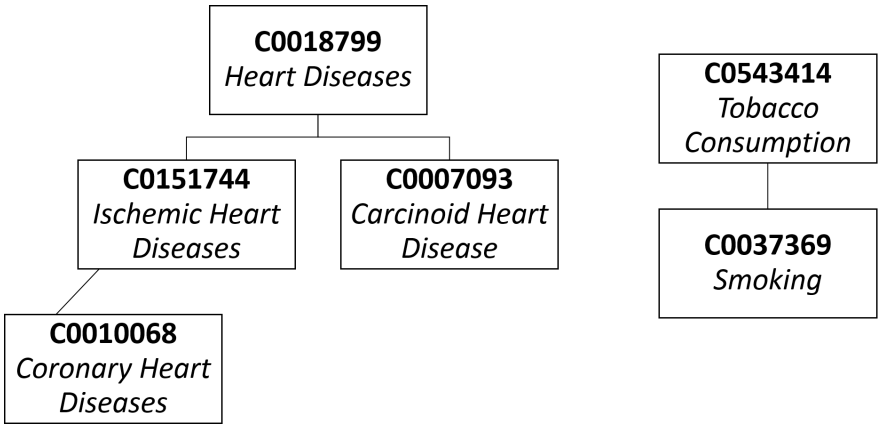

Fig. 13. Simplified descendant hierarchies of `C0018799` (*Heart Diseases*) and `C0543414` (*Tobacco Consumption*).

The contingency table for *Tobacco Consumption - Heart Diseases* is constructed such that co-occurrences of all concept pairs in the descendant hierarchy are recorded. Using our example, when retrieving the co-occurrence frequency, n_{11}, co-occurrences are counted by finding all occurrences of the term pairs: *Tobacco Consumption - Heart Diseases* OR *Smoking - Heart Diseases* OR *Tobacco Consumption - Ischemic Heart*

Diseases OR *Smoking - Ischemic Heart Diseases* OR *Tobacco Consumption - Carcinoid Heart Disease* OR *Smoking - Carcinoid Heart Disease* OR *Tobacco Consumption - Coronary Heart Diseases* OR *Smoking - Coronary Heart Diseases*. In this manner the semantic meaning of the concept captures hyponymous concept co-occurrences, producing more accurate association scores.

4.1.2 Set Association

Set associations are an extension of association measures in which we quantify the association between one set of concepts and another set of concepts rather than a single concept and another single concept. This is accomplished by modifying the contingency table values n_{11}, n_{1p}, n_{p1}, and n_{pp} (described in Section 2.5) prior to plugging them into an association measure equation. Specifically, we redefine: n_{11} as the sum of co-occurrences between set A and set C. n_{1p} as the sum of co-occurrences between set A and any term. n_{p1} as the sum of co-occurrences between set C and any term. n_{pp} remains unchanged, and is the total co-occurrence count.

Figure 14 shows an example of two sets of terms. Set A, indicated by white circles, and set C, indicated by black circles co-occur with one another, and other terms, indicated by gray squares. Co-occurrences are shown as edges, the number next to the edge indicates that term pair's co-occurrence count. In this example: n_{11} = 5+2+4=11, n_{1p} = 5+2+4+8=19, n_{p1} = 9+5+2+4+7=27, and n_{pp} = the total number of co-occurrences in the dataset. These numbers are then plugged into an association measure equation to output a scalar value for the association between sets A and C.

Concept expansion can be generalized as a special case of set associations, for which we define set A as concept A and its ancestors in the UMLS, and set C as concept C and its ancestors in the UMLS. Set associations have direct applicability

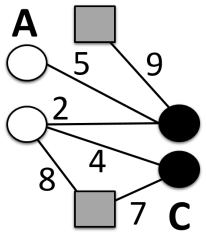

Fig. 14. A diagram showing set A and set C co-occurrences. Set A and C occur both with each other, and terms not in either set.

in indirect association measures (Chapter 5), and possible applications in other fields such as sentence or phrase similarity, or automatic keyword identification.

4.1.3 Experimental Details

In the next section we present a parameter analysis of association measures, for which we use UMLS::Association version 0.13. Co-occurrence counts were collected from the 2015 MetaMapped MEDLINE baseline. For concept expansion, we used SNOMED CT in the UMLS version 2016AA with parent/child (PAR/CHD) relations. We use the SNOMED CT taxonomy, because it is a taxonomy of clinical terms, and our evaluation data contains clinical terms. Analysis is performed as described in Section 2.4.1 by reporting Spearman's Rank Correlation (ρ) on the UMNSRS tagged for Similarity and Relatedness, and the MiniMayo dataset rated by coders and by physicians, and using Fisher's R-to-Z transformation to calculate the degree of significance between the correlation results. Statistical significances are used throughout, and here we define significant, or significance, to mean statistically significantly different correlations with two-tailed $p < 0.05$.

4.1.4 Association Measure Results

In our experiments, we attempt to answer several questions, the first of which is, how well do association measures perform for estimating semantic similarity and relatedness, and does using concept co-occurrences rather than term co-occurrences improve performance. This is followed by a comprehensive analysis of the effects of various hyper-parameters, such as corpus size, window size, enforcement of word order, and thresholding. Finally, we determine if our method of concept expansion improves results, and analyze the performance of individual association measures.

Do Concept Associations Outperform Term Associations? In this experiment, we generate baseline results of term co-occurrences rather than concept co-occurrences. We use the compoundify tool to identify multi-word terms in text, from which term co-occurrences are found using Text::NSP. We compare these term-term pair results to concept-concept pair results. For both techniques we use the same corpus and hyper-parameter settings, that is: The 2015 MEDLINE baseline titles and abstracts from the years 1975 to present. We use a window size of 8, do not consider word order in the calculation, and do not apply a minimum frequency threshold. Concept expansion is not applied. In this manner we can directly compare the performance of using term co-occurrence counts to concept co-occurrence counts. Results are shown in Table 7, where only the highest correlation among all association measures is shown for each dataset.

The results show that using concept co-occurrences rather than term co-occurrences significantly increases the performance for the UMNSRS Rel dataset. Additionally, using concepts rather than terms allows for more term pairs (indicated by n) to be calculated. **Conclusion:** Since performance increases significantly for UMNSRS Rel, and because n is always larger when using concept co-occurrences, we recommend us-

Correlation Coefficients (ρ) and number of samples (n)

	MiniMayo Cod	MiniMayo Phys	UMNSRS Sim	UMNSRS Rel
terms	0.6151 (20)	0.6985 (20)	0.6401 (122)	0.5054 (211)
concepts	**0.8092** (29)	**0.8405** (29)	**0.7061** (392)	**0.6559** (418)
p-value	0.1902	0.2501	0.2460	0.0069*

Table 7. Results and statistical analysis of using term associations versus concept associations. Bold terms indicate the best performing method for a dataset. Statistically significant p-values are marked by an asterisk

ing concept co-occurrence counts rather than term co-occurrence counts for semantic association measures.

What is the best subset of MEDLINE to use? Language use and the meaning of words change over time. This can be the result of new science, changes in word meanings, or new vocabulary. Outdated information or word usage affects co-occurrences patterns in text, and therefore may influence association measures. For instance, tobacco use was considered healthy until studies in the 1960's indicated otherwise. To determine the impact of these effects, we divide the MetaMapped MEDLINE baseline into three subsets: all of MEDLINE (1809+), which contains citations dating back to 1809; MEDLINE from the year 1975 onward (1975+), and MEDLINE from the year 2000 onward (2000+). We chose 1975+ because prior to 1975, only 2% of the citations contained an abstract, and we chose 2000+, because it seems adequately representative of recent citations. Table 8 shows the results of using these three date ranges on each evaluation dataset. Results using the best hyper-parameters are shown, that is, the best window size, with or without enforcement of word order, with or without concept expansion, and best association measure.

The results and statistical analysis show that there is no significant difference between any of the date ranges. A disadvantage of using 2000+ though, is that scores for all term pairs are not generated for both UMNSRS datasets. **Conclusion:** Although

Correlation Coefficients (ρ) and number of samples (n)

Date Range	MiniMayo Cod	MiniMayo Phys	UMNSRS Sim	UMNSRS Rel
1809+	0.7777 (29)	0.8093 (29)	**0.7115** (392)	0.6247 (418)
1975+	0.7811 (29)	0.8109 (29)	0.7109 (392)	**0.6267** (418)
2000+	**0.8231** (29)	**0.8132** (29)	0.6885 (390)	0.5870 (414)

p-values

Date	MiniMayo Cod		MiniMayo Phys		UMNSRS Sim		UMNSRS Rel	
Range	1809+	1975+	1809+	1975+	1809+	1975+	1809+	1975+
1975+	0.9761	-	0.9840	-	0.9840	-	0.9601	-
2000+	0.6455	0.6672	0.9681	0.9840	0.8287	0.5419	0.3898	0.3681

Table 8. Results and statistical analysis of different date range subsets of MEDLINE. Bold terms indicate the best performing method for a dataset. The p-values table assesses the significance of difference between date ranges for each dataset. No p-values are significant.

there is no statistical significance between the results generated using different MEDLINE subsets, we prefer using 1975+ because this is when recording abstracts became common, and association scores are calculated for all term pairs in the dataset. The p-values of using 1975+ compared to 1809+ are very high, >0.96, indicating that using 1975+ is a good estimate of 1809+. The lack of significant difference between results is a testament to the robustness of association measures to corpus size, but also likely a result of the exponential growth of publications. The number of publications in more recent years greatly outnumbers older publications, drowning out older word usage and outdated science.

What is the best window size, and does word order matter? When collecting co-occurrence counts, the window size, which is the proximity between words for which co-occurrences are counted, may be changed. Similarly word order, which is whether or not terms only after, or terms both before and after the term of interest are recorded may be enforced. Association measures traditionally use bigram counts, meaning the window size is 1, and word order is enforced. For example, if *Stop*

Smoking is seen in text, the co-occurrence count (n_{11}) between *Stop* and *Smoking* will be incremented, but not *Smoking* and *Stop*. This makes sense for many tasks, but it is not clear that word order matters, or that only direct adjacencies should be counted when estimating relatedness; particularly because MetaMap automatically detects and identifies multi-word terms. Let's expand the example to the text, "Smoking increases risk for Heart Disease and Lung Disease". It makes sense to increment co-occurrences for both *Smoking* and *Heart Disease*, *Smoking* and *Lung Disease*, *Heart Disease* and *Smoking*, and *Lung Disease* and *Smoking*.

Here, we compare the performance of collecting co-occurrence counts with a window size 1 and enforcing word order (window 1 ordered), a window size of 1 and ignoring word order (window 1 no order), a window size of 8 and enforcing word order (window 8 ordered), and a window size of 8 and ignoring word order (window 8 no order). When enforcing word order, we increment co-occurrence counts for all word pairs of the word of interest and each of the window size words after it. When ignoring word order, we increment co-occurrence counts for all word pairs of the word of interest and each of the window size words before and after it. In both cases, sentence boundaries are ignored. Results are shown in Table 9 using the 1975+ subset of the MetaMapped MEDLINE baseline as the corpus, and the best result among all association scores, and with or without concept expansion reported.

When comparing word order enforcement and window sizes, the results are never significantly different. This indicates that association measure are robust to changes in these parameters. The p-values are all quite high for window 8 ordered versus unordered, while the p-values are slightly lower for window 1 ordered versus unordered. This indicates that when using larger window sizes, order and no-order distributions more similar than those generated with a window size of 1. This is likely because co-occurrences that cross sentence boundaries are counted, and an increased count of

Correlation Coefficients (ρ) and number of samples (n)

parameters	MiniMayo Cod	MiniMayo Phys	UMNSRS Sim	UMNSRS Rel
window 1 ordered	0.8213 (29)	0.8337 (29)	0.6890 (392)	0.5924 (418)
window 1 no order	0.7811 (29)	0.8109 (29)	0.7109 (392)	0.6267 (418)
window 8 ordered	0.8219 (29)	0.8483 (29)	0.7208 (392)	**0.6559** (418)
window 8 no order	**0.8281** (29)	**0.8484** (29)	**0.7299** (392)	0.6512 (418)

p-values

comparison	MiniMayo Cod	MiniMayo Phys	UMNSRS Sim	UMNSRS Rel
best window 1 vs. best window 8	0.9362	0.8572	0.5823	0.4777
best ordered vs. best no order	0.9442	1.0000	0.7872	0.9045
window 1 order vs. window 1 no order	0.6818	0.8026	0.5485	0.4295
window 8 order vs. window 8 no order	0.9442	1.0000	0.7872	0.9045

Table 9. Results of different window sizes and how enforcement of word order affects performance. Bold terms indicate the best performing method for each dataset. The p-values assess the significance of difference for the comparison of that row on each evaluation dataset. No p-values are significant.

co-occurrences overall means that the same distribution is estimated. **Conclusion:** A window size of 8, and ignoring word order is preferred for theoretical reasons, although the improvements are not significant.

Does concept expansion increase performance? The process of concept expansion described in section 4.1.1 augments the co-occurrence counts of concepts by using the UMLS hierarchy. Here, we compare the results across window sizes, and with or without word order. Results both with (with) and without (w/o) concept expansion are listed in Table 10. Only the best performing association measure for the parameters listed is shown. The number of terms compared (n) does not increase with concept expansion, so just a single n is listed. The results show that concept expansion does not significantly improves performance for any dataset or parameter settings. **Conclusion:** Concept expansion requires additional computation cost, and does not significantly increase performance for any dataset or parameter settings. For these reasons it is not recommended, however concept expansion may increase the number

of terms that can be compared and provide more accurate co-occurrence statistics particularly when using smaller corpora, for which data sparsity is a problem.

Correlation Coefficients (ρ) and number of samples (n)

parameters	MiniMayo Cod			MiniMayo Phys		
	w/o	with	n	w/o	with	n
1975+ window 1 ordered	0.8213	0.8010	(29)	0.8337	0.8320	(29)
1975+ window 1 no order	0.7731	0.7811	(29)	0.8043	0.8109	(29)
1975+ window 8 ordered	0.8092	0.8219	(29)	0.8405	0.8483	(29)
1975+ window 8 no order	0.8162	**0.8281**	(29)	0.8419	**0.8484**	(29)
parameters	UMNSRS Sim			UMNSRS Rel		
	w/o	with	n	w/o	with	n
1975+ window 1 ordered	0.6705	0.6890	(392)	0.5900	0.5924	(418)
1975+ window 1 no order	0.6983	0.7109	(392)	0.6267	0.3201	(418)
1975+ window 8 ordered	0.7061	0.7208	(392)	**0.6559**	0.6372	(418)
1975+ window 8 no order	0.7129	**0.7299**	(392)	0.6512	0.6374	(418)

p-values

with vs. w/o concept expansion	MiniMayo Cod	MiniMayo Phys	UMNSRS Sim	UMNSRS Rel
1975+ window 1 ordered	0.8337	0.9840	0.6312	0.9601
1975+ window 1 no order	0.9442	0.9442	0.7263	0.8729
1975+ window 8 ordered	0.8870	0.9203	0.6745	0.6455
1975+ window 8 no order	0.8966	0.9362	0.6241	0.7339
1975+ window 1 best	0.8650	0.9840	0.5353	0.8729
1975+ window 8 best	0.9203	0.9442	0.5961	0.9840

Table 10. Results and statistical analysis of whether or not concept expansion increases performance. The p-values table assesses the significance of difference between using concept expansion and not using concept expansion. Each row indicates the parameters used for the comparison, for instance the *1975+ window 1 ordered* row shows the p-values for the results with versus without concept expansion using the 1975+ MEDLINE subset, a window size of 1, and enforcement of word order. The *1975+ window 1 best* and *1975+ window 8 best* rows compare the best performance with or without word order for that window size. Bold values indicate the best performing set of parameters for each dataset. No p-values are significant.

Does minimum count thresholding improve results? Co-occurrence counts can be noisy as a result of the wording of a sentence or paragraph, choosing a window size that is too large, or ambiguity when mapping text to a concepts. A minimum co-occurrence count threshold may be applied to reduce co-occurrence noise. Applying

this threshold eliminates any co-occurrence counts from the co-occurrence database that are less than or equal to the minimum threshold (e.g. a threshold of 1 removes any term pairs that co-occur just once). Here, we compare performance using no minimum co-occurrence threshold, and thresholds of 1, 3, 5, and 10. Results are shown in Table 11.

Minimum Threshold (thresh) Results

1975+ window 8					1975+ window 1				
thresh	dataset	score	n	p-value	thresh	dataset	score	n	p-value
none	cod	0.8281	29	-	none	cod	0.8213ˆ	29	-
none	phy	0.8484	29	-	none	phy	0.8337	29	-
none	sim	0.7299ˆ	392	-	none	sim	0.7109ˆ	392	-
none	rel	0.6559ˆ	418	-	none	rel	0.6267ˆ	418	-
1	cod	0.8301	29	0.9840	1	cod	**0.8178**	29	0.9681
1	phy	0.8460	29	0.9761	1	phy	0.8310	29	0.9761
1	sim	**0.7213**	390	0.8026	1	sim	**0.6991**	390	0.7414
1	rel	**0.6404**	416	0.7039	1	rel	**0.6109**	414	0.7114
3	cod	0.8339	29	0.9442	3	cod	0.8178	29	0.6981
3	phy	0.8462	29	0.9762	3	phy	**0.8366**	29	0.9761
3	sim	0.7184	387	0.7339	3	sim	0.6627	386	0.2077
3	rel	0.6305	412	0.5333	3	rel	0.5471	402	0.0836
5	cod	0.8373	29	0.9124	5	cod	0.8178	29	0.9681
5	phy	**0.8537**	29	0.9442	5	phy	0.8328	29	0.9920
5	rel	0.7080	387	0.5287	5	rel	0.6373	384	0.0601
5	sim	0.6226	407	0.4178	5	sim	0.5147	396	0.0178*
10	cod	**0.8410**	29	0.8808	10	cod	0.7607	29	0.5619
10	phy	**0.8537**	29	0.9442	10	phy	0.7622	29	0.4777
10	rel	0.6970	383	0.3524	10	rel	0.6205	368	0.2051*
10	sim	0.5985	398	0.1770	10	sim	0.4946	383	0.0063*

Table 11. Results of various minimum thresholds. The left set of tables shows results using a window size of 8, and the right shows results using a window size of 1. The highest correlation using a threshold for each dataset and window size is bolded. A ˆ indicates that using no threshold performed the best for the dataset and window size. The p-values shows the degree of significance between the threshold and using no threshold. Statistically significant p-values are marked with an asterisk (*).

The results show that using a threshold has little impact on performance using

either a window size of 8 or 1, although there are significantly different results with thresholds of 5 and 10 and a window size of 1. The number of terms compared (n) decreases as the threshold size increases for both a window size of 1 and 8. **Conclusion:** The results show applying a minimum co-occurrence threshold of 3 or less does not significantly change the performance of association measures. This shows that association measures are robust to the noise that thresholds attempt to remove. Applying a threshold can greatly reduce the number of elements of a co-occurrence database, making thresholding a method to improve computation time while not significantly decreasing performance.

What is the best association measure? For all the results shown so far we have reported the best result among all association measures. Here, we determine which association measure is best. To do this, we compare both the best results across all parameters (any), and the results using a recommended set of parameters (rec.). The recommended parameters are chosen from the results of experiments in previous subsections, and are: MetaMapped MEDLINE citations from 1975 onwards (1975+), a co-occurrence window size of 8 (window 8), ignoring word order (no order), not applying a minimum co-occurrence threshold, and not using concept expansion. Table 12 shows these results, and p-values to assess the statistical significances using five association measures: Log Likelihood Ratio (ll), Left Tailed Fisher (lf), Pearson's Chi Squared (χ^2), Dice Coefficient ($dice$), and Odds Ratio ($odds$).

Analysis of the results show that when using the recommended parameters (*best of rec. vs. worst of rec.*) there is no significant difference between any of the association measures on three of four evaluation datasets. There is a significant difference for UMNSRS Sim, for which Odds Ratio (odds) significantly outperforms Dice Coefficient (dice), the worst performing association measure for that dataset. It does not significantly outperform any other measure though ($p = 0.2585$ for Odds against Log

Association Measure Results using Recommended Parameters

MiniMayo Coders			MiniMayo Phys			UMNSRS Sim			UMNSRS Rel		
ll	0.7945	29	ll	0.8138	29	ll	0.6708	392	ll	0.6378	418
textbf χ^2	**0.8162**	29	χ^2	**0.8419**	29	χ^2	0.6934	392	χ^2	0.6406	418
dice	0.7207	29	dice	0.7103	29	dice	0.5298	392	dice	0.6108	418
odds	0.8027	29	odds	0.8188	29	odds	**0.7129**	392	odds	0.6266	418
lf	0.7582	29	lf	0.7199	29	lf	0.6782	392	lf	**0.6512**	418
freq	0.7431	29	freq	0.7603	29	freq	0.6604	392	freq	0.6267	418

Association Measure Results using any Parameters

MiniMayo Coders			MiniMayo Phys			UMNSRS Sim			UMNSRS Rel		
ll	0.8234	29	ll	0.8382	29	ll	0.6764	392	ll	0.6439	29
χ^2	0.8288	29	χ^2	**0.8537**	29	χ^2	0.7044	392	χ^2	0.6427	29
dice	0.7120	29	dice	0.7002	29	dice	0.5428	392	dice	0.6200	29
odds	**0.8410**	29	odds	0.8241	29	odds	**0.7299**	392	odds	0.6251	29
lf	0.8261	29	lf	0.7176	29	lf	0.6880	392	lf	**0.6559**	29

p-values summary

	MM Cod	MM Phys	UMNSRS Sim	UMNSRS Rel
best of rec. vs. worst of rec.	0.2301	0.1471	0.0000*	0.3843
best of any vs. worst of any	0.3953	0.2187	0.0000*	0.3320
best any vs. best rec.	0.7718	0.8808	0.6241	0.9045
best rec vs. freq	0.4965	0.4065	0.1645	0.5485

p-values for UMNSRS Sim using recommended parameters

	ll	χ^2	dice	odds	lf
χ^2	0.5552	-	-	-	-
dice	0.0019*	0.0002*	-	-	-
odds	0.2585	0.5892	0.0000*	-	-
lf	0.8493	0.6892	0.0010*	0.3472	-
freq	0.1949	0.3953	0.0045*	0.1645	0.6527

p-values for UMNSRS Sim using any parameters

	ll	χ^2	dice	odds
χ^2	0.4533	-	-	-
dice	0.0028*	0.0001*	-	-
odds	0.1389	0.4654	0.0000*	-
lf	0.7642	0.6599	0.0010*	0.2380

Table 12. Results using the recommended parameters, and the best of any parameters for each association measure (top two tables). Each sub-table shows the results for a single dataset. The columns within the subtables show the association measure, the Spearman's Rank Correlation, and the number of samples. The best results for each dataset are bolded. The p-values summary table shows the p-values of the comparison described in that row. Since UMNSRS Sim is the only dataset with significant differences, the bottom two tables show p-values between all association measures for that dataset.The p-value tables for UMNSRS Sim indicate p-values for each association measure versus other association measures for the UMNSRS Sim dataset using recommended (second from bottom table) or any parameters (bottom table). Statistically significant p-values are marked with an asterisk.

Likelihood Ratio (ll), the second worst).

We also compare results across the best performance using any parameter settings (date range, window size, word order, threshold, with or without concept expansion). These results are shown in the second from top table of Table 12, and p-values are shown in the *best of any vs. worst of any* row of the p-values table. Again, there is no significant difference between any of the association measures on three of four evaluation datasets. There is a significant difference for UMNSRS Sim, for which Odds significantly outperforms Dice, the worst performing association measure for that dataset. It does not significantly outperform any other measure though ($p = 0.1389$ for Odds against ll, the second worst). The results for any parameters and the results for recommended parameters indicate that choosing an association measure is somewhat arbitrary. Dice is not recommended, since Odds Ratio outperforms it using any parameters and recommended parameters on the UMNSRS Sim dataset.

To further compare different association measures we also compare each association measure to each other using recommended and any parameters on the UMNSRS Sim dataset. This was selected, since it is the only dataset for which any association measure is significantly different. The p-values in these tables show that every association measure, significantly outperforms dice using recommended and any parameters, however none of the association measures significantly outperform any of other association measures.

The similarity between results using any parameters and recommended parameters indicates that association measures are robust to parameter selection, so we compare the best results using any parameters, to the best results using the recommended parameters (*best any vs. best rec.*). We find that for all datasets there is no significant difference between performance, and confirm the robustness of association measures to the parameter selection.

Lastly, we compare the performance of association measures using recommended parameters to using frequency of co-occurrence (*best rec. vs. freq*), which scores term-term pairs as the number of times they co-occur in a corpus. This serves as our baseline. For all datasets association measures do not outperform the frequency baseline. This is surprising, but hints at the power of co-occurrence information in estimating semantic similarity and relatedness. **Conclusion:** Selecting an association measure is somewhat arbitrary, but we do not recommend Dice, since every other association measure significantly outperforms it. Left Fisher is computationally the most expensive, so is not recommended for large datasets. Odds does not take into account n_{pp}, in its association calculation, which differentiates it from other association measures, and may affect its selection for theoretical reasons.

Summary

In this section, we presented an in depth analysis of parameters that affect the performance of association measures at estimating direct semantic relatedness. We found that using UMLS concepts instead of terms increases performance, association measures are robust to corpus size and our selection of a MEDLINE subset, association measures are robust to window size and word order, concept expansion does not significantly increase performance, applying a minimum co-occurrence threshold of 3 or less does not affect performance, and that choosing an association measure equation is somewhat arbitrary. This robustness to parameter choice is encouraging, and indicates that association measures are applicable for a wide range of training corpora and applications, including the development of indirect association measures. Based on these findings, we select some recommended parameters, and use them to evaluate direct and indirect association moving forward. These parameters are to use concept co-occurrence counts collected from titles and abstracts of MEDLINE from

dates ranging from 1975 onward with a window size of 8 and word order ignored, do not use concept expansion, apply a minimum co-occurrence threshold of 1, and use the Pearson's chi squared (χ^2) association measure.

4.2 Vector Representations

In this section, we perform an analysis of different vector representations and parameters affecting their performance at estimating direct semantic relatedness. Since vector cosine distance can intrinsically calculate indirect relatedness - no direct co-occurrence is needed to calculate the cosine of two vectors - they are directly applicable to calculating indirect relatedness. Additionally, distributional context vectors have been shown to perform well on direct semantic similarity and relatedness tasks [102, 103, 104], and are therefore exciting for our research.

In previous work, vectors are built for individual words, and the best method to represent multi-word terms (e.g. New York City, or Heart Attack) is unclear. To answer this question, we explore several aggregation methods for vector representations of multi-word terms. An answer to this question would be incomplete without addressing the challenges of sparseness and noise of distributional context vectors. We define sparseness as the vectors contain mostly zero values, and noise as information that is overly general and does not contribute to the word's representation. Dimensionality reduction techniques may be applied to transform the data from a higher dimensional space to a lower dimensional space, in which sparseness and noise are reduced. Most previous methods exploring feature extraction for quantifying the relatedness between biomedical and clinical term pairs have ignored multi-word terms [105, 106, 107]. We analyze several dimensionality reduction techniques for each multi-word term aggregation method, and vary the vectors dimensionality parameter for these techniques. Specifically, the contributions of this section are an analysis of:

- Multi-Word Term Aggregation Methods: We compare the performance of summation of component word vectors, averaging of component word vectors, direct creation of multi-word term vectors using the compoundify tool, and direct creation of concept vectors using the MetaMap tool.

- Dimensionality Reduction Techniques: singular value decomposition (SVD), word embeddings using skip-gram, and word embeddings using continuous bag of words (CBOW) are evaluated as dimensionality reduction techniques. Direct co-occurrence vectors (direct) of word-to-word, term-to-term, or concept-to-concept co-occurrences are used as a baseline.

- Vector Dimensionality: the dimensionality of the generated vectors is a parameter that effects performance. We evaluate each multi-word term aggregation method's and dimensionality reduction technique's performance at dimensionalities of 100, 200, 500, 1000, and 1500. For SVD we also evaluate at dimensionalities of 2000, 2500, and 3000.

4.2.1 Methods and Experimental Details

Analysis is performed as described in Section 2.4.1 by reporting Spearman's Rank Correlation (ρ) on the UMNSRS tagged for Similarity and Relatedness, and the MiniMayo dataset rated by coders and by physicians, and using Fisher's R-to-Z transformation to calculate the degree of significance between the correlation results.

Figure 15 shows the overall method used to generate relatedness scores. The major steps in this process are: 1) preprocess the text, 2) create vector representations, 3) aggregate terms, and 4) calculate relatedness. The arrows show how vector representations flow through the process, and the boxes show different steps in the process. In this section, we discuss each major step.

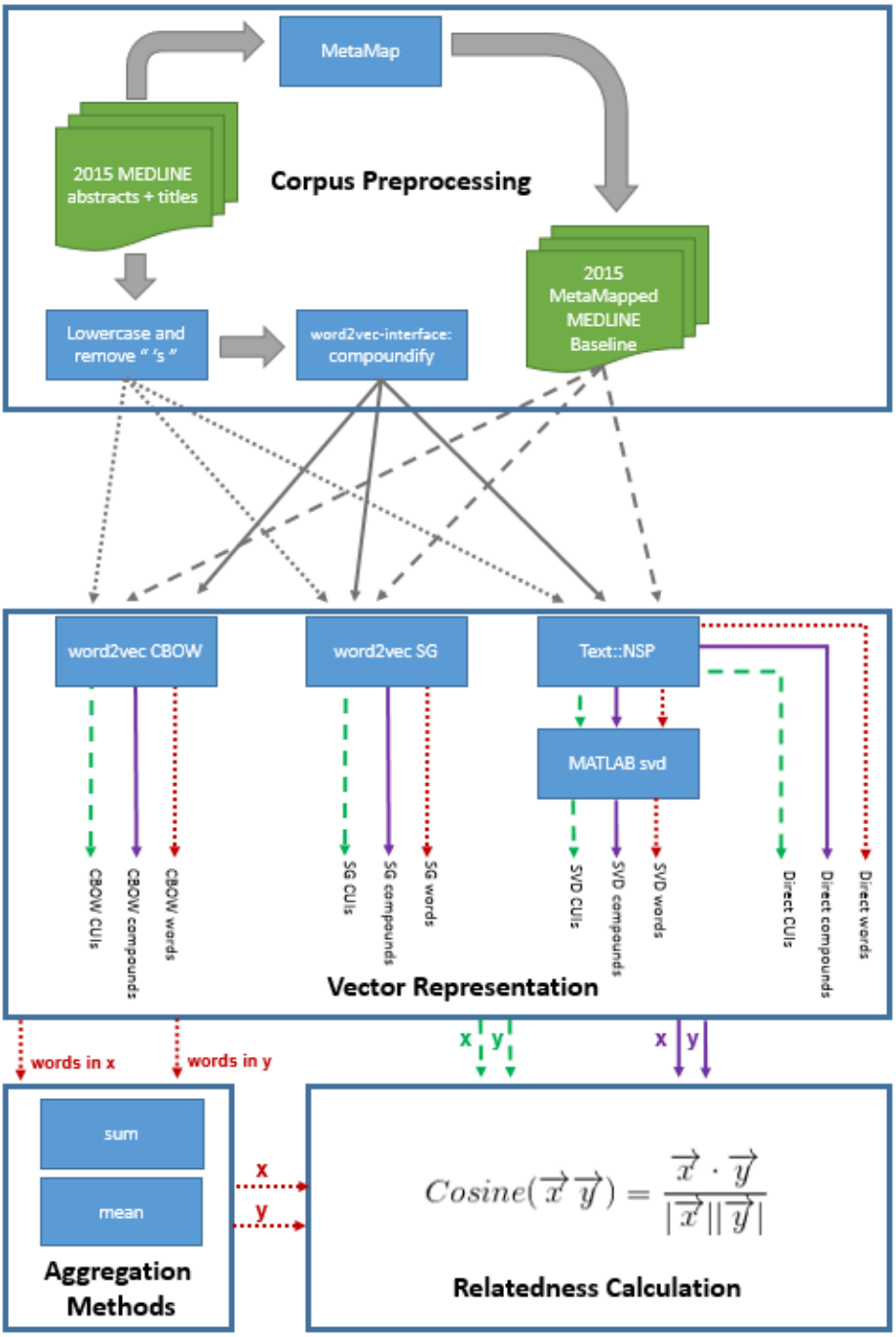

Fig. 15. Procedure for generating vector representations using different dimensionality reduction techniques and term aggregation methods.

Corpus Preprocessing: We use the titles and abstracts from the 2015 MEDLINE baseline as the training corpus for all techniques, however different multi-word term aggregation methods, and different dimensionality reduction techniques require different preprocessing steps. Figure 15 shows how each vector representation was obtained. First, the raw 2015 MEDLINE corpus, and the 2015 MetaMapped MEDLINE corpus are downloaded for the years 1975 onward. The 2015 MetaMapped MEDLINE baseline is used directly, without further preprocessing to generate CUI term aggregation vectors, but the raw (not MetaMapped) 2015 MEDLINE corpus must be preprocessed for the other methods. The raw corpus is converted to all lowercase, and all apostrophe s's ('s) are removed. This normalized corpus is used as input for each dimensionality reduction technique to generate the word vectors used in the sum and mean aggregation methods. To generate compound vectors, this normalized corpus is compoundified using the compoundify tool.

Vector Representation and Dimensionality Reduction: When applying a dimensionality reduction technique, the dimensionality of the resulting vector is a parameter. A dimensionality that is too small will not be able to effectively differentiate between words in vector space. A dimensionality that is too large does not successfully solve the problems of sparsity and noise. We aim to find the optimal vector dimensionality, that is, the most compact vector representation with the best performance.

We used the following packages and settings to obtain our vector representations and perform dimensionality reduction:

1. Direct Representation: We used the Text::NSP package to create a direct co-occurrence matrix, the rows of which are direct co-occurrence vector representations for terms. To construct these **direct** representations, we used

a windows size of 8, a frequency cutoff of 5, and removed stopwords (stopword removal is a package option, for a complete list of stop words see the Text::NSP package).

2. Singular Value Decomposition. We ran the MATLAB R2016b implementation of sparse matrix **SVD** (svds) on the direct representation matrix, and used each row of the resulting U matrix as a reduced vector.

3. Word Embeddings: We used the *word2vec* package developed by Mikolov, et al. [23]. for the continuous bag of words (**CBOW**) and skip-gram word (**SG**) embedding models with a window size of 8, a frequency cutoff of 5, and default settings for all other parameters.

Aggregation Methods: To compute the relatedness of multi-word terms using distributional context vectors, an aggregation method must be used. Typically, distributional context vectors are created for individual words in a corpus. For example, the words "heart" and "attack" are represented by their vector representations, $\overrightarrow{heart}$ and $\overrightarrow{attack}$. These component word vectors of a multi-word term must be combined to create a single vector for that term ($\overrightarrow{heart\_attack}$). We compare two combine operations:

1. **sum** - the multi-word term vector is the summation of the component word vectors - $\overrightarrow{heart} + \overrightarrow{attack} = \overrightarrow{heart\_attack}$

2. **mean** - the multi-word term vector is the average of the component word vectors - $(\overrightarrow{heart} + \overrightarrow{attack})/2 = \overrightarrow{heart\_attack}$

Rather than combine word vectors after construction, multi-word term vectors may be constructed directly from a preprocessed training corpus in which multi-word

terms have been identified. For example, in the training corpus, any occurrence of "heart" followed by "attack" will be tagged as the multi-word term "heart_attack" rather than either of the component word vectors. A heart_attack vector may then be directly from this corpus. We compare two preprocessing methods for direct construction of multi-word term vectors:

3. **compounds** - text is preprocessed using the compoundify tool. Multi-word term vectors are directly constructed from this compoundified text.

4. **CUIs** - the MetaMapped MEDLINE baseline is used, which contains text that has been preprocessed using the MetaMap tool. MetaMap maps raw text to UMLS concepts (CUIs) resulting in an ordered list of CUIs, from which multi-word concept vectors are directly constructed. A complication of MetaMap is that multi-word terms may be ambiguous. In these cases, MetaMap will generate all possible concepts that map to that term. When this occurs, as is the case with association measures (Section 4.1), we replace the term with each possible concept mapping.

4.2.2 Vector Representations Results

Table 13 shows all of the results for each term aggregation method, dimensionality reduction method, and vector dimensionality tested. Values in each cell show the correlation, slash, n, the number of samples compared. A hyphen ('-') indicates a score could not be calculated using those parameters. CBOW at a dimensionality of 1500 could not be calculated due to errors in the word2vec package. The first column ("100/e") shows results for a dimensionality of 100, and results with direct vector representation (for which dimensionality is the vocabulary size and does not vary). A discussion and analysis of these results is presented in the next few subsections.

		MiniMayo Phys					MiniMayo Cod				
		Dimensionality					Dimensionality				
Aggreg.	**Red.**	100/e	200	500	1000	1500	100/e	200	500	1000	1500
sum	SG	0.78/29	0.79/29	0.74/29	0.76/29	0.74/29	0.79/29	0.80/29	0.78/29	0.79/29	0.78/29
	CBOW	0.81/29	**0.82**/29	0.79/29	0.75/29	-	**0.82**/29	**0.82**/29	0.79/29	0.78/29	-
	SVD	0.38/28	0.57/28	0.56/28	0.79/28	0.66/28	0.36/28	0.53/28	0.52/28	0.54/28	0.71/28
	direct	0.37/28	-	-	-	-	0.34/28	-	-	-	-
mean	SG	0.78/29	0.79/29	0.74/29	0.76/29	0.74/29	0.79/29	0.80/29	0.78/29	0.79/29	0.78/28
	CBOW	0.81/29	**0.82**/29	0.79/29	0.75/29	-	**0.82**/29	0.81/29	0.79/29	0.78/29	-
	SVD	0.37/29	0.52/29	0.54/29	0.77/29	0.65/29	0.36/29	0.53/29	0.53/29	0.54/29	0.71/29
	direct	0.34/28	-	-	-	-	0.36/29	-	-	-	-
compound	SG	0.78/28	0.78/28	0.77/28	0.76/28	0.75/28	0.75/28	0.76/28	0.76/28	0.75/28	0.76/28
	CBOW	0.79/28	**0.80**/28	0.79/28	0.77/28	-	0.76/28	**0.78**/28	**0.78**/28	**0.78**/28	-
	SVD	0.65/28	0.74/28	0.75/28	0.72/28	0.70/28	0.65/28	0.73/28	0.70/28	0.72/28	0.72/28
	direct	0.49/28	-	-	-	-	0.51/28	-	-	-	-
cui	SG	0.76/29	0.76/29	0.77/29	0.76/29	0.76/29	0.77/29	0.77/29	0.78/29	0.77/29	0.79/29
	CBOW	0.77/29	0.75/29	**0.78**/29	0.76/29	-	**0.83**/29	**0.83**/29	**0.83**/29	0.82/29	-
	SVD	0.41/28	0.42/28	0.50/28	0.40/28	0.38/28	0.35/28	0.39/28	0.58/28	0.48/28	0.35/28
	direct	0.37/28	-	-	-	-	0.26/28	-	-	-	-

		UMNSRS Rel					UMNSRS Sim				
		100/e	200	500	1000	1500	100/e	200	500	1000	1500
sum	SG	**0.70**/374	**0.70**/374	0.68/374	0.69/374	0.68/374	0.59/396	0.61/396	**0.62**/396	**0.62**/396	**0.62**/396
	CBOW	0.68/374	0.69/374	0.66/374	0.61/374	-	0.55/396	0.61/396	0.61/396	0.58/396	-
	SVD	0.53/331	0.52/331	0.55/331	0.56/331	0.52/331	0.41/343	0.36/343	0.45/343	0.47/343	0.45/343
	direct	0.46/331	-	-	-	-	0.42/343	-	-	-	-
mean	SG	**0.70**/374	**0.70**/374	0.68/374	0.69/374	0.68/374	0.58/397	0.60/397	**0.61**/397	**0.61**/397	**0.61**/397
	CBOW	0.68/374	0.69/374	0.66/374	0.61/374	-	0.55/397	0.59/397	0.59/397	0.57/397	-
	SVD	0.53/332	0.52/332	0.55/332	0.55/32	0.52/332	0.39/346	0.34/346	0.46/346	0.47/346	0.43/346
	direct	0.33/400	-	-	-	-	0.36/430	-	-	-	-
compound	SG	**0.72**/373	0.71/373	0.70/373	0.69/373	0.70/373	0.63/393	0.64/393	0.64/393	0.65/393	**0.66**/393
	CBOW	0.70/373	0.70/373	0.68/373	0.65/373	-	0.62/393	0.64/393	0.65/393	0.65/393	-
	SVD	0.49/328	0.51/328	0.58/328	0.60/328	0.58/328	0.39/335	0.38/335	0.48/335	0.54/335	0.52/335
	direct	0.45/328	-	-	-	-	0.45/335	-	-	-	-
cui	SG	**0.74**/388	**0.74**/388	**0.74**/388	**0.74**/388	**0.74**/388	0.62/413	0.62/413	0.63/413	**0.64**/413	**0.64**/413
	CBOW	0.72/388	0.73/388	0.73/388	0.72/388	-	0.56/413	0.56/413	0.59/413	0.60/413	-
	SVD	0.41/362	0.45/362	0.50/362	0.53/362	0.57/362	0.26/380	0.31/380	0.30/380	0.34/380	0.38/380
	direct	0.35/362	-	-	-	-	0.20/380	-	-	-	-

Table 13. Results of each term aggregation method, dimensionality reduction technique, and vector dimensionality on all datasets. Values in each cell show the correlation, slash, n, the number of samples compared. A hyphen ('-') indicates a score could not be calculated using those parameters. The first column ("100/e") shows results for a vector dimensionality of 100, and results with direct vector representation. Bolded scores indicate the highest performing combination of parameters in each box.

Comparison between Dimensionality Reduction Techniques: Here, we compare the different dimensionality reduction techniques of SVD, SG, and CBOW against each other, and against the baseline of direct. Figure 16 shows the highest correlation of each of these techniques, for each dataset. Here, the highest correlation indicates the highest correlation among any results generated for that technique regardless of dimensionality or aggregation method. Table 14 shows statistical significances among the scores shown in Figure 16.

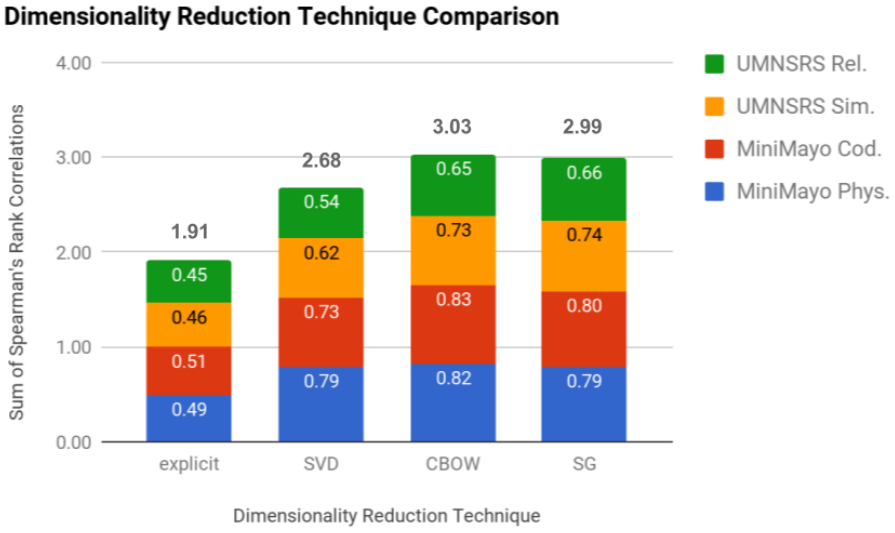

Fig. 16. Best results for each dimensionality reduction technique. The correlation for each dataset is shown within its rectangle, and the sum of correlations is shown above each column. A sum of 4.0 would indicate a perfect correlation of 1.0 for every dataset.

CBOW generates the highest overall accuracy, with a sum of correlations at 3.03. Although CBOW's overall accuracy is slightly higher than SG, SG outperforms CBOW on some datasets, and CBOW and SG do not perform significantly different on any single dataset. CBOW is the only method to perform significantly better than direct on MiniMayo Phys, or MiniMayo Cod The word embeddings methods

MiniMayo Phys

	direct	SVD	SG
SVD	0.0588	-	-
SG	0.0561	1.0	-
CBOW	0.0264*	0.7642	0.7566

MiniMayo Cod

	direct	SVD	SG
SVD	0.1971	-	-
SG	0.0561	0.5419	-
CBOW	0.0257*	0.3524	0.749

UMNSRS Sim

	direct	SVD	SG
SVD	0.0029*	-	-
SG	0.0*	0.0021*	-
CBOW	0.0*	0.0054*	0.7642

UMNSRS Rel

	direct	SVD	SG
SVD	0.1236	-	-
SG	0.0*	0.0114*	-
CBOW	0.0001*	0.022*	0.8103

Table 14. The two tailed p-values using Fishers R-to-Z transform, comparing the results of each dimensionality reduction technique. Each table corresponds to a different dataset, each row and column a different dimensionality reduction technique. p-values less than 0.05 are marked with an asterisk (*).

SG and CBOW, generate statistically significant higher correlations than SVD and direct on UMNSRS Sim and UMNSRS Rel SVD only generates a significantly higher correlation than direct for a single dataset (UMNSRS Sim). **Conclusion:** all the dimensionality reduction techniques improve correlation accuracy, but the word embeddings approaches (SG and CBOW) perform significantly better than SVD and direct. SG and CBOW perform on par with each other, with no significant differences. CBOW does, however generate the vector representations much more quickly than SG (our rough estimates indicate that SG takes between 5 and 9 times as long to train), and may be preferred due to this decreased computation time.

Comparison between Vector Dimensionality: Here, we tested the effects of vector dimensionality on performance for each dimensionality reduction technique.

Beginning with a vector dimensionality of 100, we increased the dimensionality of each vector representation to values of 200, 500, 1000, and 1500. Due to errors generated by the word2vec package, we were unable to generate vectors with CBOW at a dimensionality of 1500, although to more comprehensively test SVD, we continued to increase the dimensionality of SVD vectors to 2000, 2500, and 3000. Figure 17 shows the correlations of each technique on each dataset as the dimensionality increases. Each sub-figure shows performance on a different dataset. We show the highest correlation for each dimensionality reduction method and vector dimensionality, regardless of aggregation method. These results show that SG's and CBOW's correlations remain nearly constant as dimensionality increases, indicating that vector dimensionality has little impact on their performance, and a dimensionality of 200 is sufficient for these methods. For SVD, we do see an increase in performance as dimensionality increases. The performance continues to increase to 1500, and to determine whether a dimensionality greater than 1500 would continue to increase performance, we created additional SVD vectors with dimensionalities of 2000, 2500, and 3000. Figure 18 shows the overall performance of SVD at each vector dimensionality. The values above each column indicates the sum of the dataset correlations. We see an increase in correlation up to a dimensionality 1000, at which point correlation scores remain constant at 1500, and decrease at 2000, 2500, and 3000, indicating a dimensionality of 1000 is sufficient for SVD. **Conclusion:** a dimensionality of 200 is sufficient for the word embeddings methods, SG and CBOW. A larger dimensionality of 1000 is sufficient for SVD.

Comparison between Term Aggregation Methods: Here, we compare the multi-word term aggregation methods, sum, mean, compounds, and CUIs. Figure 19 shows the highest correlations of different aggregation methods across all dimensionality reduction techniques, and dimensionalities for each dataset. From this, we see

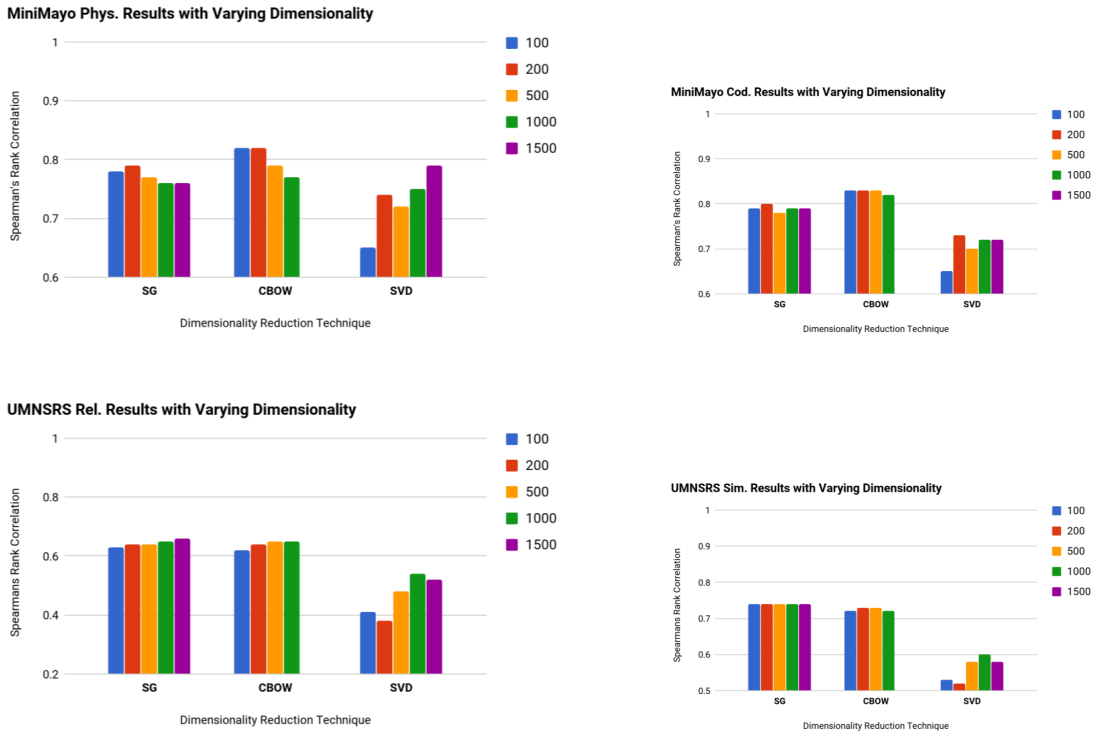

Fig. 17. Results of varying vector dimensionality on each datasets. The subgraphs correspond to a single dataset, and the column groupings a dimensionality reduction technique. Different colored columns indicate a different vector dimensionality.

that CUIs achieves the highest sum of correlations, but only by a small amount, and is not the best performing method on all datasets. Each method performs the best (or ties) on different datasets: sum and mean tied for highest on the MiniMayo Phys dataset, compounds on the UMNSRS Rel dataset, and CUIs on the UMNSRS Sim and MiniMayo Cod datasets. Table 15 shows the p-values for the correlations shown in Figure 19. No aggregation method performs significantly better on any single dataset, and the sum of correlations for CUIs is only marginally higher than other methods. **Conclusion:** there is no significantly best aggregation method, when comparing over all parameters.

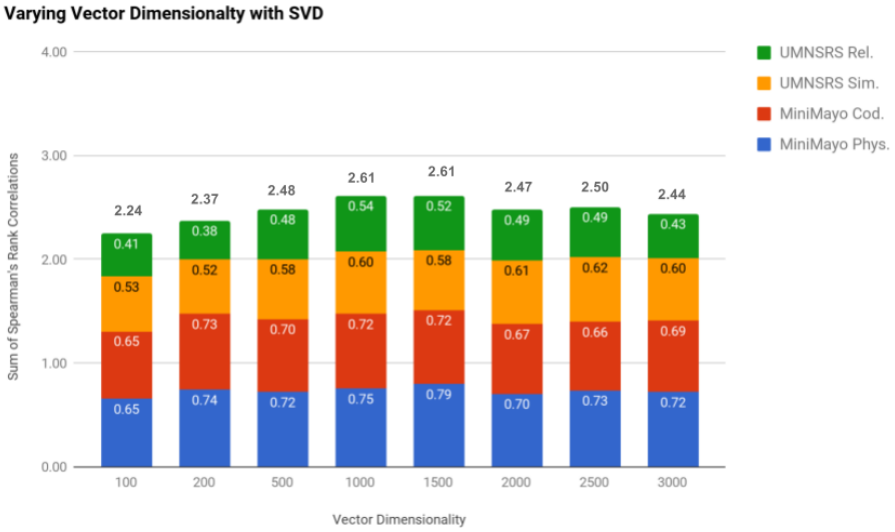

Fig. 18. Sum of results across datasets for each vector dimensionality tested with SVD. Each column corresponds to a vector dimensionality, individual dataset correlations are shown within the colored rectangles, and sum of correlations are shown above each column.

To further analyze the aggregation methods, we set the dimensionality reduction technique, and vector dimensionality parameters to our recommendations found in previous subsections. That is, we compared results of each multi-word term aggregation method using dimensionality reduction of word embeddings using CBOW and a vector dimensionality of 200. Table 16 (restated from Table 13) shows these results. Table 17 shows the statistical significance between these results. No multi-word term aggregation method achieves the highest correlation on all datasets, and each method achieves (or ties) the highest result for one of the datasets. There is no statistical significance between any of the techniques on any of the datasets. After preprocessing, the computational cost of each method is nearly identical. Ambiguity is only an upfront problem for concept vectors, but they also have the advantage of mapping synonymous terms to the same concept before vector construction. This is likely why

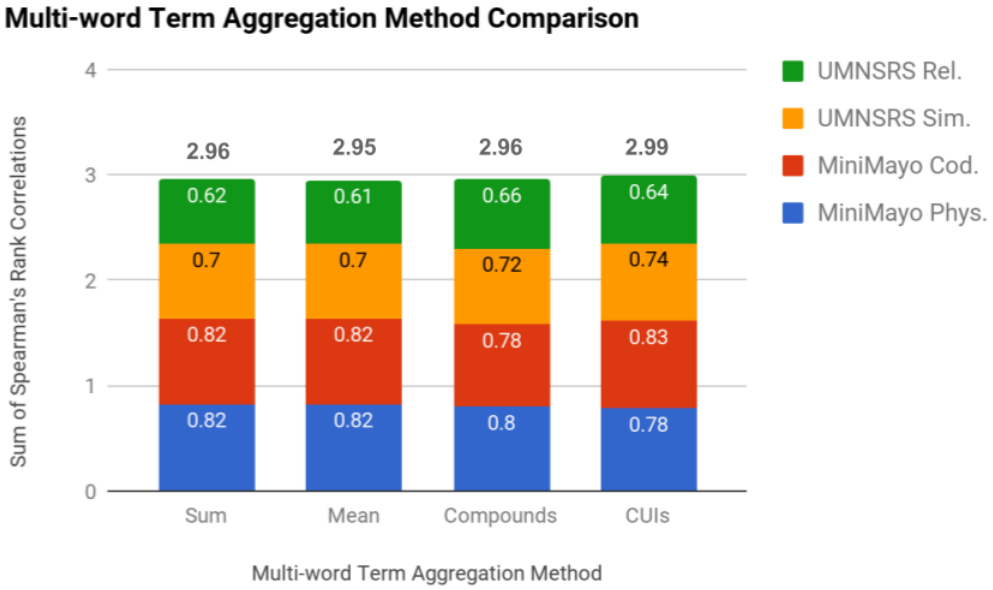

Fig. 19. Results of different aggregation methods on each dataset.

concept vectors (CUI) achieve the highest correlation on two datasets (MiniMayo Cod, and UMNSRS Rel), and is why they are able to compare more terms (higher n). Concept mapping is however, an expensive preprocessing step, and it is unclear how to best handle ambiguous concept mappings. Using compounds requires less expensive data preprocessing, and since it does not map synonymous terms to the same concept, the problem of ambiguity is essentially ignored. Using sum or mean requires the least preprocessing, and does not rely on external lexicons to construct mutli-word term vectors, ambiguity is ignored, and vectors can be created for any term in any domain. **Conclusion:** There is no significantly best multi-word term aggregation method across either all parameters or recommended parameters. Using concept vectors achieves the highest correlation on more datasets and it is able to compare more terms. It is therefore slightly favored, but requires the term to map to a concept. Sum or mean provide more flexibility and less preprocessing.

MiniMayo Phys

	sum	mean	compound
mean	1.0	-	-
comp.	0.8337	0.8337	-
CUI	0.6892	0.6892	0.8493

MiniMayo Cod

	sum	mean	compound
mean	1.0	-	-
comp.	0.5823	0.5823	-
CUI	0.2543	0.2543	0.5552

UMNSRS Sim

	sum	mean	compound
mean	1.0	-	-
comp.	0.5823	0.5823	-
CUI	0.2543	0.2543	0.5552

UMNSRS Rel

	sum	mean	compound
mean	0.8181	-	-
comp.	0.3421	0.242	-
CUI	0.6384	0.4839	0.6241

Table 15. The two tailed p-values using Fishers R-to-Z transform of each term aggregation method's correlation scores of each dataset. Each table corresponds to a different dataset, each row and column a different term aggregation method.

Summary

In this section, we explored methods to create distributional context vectors of multi-word terms for the task of direct semantic relatedness. We used direct co-occurrence vectors (direct) as a baseline, and applied dimensionality reduction using singular value decomposition (SVD), word embeddings with skip-gram (SG), and word embeddings with CBOW (CBOW). We found that all dimensionality reduction techniques improved correlation accuracy, and that SG and CBOW perform the best. Since SG takes much longer to train, we favor using CBOW in future experiments. We varied the dimensionalities of the vectors, and found that a dimensionality of 200 is sufficient for SG and CBOW, and a larger dimensionality of 1000 is sufficient for SVD.

CBOW Results at Vector Dimensionality of 200

	sum	mean	compound	CUI
MiniMayo Cod	0.82/29	0.81/29	0.78/28	0.83/29
MiniMayo Phys	0.82/29	0.82/29	0.80/28	0.75/29
UMNSRS Sim	0.61/396	0.59/397	0.64/393	0.56/413
UMNSRS Rel	0.69/374	0.69/374	0.70/373	0.73/388

Table 16. Results of each multi-word term aggregation technique using the recommended settings of word embeddings using CBOW at a vector dimensionality of 200.

MiniMayo Phys

	sum	mean	compound
mean	1.0	-	-
comp.	0.4168	0.4168	-
CUI	0.5093	0.5093	0.8337

MiniMayo Cod

	sum	mean	compound
mean	0.9124	-	-
comp.	0.6892	0.3859	-
CUI	0.9124	0.8259	0.6101

UMNSRS Sim

	sum	mean	compound
mean	0.6599	-	-
comp.	0.4902	0.2585	-
CUI	0.2801	0.5222	0.0767

UMNSRS Rel

	sum	mean	compound
mean	1.0	-	-
comp.	0.7949	0.7949	-
CUI	0.2670	0.2670	0.4009

Table 17. The two tailed p-values using Fishers R-to-Z transform of the correlation scores of multi-word term aggregation methods using recommended parameters. Each table corresponds to a different dataset, each row and column a different term aggregation method using CBOW and a dimensionality of 200.

In our evaluation of multi-word term aggregation methods, we compared using text with compound words identified, text mapped to concepts, text with individual words for which vectors are combined via mean or summation operations. We found that none of these methods are significantly better than another, indicating that vector representations are robust to data preprocessing and vector combination methods. In our development of indirect relatedness measures, we use CBOW vectors with a dimensionality of 200, since CBOW and SG performed the best, and because using UMLS concepts allows additional information from the UMLS such as semantic types and semantic groups to be incorporated. Additionally, we use direct co-occurrence vectors as a baseline.

4.3 Comparison with Related Work

In this section, we summarize and compare other state of the art semantic similarity and relatedness measures to our association measures and vector cosine results. We generate results using the experimental set up as described in Sections 4.1.4 and 4.2.2.

Although previous work explores different parameters and corpora, multi-word term aggregation methods have been ignored, and multi-word terms are often dropped from the evaluation standards to account for the inability to represent such terms [105, 106, 107]. Table 18 summarizes the contributions of previous authors' work with word embeddings for semantic similarity and relatedness.

Sajadi, et al. [108] use the OHSUMED corpus, a collection of 348,566 biomedical research articles to train the word2vec [23] skip-gram model over UMLS concepts identified by MetaMap. They also develop the HITS-sim algorithm, a graph-based similarity metric based on Wikipedia hyperlinks. They show that incorporating an ontology into vector based methods can improve results, and that using specialized

	Method			
Citation	CBOW	SG	Training Dataset(s)	Hyperparams
This Work	X	X	MEDLINE	X
Sajadi, et al. [108]		X	OHSUMED	
Muneeb, et al. [105]	X	X	PMC	X
Chiu, et al. [106]	X	X	Pubmed, PMC	X
Pakhomov, et al. [107]	X		Fairview, PMC, Wiki	

Table 18. Summary of related work using word embeddings for semantic relatedness in the biomedical domain. The citation column indicates the author and reference. The method column indicates whether the author used CBOW, skip-gram (SG), or both for their evaluation. The training dataset(s) column shows the training corpora used in the experiments, and the hyperparams column indicates whether the author reported results with various hyperparameter settings (e.g. vector dimensionality, window size, etc.).

corpora (OHSUMED) performs comparably to general domain resources (Wikipedia).

Muneeb, et al. [105] trained both the skip-gram and continuous bag of words (CBOW) word2vec models over the PubMed Central Open Access (PMC) corpus of approximately 1.25 million articles using a window size of 9. They evaluated the models on a subset of the UMNSRS data, removing word pairs that did not occur in the corpus more than ten times. The authors evaluated vector dimensionalities of 25, 50, 100, and 200, finding that a dimensionality of 200 performed better than the lower dimensionalities. They report a correlation with the UMNSRS tagged for similarity of 0.46 with CBOW and 0.52 with skip-gram; and a correlation with the UMNSRS tagged for relatedness of 0.41 with CBOW and 0.45 with skip-gram.

Chiu, et al. [106] evaluated both the skip-gram and CBOW word2vec models over the PMC corpus and PubMed. They evaluated a lot of hyper-parameters in-

cluding vector dimensionality, windowing, learning rate, and minimum count. They used a subset of the UMNSRS dataset that ignores word pairs that do not exist in the corpus. They found that using PubMed obtained a higher correlation than PMC or a combination of PMC and PubMed. They also found that although the hyper-parameter settings can improve the performance of the model, the effects vary. They find a dimensionality of 400 and 500 for UMNSRS Sim and Rel are the best respectively, however their extrinsic evaluations show a dimensionality of 200 is the best. Upon closer examination, the dimensionality of 400 and 500 are almost certainly not statistically significantly different from their reported results with a vector dimensionality of 200 (n is not reported, therefore statistical significance cannot be exactly computed). They find the skip-gram model outperforms CBOW on UMNSRS.

Pakhomov, et al. [107] trained CBOW word2vec over three different types of corpora: clinical (clinical notes from the Fairview Health System), biomedical (PMC corpus), and general English (Wikipedia). They evaluated the method using a subset of the UMNSRS restricting to single word term pairs. Using a window size of 8, and a dimensionality of 200, they create CBOW vectors with each corpus, and evaluate on UMNSRS. They find that the model trained on the PMC corpus obtained the highest correlation for both similarity and relatedness, with a correlation of 0.62, and 0.58 respectively. They explored varying the corpus size for the clinical data, and found that the results increased as they systematically increased the corpus size from 1 million to over 4 billion tokens. They also evaluate on several extrinsic tasks. Although techniques and parameters differ between other authors and us, our results with multi-word terms achieve comparable results to previous work evaluated on single word terms.

Yu, et al. [109] retrofit distributional word vectors with hierarchical information from the MeSH hierarchy of the UMLS. In their method, word vectors are built

using observed co-occurrence information from a corpora, and inferred word vectors are constructed by incorporating hierarchical information. The inferred vectors are expected to be close to both their observed word vectors and connected (based on the hierarchy) inferred vectors. The final inferred vectors are constructed by minimizing an objective function.

Workman, et al. [110] use SemRep predications to generate similarity and relatedness scores of CUIs using a variety of metrics including: semantic predication frequency, Pointwise Mutual Information, Information Radius, and the L1 norm distance.

Table 19 summarizes our results and results from other authors. The table shows the Spearman's rank correlation coefficient followed by the number of terms compare in parentheses. A '-' indicates results were not reported for that method. Table 19 is divided into several sections: The topmost section shows our results from association scores. These are our best results using any parameter settings (*Association Best*) and our best results using our recommended parameters (*Association Recommended*) of a co-occurrence window size of 8 (window 8), ignoring word order (no order), not applying a minimum co-occurrence threshold, not using concept expansion, and the Pearson's chi squared association measure. The section below that shows our results with vector representation of terms. These are our results using word2vec word embeddings using CBOW and a vector dimensionality of 200, trained over with words (*CBOW words*), compoundified (*CBOW comp.*), and with CUIs (*CBOW cuis*). The section below that shows results from other authors (see Section 4.3) that are not word2vec implementations, and below that from authors using word2vec word embeddings with different parameters and datasets than our own. Chiu, et al. do not report n, so we list this as "n/a". The bottom box shows results we generated results for several similarity and relatedness measures as implemented in

the UMLS::Similarity [111] package version 1.47 using the default parameters unless specifically stated below. We used the SNOMED CT with PAR/CHD relations for the path–based similarity measures (Path [25]), information content (IC)-based measures (Resnik [27], Lin [26]), and the entire UMLS for the relatedness measures (Lesk [112], Vector [113]).

	Data Set			
Measure	MiniMayo Phy	MiniMayo Cod	UMNSRS Sim	UMNSRS Rel
Association Best	0.85 (29)	0.84 (29)	0.73 (392)	0.66 (418)
Association Recommended	0.84 (29)	0.81 (29)	0.69 (392)	0.64 (418)
CBOW words (ours)	0.82 (29)	0.82 (29)	0.61 (396)	0.69 (374)
CBOW comp. (ours)	0.80 (29)	0.78 (28)	0.65 (393)	0.70 (373)
CBOW cuis (ours)	0.77 (29)	0.83 (29)	0.60 (413)	0.73 (388)
Yu, et al. [109]	0.70 (25)	0.67 (25)	-	-
Sajadi, et al. (HITS-sim) [108]	0.67 (29)	0.72 (29)	0.58 (566)	0.51 (587)
Workman, et al. [110]	0.67 (29), 0.69 (25)	-	-	-
Sajadi, et al. (word2vec) [108]	-	-	0.39 (566)	0.39 (587)
Pakhomov, et al. [107]	-	-	0.62 (449)	0.58 (458)
Muneeb, et al. [105]	-	-	0.52 (462)	0.45 (465)
Chiu, et al. [106]	-	-	0.65 (n/a)	0.60 (n/a)
UMLS::Similarity Path [25]	0.35 (26)	0.44 (26)	0.53 (340)	0.29 (360)
UMLS::Similarity Resnik [27]	0.34 (26)	0.46 (26)	0.49 (340)	0.26 (360)
UMLS::Similarity Lin [26]	0.42 (26)	0.53 (26)	0.49 (340)	0.29 (360)
UMLS::Similarity Lesk [112]	0.52 (29)	0.57 (29)	0.50 (387)	0.33 (412)
UMLS::Similarity Vector [113]	0.59 (29)	0.58 (29)	0.58 (387)	0.45 (412)

Table 19. Our best and recommended parameters results for association measures and vector representations compared to state of the art methods from other authors. "-" indicate that results were not reported for that dataset. Each row shows results using a different technique, and each column corresponds to a different dataset. The Spearman's correlation coefficient is shown, followed by n, the number of terms compared.

These results are not directly comparable to each other, or ours due to different number of term pairs compared. This is due to differences in corpora and preprocessing methods. Still, the results indicate that we achieve state of the art performance using both association measures, and vector cosine, and that vector measures and association measures achieve comparable performance.

4.4 Conclusions

In this chapter we achieved state of the art performance at estimating direct semantic relatedness using two different methods, association measures and vector cosine. Since estimating direct semantic relatedness is a well studied field, it serves as a good starting point for the development of indirect relatedness measures and their application to LBD. Contributions of this chapter included a parameter analysis for association measures and state of the art results for estimating semantic relatedness, concept associations, concept expansion, and set associations. A parameter analysis of vector representations for estimating semantic relatedness, including the effects of multi-word term aggregation methods, dimensionality reduction techniques, and vector dimensionality, and a comparison of our results to other state of the art methods. These contributions are interesting for direct relatedness measures, and its applications, but are also interesting for our overall goal of developing more effective LBD systems. These results show the effectiveness of association measures at estimating semantic relatedness, indicating that they are good candidates for their development as indirect association measures. Vector measures can intrinsically compute indirect association, and therefore are good candidates for indirect association measures without further modification.

Specifically, these results indicate that using MetaMapped text is a good idea. Association measures perform significantly better using concepts rather than terms, and there is no statistical difference for vector representations. Additionally, we found that using the 1975 onward subset of MEDLINE is sufficient. Through an extensive parameter analysis, we found a recommended set of parameters for association measures, which include using a window size of 8 and word order ignored, do not use concept expansion, apply a minimum co-occurrence threshold of 1, and use

the Pearson's chi squared (χ^2) association measure. We found our preferred vector representation, which is word2vec CBOW with a vector dimensionality of 200. These findings inform our development and analysis of indirect relatedness measures and their application to LBD in Chapter 5.

CHAPTER 5

INDIRECT RELATEDNESS

In this chapter, we address our first critical problem of LBD, the over-generation of knowledge by LBD systems, by creating more effective LBD hypothesis ranking methods. Using these methods, we can rank and filter uninteresting, false, or too obvious hypotheses. In their simplest form, hypotheses can be represented as start term-target term pairs, and we hypothesize that the likelihood that the term pair represents a true and interesting future relationship can be estimated by the strength of the future relatedness between the start and target term. We view prediction of future relatedness as a task of estimating the relatedness between two terms which never co-occur in a corpus.

Since estimating indirect relatedness is not a well studied field, we first analyzed the performance of several methods for estimating direct relatedness in Chapter 4. That laid the groundwork for our development of indirect relatedness measures presented in this chapter. Specifically, we showed in Chapter 4 that association measures are good indicators of semantic relatedness, but were designed for direct relatedness, rather than indirect relatedness. In this chapter, we present indirect association measures, which incorporate connecting term information to quantify the relatedness between terms that never directly co-occur. Specifically, in this chapter we introduce four indirect association measures:

1. Linking term association (LTA), which quantifies association using counts of unique connecting terms.

2. Minimum weight association (MWA), which quantifies association using co-

occurrence counts of A-B-C pathways.

3. Shared B to C association (SBC), which quantifies association using the set of shared B terms as a proxy for A.

4. Linking set association (LSA), which quantifies association using the sets of terms A and C co-occur with as proxies for themselves.

Since it is unclear how to best evaluate a method's ability to estimate indirect relatedness, we evaluate each indirect relatedness measure using three evaluation methods: (1) their ability to estimate direct semantic similarity and relatedness, (2) their ability to distinguish between term pairs that will or will not be related in the future in a time-slicing based link prediction task, and (3) their ability to predict future semantic relatedness using a time-slicing based approach.

Specific contributions of this chapter are:

1. The development of four indirect association measures, including LTA, MWA, SBC, and LSA.

2. The development of a dataset and method (cumulative relatedness graphs (CRGs)) for evaluating the ability to estimate future relatedness.

3. A evaluation of indirect association measures and vector-based methods on their ability to estimate direct relatedness, for link prediction, and on their ability estimate future relatedness.

In this chapter, we first introduce our indirect association measures. Next, we explain our evaluation procedures, including CRGs, and our experimental details. Next, we show results for each evaluation method, and lastly we discuss conclusions and limitations.

5.1 Indirect Association Measures

To quantify indirect relatedness, we developed four indirect association measures. Indirect association measures are modifications of direct association measures, which are presented in detail in Chapter 2. Both direct and indirect association measures are based on co-occurrence statistics in a corpus. They quantify two terms' expected co-occurrence together by chance versus their observed co-occurrence together in text, and follow the calculation process shown both in Chapter 2, and summarized in Figure 20. The difference is direct association measures require that the two terms co-occur together in a corpus, and indirect association measures do not.

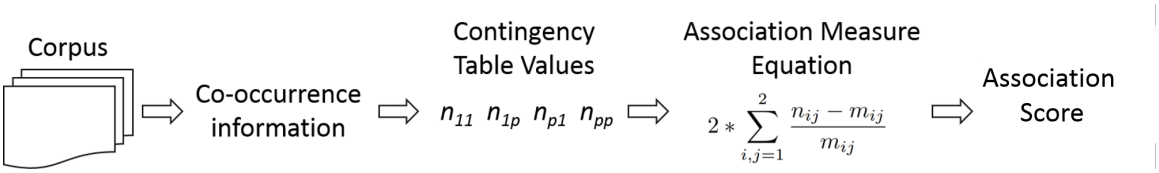

Fig. 20. The association measure calculation process. In this process, co-occurrence counts are collected from a corpus, a contingency table is generated, and association is quantified. The equation shown is Pearson's chi squared, where m_{ij} values are: $m_{11} = \frac{n_{1p} * n_{p1}}{n_{pp}}$, $m_{12} = \frac{n_{1p} * n_{p2}}{n_{pp}}$, $m_{21} = \frac{n_{p1} * n_{2p}}{n_{pp}}$, $m_{22} = \frac{n_{p2} * n_{2p}}{n_{pp}}$ and $n_{2p} = n_{pp} - n_{1p}$, $n_{p2} = n_{pp} - n_{p1}$

As previously described in Section 2.5, co-occurrence counts are collected from a corpus, and used to populate a contingency table. The contingency table values, n_{11}, n_{1p}, n_{p1}, and n_{pp} are input into an association measure equation, such as Pearson's Chi-Squared [33] (shown in Figure 20) to produce a single number that quantifies the association between two terms. For indirect association measures, we modify how the contingency table values are calculated prior to input into the association measure equation. These modifications make it possible to quantify the association between indirectly related terms while preserving the beneficial statistical properties encoded

in association measure equations. We modify the contingency table values as follows:

1. **Minimum weight association (MWA):** modifies n_{11} as the average minimum co-occurrence for each A-to-B-to-C pathway.

2. **Linking term association (LTA):** uses the counts of unique linking terms to populate the contingency table.

3. **Shared B to C association (SBC):** uses the co-occurrences between the shared B term set and C to populate the contingency table.

4. **Linking set association (LSA):** uses co-occurrences between terms that co-occur with A (B_A) and terms that co-occur with C (B_C) to populate the contingency table.

MWA and LTA are similar, since rather than using direct A to C co-occurrences, they combine A to B and B to C co-occurrence information. MWA uses $A - B$ and $B - C$ co-occurrence counts, while LTA uses the count of unique B terms. SBC and LSA are similar, since they are based on set associations. SBC uses the shared B terms set as a proxy for A when collecting co-occurrence counts, and LSA uses B_A and B_C as proxies for A and C respectively when collecting co-occurrence counts.

In the next few subsections, we give detailed explanations on how each association measure is calculated. We define the following terminology: A is the set of starting term(s); B_A is the set of terms preceded by A (A's connecting terms); C is the set of target terms; B_C is the set of terms preceding C (C's connecting terms); V is the set of all terms (the vocabulary); w_{ij} is the weight of the edge going from node i to node j in the co-occurrence graph, and is the frequency that term i is followed by term j. We also use Figure 21 as an example. Figure 21 shows a co-occurrence graph between term A and term C. A and C directly co-occur with a set of connecting terms, B.

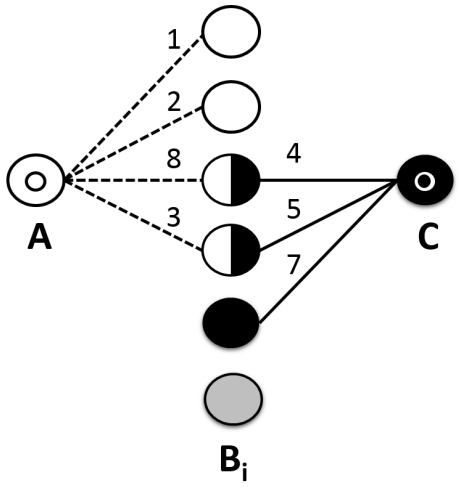

Fig. 21. A co-occurrence graph showing A and C co-occurrences with a set of B terms

A subset of which co-occur with both A and C; an indirect relationship is created through this set of shared B terms. Each circle in Figure 21 indicates a unique term, and each edge indicates a direct co-occurrence. The number above each edge indicates that edge's co-occurrence frequency. The terms A co-occurs with are shown as white and black-and-white circles. The terms C co-occurs with are shown as black and black-and-white circles. White B terms co-occur with A only, black B terms co-occur with C only, and black-and-white terms are shared connecting terms that co-occur with both A and C. Gray B terms exist in the vocabulary, but do not co-occur with A or C. Figure 21 therefore shows that A co-occurs with four different terms, it co-occurs with the top-most B term one time, the second top-most B term two times, the B term below that eight times, and the term below that three times. C co-occurs with three different terms, four times, five times, and seven times respectively, and there is a single B term that co-occurs with neither A or C.

Using Figure 21 as an example, we define direct association contingency table values as follows:

n_{11} (Equation 5.1) is the sum of weights (co-occurrences) between A and C. In

Figure 21, $n_{11} = 0$, since A and C never directly co-occur.

$$n_{11} = \sum^{a \in A} \sum^{c \in C} w_{ac} \tag{5.1}$$

n_{1p} (Equation 5.2) is the sum of weights between A and each B_i. In Figure 21, $n_{1p} = 1 + 2 + 8 + 3 = 14$.

$$n_{1p} = \sum^{a \in A} \sum^{j \in V} w_{aj} \tag{5.2}$$

n_{p1} (Equation 5.3) is the sum of weights between each B_i and C. In Figure 21, $n_{p1} = 4 + 5 + 7 = 16$.

$$n_{p1} = \sum^{j \in V} \sum^{c \in C} w_{jc} \tag{5.3}$$

n_{pp} (Equation 5.4) is the sum of all weights in the dataset, which is the total number of co-occurrences between all terms.

$$n_{pp} = \sum^{i \in V} \sum^{j \in V} w_{ij} \tag{5.4}$$

Using these four contingency table values, we can calculate the rest of the values in a contingency table, and calculate an association measure equation, such as Pearson's chi squared (shown in Figure 20) to test for association between A and C with a single value. Start-target term pairs in LBD explicitly do not co-occur, meaning $n_{11} = 0$ for all start-target term pairs. This is shown in Figure 21, where A and C never co-occur. Since there is no direct co-occurrence between the terms, we develop indirect association measures which incorporate additional information to quantify the association between two terms which do not directly co-occur, but are instead, indirectly related.

5.1.1 Minimum Weight Association

Minimum Weight Association (MWA) calculates association between A and C based on the information flow between them relative their co-occurrences with all terms in the dataset. It uses co-occurrence counts to populate the contingency table, however we modify the value of n_{11} to allow indirect associations to be quantified. We can view each A to B_i to C link as a weighted path connecting A and C, and use the co-occurrence information along this path to calculate n_{11}. The question becomes how to combine the $A-B_i$ and B_i-C weights. One approach may be to sum, average, or take the maximum value of weights along a path, but association measures require that $n_{11} \leq n_{1p} \leq n_{pp}$ and $n_{11} \leq n_{p1} \leq n_{pp}$; sums, averages, or maximums may violate this. Therefore, for MWA we take the minimum value along each $A - B_i - C$ path, and sum over all $A - B_i - C$ pathways. If we imagine the co-occurrence counts along each $A - B_i - C$ pathway as information flowing between A and C, then summing the co-occurrence counts is analogous to finding the total information flow between A and C, where each $A - B_i - C$ pathway cannot carry more than its minimum capacity. n_{1p}, n_{p1}, and n_{pp} remain unchanged from direct association measures. The contingency table values are defined formally as:

n_{11}, (Equation 5.5) is the sum of minimum A to B_i and B_i to C weights of each shared B_i term. In Figure 21, $n_{11} = min(8, 4) + min(3, 5) = 7$.

$$n_{11} = \sum^{b \in B_A \cap B_C} min(\sum^{a \in A} w_{ab}, \sum^{c \in C} w_{bc}) \tag{5.5}$$

n_{1p}, (Equation 5.2) remains unchanged from direct association measures. It is the sum of A to B_i weights. In Figure 21, $n_{1p} = 1 + 2 + 8 + 3 = 14$.

n_{p1}, (Equation 5.3) remains unchanged from direct association measures. It is the sum of B to C weights. In Figure 21, $n_{p1} = 4 + 5 + 7 = 16$.

n_{pp}, (Equation 5.4) remains unchanged from direct association measures. It is the sum of all possible weights (total co-occurrence count) of the whole dataset.

5.1.2 Linking Term Association

Linking term association (LTA) quantifies the association between A and C based on the counts of their linking terms. It combines the empirically proven performance of Linking Term Count (LTC) [62] (see Section 3.3.1) with the statistical properties of association measures. Rather than using co-occurrence counts for contingency table values, LTA uses counts of unique co-occurring terms. If we view the co-occurrence graph in Figure 21 as an unweighted graph, the contingency table value equations for LTA are identical to MWA, but it is perhaps more intuitive to define these values in terms of set theory.

n_{11}, (Equation 5.6) is the count of unique shared linking terms. In Figure 21, $n_{11} = 2$.

$$n_{11} = |B_A \cap B_C| \tag{5.6}$$

n_{1p}, (Equation 5.7) is the count of unique terms A co-occurs with. In Figure 21, $n_{1p} = 4$.

$$n_{1p} = |B_A| \tag{5.7}$$

n_{p1}, (Equation 5.8) is the count of unique terms C co-occurs with. In Figure 21, $n_{p1} = 3$.

$$n_{p1} = |B_C| \tag{5.8}$$

n_{pp}, (Equation 5.9) is the count of all possible unique terms (vocabulary size).

$$n_{pp} = |V| \tag{5.9}$$

In this formulation, the value of n_{11} is equivalent to the LTC between A and C,

but we weight the LTC by the number of terms A and C independently co-occur with in the association measure equation. This makes the associations between terms that co-occur with many terms lower than those that co-occur with a few.

5.1.3 Shared B to C Association

Shared B to C association (SBC) quantifies the association between A and C as the set association between their shared B terms and C itself. It builds upon the idea of set associations (introduced in Chapter 4) by using the shared B terms between A and C as a proxy for the A term, then calculates the direct set association. That is, we populate the contingency table using the co-occurrences between the shared B term set and C, rather than between A and C. Equation 5.10 formally defines SBC in terms of set theory, where $assoc()$ is an association measure equation such as Pearson's chi squared.

$$assoc(B_A \cap B_C, C) \tag{5.10}$$

5.1.4 Linking Set Association

Linking Set Association (LSA) quantifies association between A and C using the set association between B_A and B_C. Like SBC, LSA is based on direct associations between sets of terms, but for LSA, we replace both A and C. A is replaced with the set of all terms that it co-occurs with (B_A), and C is replaced with the set of all terms that it co-occurs with (B_C). The association between these two sets is then calculated. Equation 5.11 formally defines LSA in terms of set theory.

$$assoc(B_A, B_C) \tag{5.11}$$

Intuitively, both SBC and LSA are estimating A and C with a set of terms. LSA

estimates A and C with their co-occurrences, which is indicative of their context, and therefore their meaning. SBC estimates A with respect to its shared co-occurrences with C, and uses C directly. In other words, the association between how A's meaning overlaps with C's, and C itself. LSA defines its proxies in terms of their own independent contexts, where as SBC defines the proxies in terms of the more constrained, shared context.

5.2 Evaluation Methods

Since it is our hypothesis that ranking in LBD should be based on estimating the strength of a relationship between two unrelated terms, ranking methods should perform well at estimating direct as well as indirect semantic relatedness. Estimating direct semantic relatedness is a well established field with standard evaluation datasets.

Evaluating indirect relatedness measures on their ability to predict direct relatedness is not sufficient in isolation. They should also be able to determine if a relationship exists between two indirectly related terms, and if so the strength of that relationship. Since our application of indirect association measures is for ranking target terms in LBD, we use time-slicing based techniques to evaluate this, and create a co-occurrence time-slicing based dataset. We divide the dataset into pre- and post-cutoff datasets, and predict whether a relationship between two terms in the post-cutoff dataset for which no relationship exists in the pre-cutoff dataset. Effectively we are estimating if a future relationship exists between two terms, and therefore constitutes a discovery.

Previous work [85, 89, 87] poses this as a link prediction task and evaluate it using receiver operating characteristic (ROC) curves. ROC curves indicate whether a measure can distinguish between terms that will or will not have a future rela-

tion. This is an important feature of an indirect relatedness measure, particularly for hypothesis filtering, where false hypotheses are removed from our list of LBD predictions.

ROC curves, however do not explicitly take rank into account, meaning they do not evaluate an indirect relatedness measure's ability to distinguish between strong and weak future relations. This is important both due to the intrinsic value in quantifying relatedness (as is the case with direct relatedness measure evaluation), but also specifically for LBD, where gold standard datasets are noisy, and often contain many false positive future relationships. Presumably a true future relationship will have a higher relatedness score than a false future relation, and we can estimate the likelihood of that relationship being true by the start and target term pair's indirect relatedness. To evaluate a measure's ability to estimate future relatedness, we develop cumulative relatedness graphs (CRGs).

In the following subsections, we describe each evaluation method in detail, these include: 1) estimating direct semantic relatedness, 2) link prediction using ROC curves, and 3) estimating future relatedness using CRGs.

5.2.1 Evaluation Details

Evaluation Details: Estimating Direct Semantic Relatedness. We evaluate indirect relatedness measures on their ability to estimate direct semantic similarity and relatedness as described in detail in Section 2.4.1. We use the reference standards of: the UMNSRS [31] tagged for similarity (UMNSRS Sim), the UMNSRS tagged for relatedness (UMNSRS Rel), the MiniMayoSRS dataset [114] rated by medical coders (MiniMayo Cod) and the MiniMayoSRS rated by physicians (MiniMayo Phys). We report Spearman's Rank Correlation Coefficient (ρ) between the scores generated for each term pair and these gold standards scores.

The use of these datasets is common, but the MiniMayoSRS and UMNSRS datasets contain term pairs of different UMLS semantic groups, concept pairs of the same semantic group, and synonymous term pairs. LBD is often used to find new treatments (chemicals and drugs) for diseases or disorders, and we are therefore only interested in term pairs of different semantic types, specifically, chemicals and drugs - diseases or disorders term pairs. Therefore, we select concepts from the semantic groups of *Disorders* and *Chemicals and Drugs* to generate 113 and 126 *Disorders-Chemicals and Drugs* or *Chemicals and Drugs-Disorders* concept pairs in the UMNSRS Sim and Rel datasets respectively [115]. In addition to results using the full datasets, we report the results with these two subsets. The results with the UMNSRS Rel subset are particularly relevant to LBD, and although we also report results with the UMNSRS Sim subset for completion, we question how meaningful they are. The UMNSRS Sim was tagged for similarity, and since similar terms are linked by isA type relationships, all *Chemicals and Drugs-Disorders* concept pairs should have low scores.

Evaluation Details: Link Prediction. As described in Section 3.3.2, we evaluate indirect relatedness measures on their ability to distinguish between true and false future predictions on a link prediction task using ROC curves. ROC curves plot the trade-off between true and false positive rates, typically to evaluate the performance of binary classifiers. We generate ROC curves by ranking all future relationships with an indirect relatedness measure and apply a threshold. All relationships above this threshold are considered true, and all relationships below it are considered false. The corresponding true and false positive rates are calculated and plotted. This threshold is applied at every rank to produce a single ROC curve. The area under the ROC (AUROC) curve can be calculated to give a single number that summarizes performance. ROC curves are calculated in a standard manner, but we

ensure that relationships with tied ranks are penalized. This is a simple modification, where in the event of a tie, the worst case scenario is always used, meaning we make sure false pairs are always ranked higher in the ROC curve calculation.

Evaluation Details: Estimating Future Relatedness. We evaluate indirect relatedness measures on their ability to estimate future relatedness using our novel evaluation method, Cumulative Relatedness Graphs (CRGs). To create a CRG, we rank a set of future relationships weighted by the future relatedness between its start and target term. Each point on the CRG shows the sum of weights for the relationship at that rank and higher. Better measures will rank relationships with higher weights first, and will have a higher cumulative relatedness at each rank.

CRGs are an extension of established mutual information (MI) graphs [63] (explained in detail in Section 3.3.2.6), which were proposed to evaluate LBD hypothesis ranking algorithms. Their key observation is that the MI between the start and target term of an LBD hypothesis can be used as an estimate of relevance or irrelevance, and that term pairs with high MI should be ranked higher than those with low MI. Established MI graphs however, only evaluate using directly related terms which isn't representative of LBD's indirectly related terms, and only use a single start term, so they don't show generalized performance.

We develop CRGs by taking their idea of using direct relationships to estimate relevance, and incorporating time-slicing techniques [116] and direct future relatedness. CRGs use the future relatedness between many terms to create an evaluation technique relevant to LBD. MI can be used as an estimate of semantic relatedness, however we found in Chapter 4 that Pearson's chi squared performs better. CRGs are generated in four steps:

1. Generate future relationships

2. Weight future relationships

3. Relationship ranking

4. Create cumulative relatedness graph

Step 1: Generate future relationships: We generate a set of future relationships in a time-slicing manner by dividing the dataset into pre- and post-cutoff segments. All relationships are extracted from the pre- and post-cutoff segments, and the gold standard is created by removing all pre-cutoff relationships from the set of post-cutoff relationships. We represent relationships as start-target term pairs.

We select 200 start terms (represented as UMLS Concept Unique Identifiers (CUIs)) by randomly selecting 50 terms from each of the semantic types of: Clinical Drug (T200, clnd), Pharmacologic Substance (T121, phsu), Disease or Syndrome (T047, dsyn), Sign or Symptom (T184, sosy). Target terms are generated for each term pair as all terms in the vocabulary. The result is a set of all possible relationships between each start term and every term in the vocabulary. We remove all co-occurring term pairs in the pre-cutoff dataset and end up R, a set of 67,754,497 possible relationships for which no direct co-occurrence exists.

Step 2: Weight future relationships: We assign a weight, w_{ij}, as the future relatedness between the start term, s_i and the target term, t_j in R. We use Pearson's chi squared [33] as our weighting function, due to its good performance in Chapter 4 at estimating direct semantic relatedness. Lastly, we normalize w_j by the total future relatedness between s_i and all possible target terms, T, such that, $w_{ij} = \frac{w_i j}{\sum_{k=0}^{|T|} w_i k}$, the percentage of total future relatedness t_i contributes.

Step 3: Relationship ranking: We calculate the indirect relatedness between each term pair in R. So, for each start term, s_i we rank all start-target term pairs to create R_i, a ranked list of s_i-tj term pairs for each s_i.

Step 4: Create cumulative relatedness graph creation: We generate a CRG using the gold standard weights (w_{ij}) from Step 2, and the ranked hypotheses (R_i) from Step 3. For each set of ranked Hypotheses R_i in R, we create a CRG and create an average of these to create our final CRG. Each CRG is created using the ranks of R_i and the weights w_{ij}. We calculate the accumulated sum of weights at each rank, such that at rank l, the point in the CRG $c_{il} = \sum_{k=0}^{l} w_{ij}$. With each c_i, we calculate the CRG averaged over all R_i, such that each point on the CRG, $c_{\mu j} = \sum_{l=0}^{n} \frac{c_{il}}{n}$. Not all R_i are the same size, so if c_{il} is undefined meaning $l > lmax$, then $c_{il} = c_{ilmax}$, the total weight accumulated for that H_i. The result is a single CRG averaged over the all future relationships of n start terms.

The result is a CRG that shows the rate at which the total future relatedness in a dataset is reached. CRGs reward methods which rank relationships with a high future relatedness highest, and penalize methods that rank relationships with low future relationships high. CRGs are more robust to noise since they don't assume the correctness of a gold standard, but instead weight the gold standard by the future relatedness.

CRGs were designed to evaluate LBD, and they explicitly use rank while evaluating. The rank of the most confident and important relationships contribute the most to a system's overall performance. Their high weights also lessen the effect of the long tail of uninteresting terms caused by the Zipfian nature of language; the most interesting part of the dataset is weighted more, while performance on the dataset as a whole still has some influence.

5.3 Experimental Details

In this section, we describe the specific implementations and datasets use to generate results. All evaluation datasets use a set of UMLS concept-concept pairs,

and our selection of parameters is based on our analysis of direct relatedness measures in Chapter 4.

Baseline Measures: Vector measures can inherently quantify indirect relatedness, and in Chapter 4 we performed an analysis of vector representations for estimating direct relatedness. In this chapter we take what we learned to study the ability of concept embeddings vector cosine [23] and the explicit vector representation of direct co-occurrence vector cosine [117] at estimating indirect relatedness. We also use linking term count (LTC) [60] as a baseline, because it has been shown to be one of the best performing target term ranking methods in LBD [62].

Corpus: All indirect relatedness methods and baselines rely on a corpus, for which we use the 2015 *MetaMapped MEDLINE baseline*. For the time-slicing tasks of link prediction and estimating future relatedness, we use a time-sliced version of this corpus for which all data from January 1, 1975 to December 31, 1999 is used as a pre-cutoff dataset. and data from January 1, 2000 to December 31, 2015 (the last date in the corpus) are used as a post-cutoff dataset. For estimating semantic relatedness, and for calculating future relatedness, no time-slicing is required, so we use all data from January 1, 1975 to December 31, 2015.

Co-occurrence Matrix: We create the pre-cutoff, post-cutoff, and full co-occurrence matrices used by all measures except concept embeddings cosine using UMLS::Association version 1.3's CUI Collector tool[1] run over titles and abstracts of the 2015 MetaMapped MEDLINE Baseline, with sentence boundaries ignored. We used a window size of 8 (meaning 8 concepts after a concept are counted) and default values for all other parameters. The result is a co-occurrence matrix for which each row corresponds to the co-occurrences of a single UMLS concept to every other UMLS

[1] `https://metacpan.org/pod/UMLS::Association`

concept (indicated by the column). We apply a minimum co-occurrence threshold of 1 to all matrices.

Indirect association measures: Indirect association measures and LTC are implemented in the UMLS::Association v1.7 package[1], a Perl implementation of association measures. LTC is calculated using the '- -lta' option with 'measure=freq'. The Pearson's chi squared association measure ('measure=x2') is used for the association equation for all indirect association measures.

Concept Embeddings: Concept embeddings rely on co-occurrence information, but not a co-occurrence matrix. Vector representations are created as the training algorithm iterates over the corpus. We use the word2vec-interface package version 0.03[2] with the Continuous bag of words (CBOW) embedding model, a window size of 8, a frequency cutoff of 5, and default settings for all other hyper-parameters. The pre-cutoff corpus is used to create embeddings for time-slicing based methods, and the full corpus is used for estimating direct relatedness.

Time-Sliced Dataset: We develop our time-sliced dataset in a similar manner as Yetisgen-Yildiz and Pratt [116]. We remove all pre-cutoff co-occurrences from the set of post-cutoff co-occurrences. This results in a set of co-occurrences present in the post-cutoff dataset only. We use this as our set of future relationships to estimate a gold standard. For CRGs a weight is added to each future relationship, which is the direct Pearson's chi squared association between the relations start and target term as implemented in UMLS::Association v1.7 with default parameters, and the full (1975-2015) co-occurrence matrix. As described in section 5.2.1 200 start terms of specific semantic types are selected, and all possible future relations are considered.

[2] `https://metacpan.org/pod/Word2vec::Word2vec`

5.4 Results

In this section, we evaluate our newly proposed indirect association measures (LTA, MWA, SBC, and LSA) against the baselines of concept embedding cosine distance (Emb Cos), direct co-occurrence vector cosine distance (Dir Cos), linking term count (LTC), and randomly assigned scores. We evaluate using 3 different evaluation techniques, each technique measures a different aspect of a relatedness measure's performance.

1. Estimating direct semantic relatedness - shows how well a measure estimates direct semantic relatedness using Spearman's rank correlation coefficient against human gold standard datasets.

2. Link prediction - shows the ability of a measure at distinguishing between term pairs that will or will not be related in the future using receiver operating characteristic (ROC) curves and a time-sliced dataset of MEDLINE co-occurrence data.

3. Estimating future relatedness - shows the ability of a measure at ranking term pairs with a high future relatedness before term pairs with a low future relatedness using cumulative relatedness graphs (CRGs) and time-sliced MEDLINE co-occurrence data.

5.4.1 Results: Estimating Direct Semantic Relatedness

Table 20 shows results for each method on each dataset on the task of estimating semantic similarity and relatedness. Each row shows the results for a single ranking method, and each column for a single dataset. The Spearman's Rank Correlation coefficient (ρ) with the number of terms (n) compared in parentheses are shown for

each method on each dataset. The top two rows show the performance of baseline methods of randomly assigning scores and LTC. The middle four rows show results for indirect association scores (LTA, MWA, SBC, LSA), and the bottom two rows show results for vector-based methods (Dir Cos, and Emb Cos). Higher values of ρ are better, and indicate higher rank correlation to the gold standard. Higher values of n indicate more terms are able to be compared. The best performing method for each dataset is shown in bold.

Correlation Coefficients (ρ) and number of samples (n)

Measure	MiniMayo Cod	MiniMayo Phys	UMNSRS Sim	UMNSRS Rel
Random	-0.0300 (29)	-0.1279 (29)	-0.0185 (401)	-0.0113 (430)
LTC	0.5132 (29)	0.5063 (29)	0.2195 (390)	0.2386 (415)
LTA	0.4930 (29)	0.5403 (29)	0.4772 (390)	0.3526 (415)
MWA	0.2902 (29)	0.3231 (29)	0.3617 (390)	0.2606 (415)
SBC	0.6351 (29)	0.5978 (29)	0.5163 (389)	0.5112 (414)
LSA	0.3881 (29)	0.4027 (29)	0.3366 (390)	0.3080 (415)
Dir Cos	0.5946 (29)	0.5165 (29)	0.5315 (390)	0.4015 (415)
Emb Cos	**0.7762** (29)	**0.6942** (29)	**0.7038** (392)	**0.5537** (418)

Table 20. Semantic relatedness results

Emb Cos performs the best for each dataset (MiniMayo Cod, MinMayo Phys, UMNSRS Sim, and UMNSRS Rel), and SBC performs the second best for each dataset except UMNSRS Sim, for which Dir Cos performs better. Emb Cos performs statistically significantly better than Dir Cos on the UMNSRS Sim and UMNSRS Rel datasets, and statistically significantly better than SBC on only the UMNSRS Sim dataset. SBC performs statistically significantly better than direct cosine on only the UMNSRS Rel dataset. The results of other indirect association measures

and LTC are mixed. MWA performs worse than LTA and LSA for all datasets, and worse than LTC for both MiniMayo datasets; it is the worst performing indirect association measure. LTC performs well for the MiniMayo datasets, and poorly for the UMNSRS datasets, indicating that since it is a simplistic method, it may not be able to effectively quantify indirect association for all concepts, and that the concepts in the MiniMayo dataset may be "easy" examples. LTA performs better than LSA for each dataset.

All methods are able to quantify most concepts in all datasets (indicated by n), but notably, SBC can calculate the association for one less concept than other indirect association measures for the UMNSRS Sim and UMNSRS Rel datasets. When concepts share no linking terms, the shared B set is undefined, and association cannot be quantified; for other methods, their modified n_{11} value is 0, which generates an association of 0.

The results for each method on estimating semantic similarity and relatedness for *Disorders* and *Chemicals and Drugs* concept pairs are shown in Table 21. Each row shows the results for a single ranking method, and each column for a single dataset. Only 109/113, and 122/126 concept pairs for the UMNSRS Sim and UMNSRS Rel subsets occur in our corpus. Only 121/122 concept pairs can be computed using the metrics based on a co-occurrence matrix (LTC, LTA, MWA, SBC, LSA, and Dir Cos), because the `C0392071` is removed from the co-occurrence matrix when the threshold of 1 is applied.

Overall, the results for *Disorders* and *Chemicals and Drugs* semantic group pairs are lower than results using the full datasets. This indicates that estimating relatedness between different concept pairs of different semantic groups is a harder problem than estimating relatedness between concept pairs that are synonymous or of the same semantic group. The order of performance of the methods is, however similar

Correlation Coefficients (ρ) and number of samples (n)

Method	UMNSRS Sim	UMNSRS Rel
Random	0.0460 (109)	0.0433 (122)
LTC	0.2480 (109)	0.2190 (121)
LTA	0.1622 (109)	0.3191 (121)
MWA	0.0412 (109)	0.2435 (121)
SBC	0.3639 (109)	0.4146 (121)
LSA	0.1982 (109)	0.2663 (121)
Dir Cos	0.2519 (109)	0.2878 (121)
Emb Cos	**0.5690** (109)	**0.5730** (122)

Table 21. Semantic relatedness results for *Disorders* and *Chemicals and Drugs* semantic group pair subsets. The Spearman's Rank Correlation coefficient (ρ) with the number of terms (n) compared in parentheses are shown for each method on each dataset. The best performing method for each dataset is shown in bold.

to the full dataset. Emb Cos performs the best, and SBC performs the second best for both UMNSRS Sim and Rel subsets. Dir Cos performs the third best, and fourth best for the UMNSRS Sim and Rel datasets respectively. Results are mixed for the other methods, but interestingly, LTC performs third best for the UMNSRS Sim subset, and the worst on the UMNSRS Rel subset. LTA performs third best on the UMNSRS Rel subset, and second worst on the UMNSRS Sim subset. MWA performs very poorly on the UMNSRS Sim subset, but OK on the UMNSRS Rel subset. Emb Cos performs statistically significantly better than SBC on neither dataset, but statistically significantly better than the third best performing measures (Dir Cos and LSA) on both datasets.

5.4.2 Results: Link Prediction

Figure 22 show the ROC curves generated for the highly-cited versus noise, and published versus noise datasets respectively.

In ROC curve analysis, an ideal classifier would produce a curve that goes straight up the y-axis at a value of 0.0 and straight across the x-axis at a value of 1.0, and produce an AUROC of 1.0. A random classifier would produce a straight ROC curve going diagonally from (0,0) to (1,1) and an AUROC of 0.5. Therefore lines closer to this perfect scenario with higher AUROCs are better, and lines closer to this random scenario with lower AUROCs are worse.

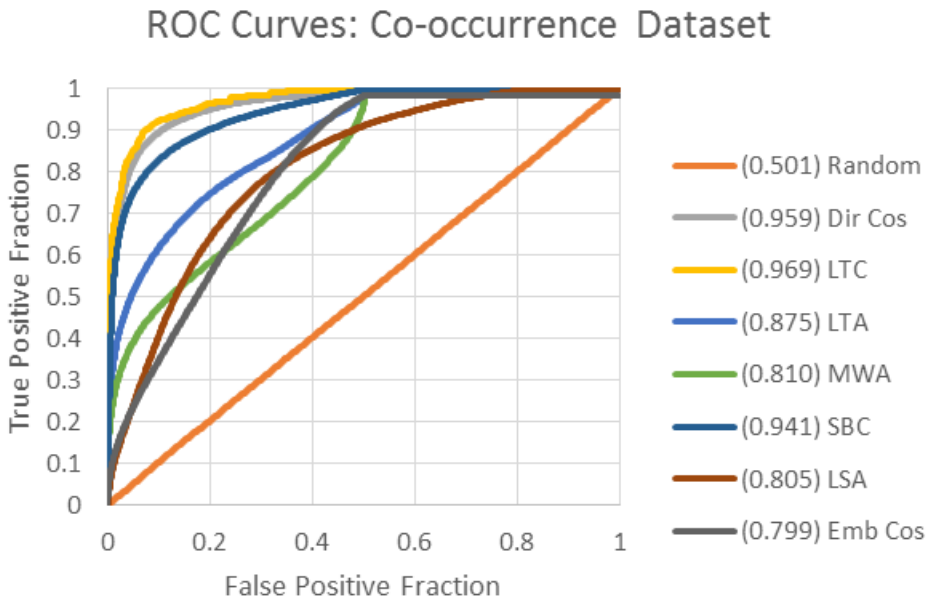

Fig. 22. This ROC curve shows the ability of each ranking method to distinguish between term pairs that co-occur in the future and those that do not using MEDLINE co-occurrences to represent a relationship. Each measure's AUROC is shown in parentheses.

The order of performance from best to worst is LTC, Dir Cos, SBC, LTA, MWA, LSA, and Emb Cos. The ROC curves of LTC, Dir Cos, and SBC are very similar, and

all have AUROC values greater than 0.9; this is excellent performance. LTA performs the next best, with decent performance, followed by ROC curves for MWA, LSA, and Emb Cos. MWA's curve is interesting in that it is not concave down throughout, with the least slope in the middle. This indicates that it has trouble distinguishing between true and false relationships ranked in the middle.

5.4.3 Results: Estimating Future Relatedness

Figure 23 shows the CRGs generated using our time-sliced co-occurrence dataset. The Y-axis shows the cumulative relatedness at the rank indicated by the x-axis. Each colored line shows the CRG of a different method. Not all lines are the same length, and no methods except random reach 100% cumulative relatedness. This is because some methods are able to quantify relatedness between more terms. For example, when randomly assigning scores, we can quantify relatedness between all possible term pairs, but for other measures the terms must share some linking term information in order to quantify relatedness. Most notably, Emb Cos quantifies the relatedness between less terms than all other methods. This is because a frequency cutoff of 5 is used, meaning that embeddings are not created for concepts that occur less than 5 times in the corpus.

The ideal CRG is generated by using the direct future relatedness scores to rank terms. It shows the best possible ranking, and very quickly achieves a cumulative relatedness of 100%. It is much shorter than the other lines, which shows how few actual future relationships in the gold standard. There is a lot of room for improvement for all methods, but Emb Cos, Dir Cos, and SBC perform the best. Emb Cos performs the best at low ranks, but as more terms are ranked, Dir Cos performs the best, followed closely by SBC. LTC, LTA, and MWA all perform similarly, and LSA performs the worst.

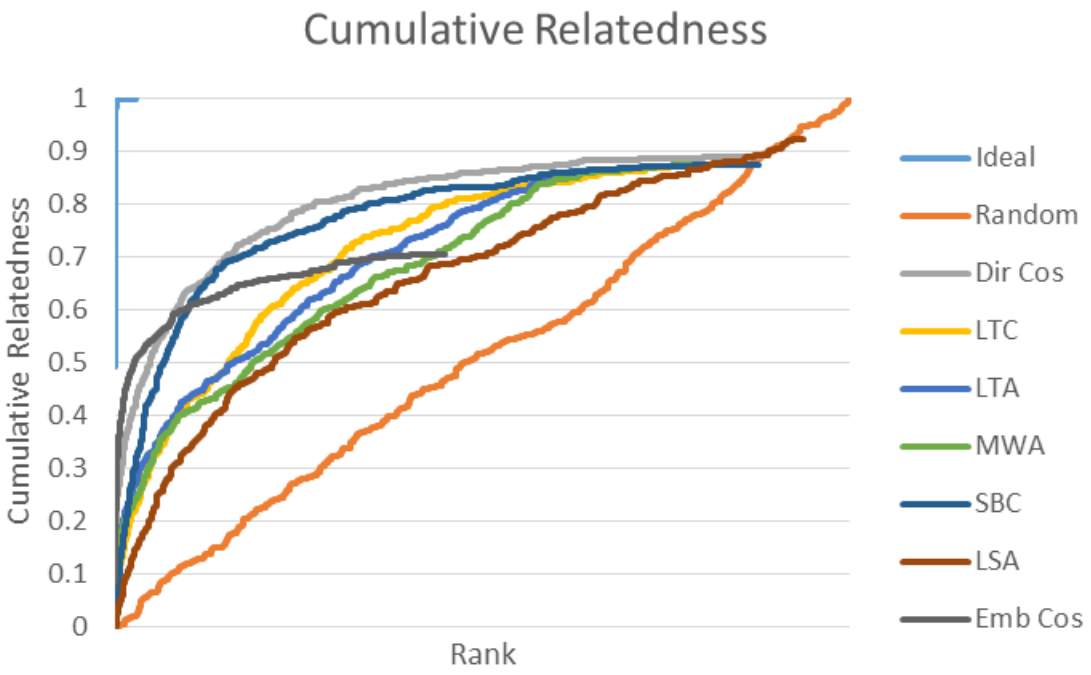

Fig. 23. This cumulative relatedness graph shows the ability of each ranking method to estimate future relatedness. Each line corresponds to a different ranking method. The ideal line is shown for which the direct future relatedness is used to rank.

5.5 Analysis

The results of indirect relatedness measures sometimes varied depending on the evaluation method. This is understandable because the evaluation methods differ in what they evaluate. Estimating direct relatedness is evaluated using Spearman's rank correlation with standard evaluation datasets. Distinguishing between true and false future predictions in a link prediction manner with ROC curves, and estimating future relatedness with CRGs. To better understand each indirect relatedness measure's overall performance, we summarized their results in Table 24. Each row shows the results of a different method and a "grade" of their performance. The grades were assigned in a somewhat subjective manner based on their order of performance and

noticeable groupings and are included to give an overall understanding of performance.

	Dir. Rel. All		Dir. Rel. Subset		ROC Curve		CRGs	**Overall**
	mean ρ	grade	mean ρ	grade	AUROC	grade	grade	
LTC	0.369	!	0.234	✗	0.969	✓	!	!
LTA	0.466	!	0.241	✗	0.875	!	!	✗
MWA	0.309	✗	0.285	✗	0.81	!	!	✗
SBC	0.565	✓	0.389	!	0.941	✓	✓	✓
LSA	0.359	!	0.232	✗	0.805	✗	✗	✗
Dir Cos	0.511	✓	0.269	✗	0.959	✓	✓	✓
Emb Cos	0.682	✓	0.571	✓	0.799	✗	✓	✓

✓ Good
! OK
✗ Bad

Fig. 24. The performance of each ranking method on each evaluation task. An overall grade of good, OK, or bad is assigned to each method to summarize performance.

The first evaluation method, estimating direct relatedness using the full UMNSRS and Mini Mayo datasets (abbreviated "Dir Rel All") shows the mean Spearman's rank correlation averaged across all four datasets, and a grade of their performance. Emb Cos, SBC, and Dir Cos all perform well. LTA, LTC, and LSA perform OK, and MWA performs poorly.

The next evaluation method, estimating direct relatedness of the subsets of UMNSRS Sim and UMNSRS Rel using the *Disorders* and *Chemicals and Drugs* concept pairs (abbreviated "Dir Rel Subset") shows the mean Spearman's rank correlation across both datasets, and a grade of their performance. All methods perform poorly, with the exceptions of SBC and Emb Cos, which perform OK and well respectively. Both Dir Rel All and Dir Rel Subset results are used to determine the performance of measures at estimating direct relatedness, but since the order of performance is similar for both methods, Dir Rel All better summarizes their performance at a glance. Overall, for estimating direct semantic relatedness Emb Cos, SBC, and Dir Cos perform the best, followed by LTA, then LTC and LSA. MWA performs the worst.

The next column shows the link prediction evaluation results with ROC curves.

The AUROC is shown for each method along with a grade. LTC, Dir Cos, and SBC perform the best, followed by LTA and MWA which perform just OK. Lastly, LSA and Emb Cos perform poorly.

The next column shows the performance at estimating future relatedness evaluated with CRGs. Just a grade is shown, since CRGs don't have a single number to quantify performance. Dir Cos, SBC, and Emb Cos perform the best, followed by LTC, LTA, and MWA, which perform OK, and lastly LSA, which performs poorly. The final column shows an overall grade based on the results of each evaluation method. SBC, Dir Cos, and Emb Cos perform the best overall, LTC performs OK, and LTA, MWA, and LSA perform poorly.

The order and quality of performance for estimating direct and indirect semantic relatedness is similar. For both evaluation methods SBC, Dir Cos, and Emb Cos perform the best, indicating these methods are the best for estimating relatedness overall. The order of performance with ROC curves is different from the others, but LSA and MWA perform poorly for all evaluation methods, and SBC performs good or OK for all evaluation methods.

The difference in performance between SBC and LSA is surprising since their methodologies are similar, but indicates that their performance may be sensitive to the selection of the proxy sets for A and C. We believe LSA performs poorly because the B_A and B_C sets are too large and too noisy. SBC uses the shared linking term set, which is much smaller and more relevant to how A and C interact. Interestingly, direct cosine also uses the overlap of co-occurring terms or shared contexts, since only concepts that co-occur with A and C (and therefore are non-zero) contribute to the cosine distance. Filtering, or selecting only the most relevant terms for LSA, SBC, and direct cosine may improve results in the future.

It is surprising the Emb Cos performs well at estimating direct and future re-

latedness, but poorly for link prediction, and that LTC performs the best for link prediction, but just OK for the other tasks. This indicates that different relatedness measures have different strength and weaknesses. Overall, these results indicate that SBC, Dir Cos, and Emb Cos are the best methods for estimating direct and indirect relatedness, where as LTC is the best method for link prediction tasks, so it may be beneficial to combine the results of the measures or use them both in a single application.

Even though the performances of each method varied, having a variety of ranking methods with different theoretical foundations is useful; it allows the best method to be selected for each application. Preliminary collaborative efforts show that researchers often know the types of connections they are looking for, and want to fix both the A and B term sets to determine how some known A interacts with some C term set via the means of a relatively small, known B set (e.g. how a drug affects a class of diseases through means of several metabolites). In this scenario, it is likely that most terms will co-occur with the entire B term set, which means that each evaluated method except MWA would produce uninteresting results. LTC, LTA, and LSA require that the linking terms between A and C are different in order to produce interesting results. SBC and direct cosine produce similarly uninteresting results, since restricting the B set restricts the shared B set making results identical for all A and C terms. Emb Cos takes the cosine between A and C vectors and ignores the B terms entirely, and therefore how A interacts with C through B. So, although MWA performs poorly at estimating future relatedness, MWA is the only method capable of producing interesting rankings in this scenario.

To summarize our findings, we divide our methods into three groups, linking term based association methods (LTC, LTA, MWA), which directly use the linking terms in their calculations. Set based association methods (SBC, LSA), which use

set associations between proxy sets in their calculations. Vector methods (Dir Cos, Emb Cos) which use vector cosine to quantify relatedness. Below, we summarize the differences between evaluated measures and indicate their strengths:

- LTC - linking term based, which is simple to compute and has the best empirical performance for link prediction.
- LTA - linking term based method, is faster to compute than other indirect association measures.
- MWA - linking term based method, which may be the only interesting method when B terms are restricted to a small set.
- SBC - set based association method, which performs well at all tasks making this a good general purpose indirect relatedness measure.
- LSA - set based association method, which performs poorly at most tasks, but uses the largest sets of proxy terms. This gives it the greatest chance of being able .to quantify relatedness and could therefore be useful in domains with small datasets.
- Dir Cos - vector method, which performs well on all tasks (except Dir Rel Subset), making this a simple to compute general purpose method.
- Emb Cos - vector method, good for estimating direct and future relatedness. This is the only method that does not rely on a co-occurrence matrix, which makes it the fastest to compute. This means the vectors can be created from the larger corpora faster than other methods.

5.6 Conclusions

We evaluated our indirect association measures, LTA, MWA, SBC, and LSA, and the baseline measures of concept embeddings cosine (Emb Cos), direct co-occurrence vector cosine (Dir Cos) and linking term count (LTC) on the tasks of estimating direct relatedness using four datasets and two subset, the task of link prediction, and on estimating future relatedness using cumulative relatedness graphs (CRGs). For our evaluation, we developed the evaluation method of CRGs, and a time-sliced co-occurrence dataset which we used with ROC curve analysis. We found that SBC, Dir Cos, and Emb Cos are the best methods for estimating direct and indirect relatedness, and LTC is the best method for link prediction tasks.

The goal of this analysis is to evaluate indirect relatedness measures applied to LBD, and although we showed that our methods perform well for these tasks, it is unclear if these tasks are indicative of performance for LBD. How to best evaluate LBD systems is an open question which we discussed in Section 3.3.2. In Chapter 6 we develop a generalized evaluation framework for LBD, and introduce our proposed evaluation methods and dataset.

CHAPTER 6

LITERATURE BASED DISCOVERY EVALUATION

In this chapter, we address our second critical problem of LBD, the lack of meaningful evaluation methods and datasets. LBD is difficult to evaluate, and evaluation methods are often ad-hoc and system specific, but representative datasets and effective evaluation standards are critical for improving, comparing, and sharing LBD systems and components. In this chapter, we create a standard LBD evaluation dataset, assign evaluation methods to individual LBD components, and evaluate indirect relatedness measures (introduced in Chapter 5) for the term filtering and term ranking steps of LBD. Specific contributions of this chapter are:

1. Development of a standard evaluation dataset
2. Assignment of evaluation methods for LBD components in isolation and in combination
3. Evaluation of indirect relatedness measures for term filtering and term rankings steps of LBD

We begin this chapter by defining desirable evaluation dataset characteristics. Next, we compare our co-occurrence-based time-slicing dataset (introduced in Chapter 5) to a SemMedDB-based time-slicing dataset developed by Sybrandt, et al. [87]. We then present a hybrid time-slicing dataset that addresses other datasets' weaknesses. We illustrate the datasets' differences with ROC Curve evaluation of indirect relatedness measures. Next, we assign evaluation methods to each of the LBD components identified in Chapter 3. These evaluation methods include precision and recall

(PR) curves, precision at K graphs, and user studies. With our hybrid dataset and evaluation methods, we evaluate indirect relatedness measures for term filtering and term ranking in LBD, and make conclusions.

6.1 Dataset Comparison

Time-slicing [116] and link prediction [85, 87] type evaluation techniques are common LBD evaluation methods, but they require a gold standard of future discoveries for evaluation. There is no widely accepted gold standard dataset, and no consensus on the best way to generate a gold standard. An ideal gold standard would contain all possible future discoveries, and no currently known discoveries. This is an impossibility, so instead, the gold standard is estimated with several assumptions and simplifications. Time-slicing is a way to estimate future discoveries. In time-slicing evaluation, a dataset is divided into pre- and post-cutoff segments. The pre-cutoff segment represents all known knowledge, and the post-cutoff segment represents all future knowledge. Future discoveries are simplified as a future relationship, and a future relationship is simplified as a term pair. Term pairs that occur in the post-cutoff dataset and not in the pre-cutoff dataset are considered new discoveries. With these simplifications, we can evaluate an LBD system's ability to predict future discoveries by its ability to predict whether a term pair that is absent from the pre-cutoff set will occur in the post cutoff set. The question remains though, how to best generate the pre- and post-cutoff sets of term pairs?

There are two general methods of doing this, (1) using co-occurrences or (2) using extracted relationships. Both methods have advantages and disadvantages, and in this section, we define several characteristics desirable for time-slicing datasets, and compare a co-occurrence-based to a relationship-based dataset. Time-slicing is used primarily to evaluate hypothesis generation and its sub-steps (Figure 29), and to

illustrate differences between the datasets, we evaluate our indirect relatedness measures of linking term association (LTA), minimum weight association (MWA), shared B to C association (SBC), and linking set association (LSA) against the baselines of concept embeddings cosine (Emb Cos), direct co-occurrence vector cosine (Dir Cos) and linking term count (LTC) for link prediction using each dataset and ROC curves (as described in Section 5.2.1). Additionally, a good time-slicing dataset should have the following characteristics:

1. Universal - be able to evaluate each component of the LBD hypothesis generation process in combination or separately.

2. Representative - be representative of real-world LBD data.

3. High Precision - contain minimal false discoveries.

4. High Recall - contain maximum true discoveries.

5. Generalizable - be usable by (nearly) all LBD systems, so systems can be compared.

In the next two sections, we compare the co-occurrence based dataset we developed in Chapter 5, an extracted-relation based dataset that uses SemMedDB predications developed by Sybrandt, et al. [87], and a hybrid dataset we developed. Table 22 summarizes which of the criteria these datasets meet. No mark means that criteria is not met, an "X" indicates the dataset meets that criteria, and a "/" indicates a dataset almost meets that criteria. All presented datasets use UMLS concepts instead of terms, however a mapping between terms and concepts exists, so these datasets can be used regardless of whether a system uses concepts or terms.

Time-Slicing Dataset Criteria

	Universal	Representative	High Precision	High Recall	Generalizable
Our Co-occurrence	X	/		X	X
Sybrandt SemMedDB			/		/
Our Hybrid	X	X	/	/	X

Table 22. Evaluation dataset criteria and which ones they meet. No mark means that criteria is not met, an "X" indicates the dataset meets that criteria, and a "/" indicates a dataset almost meets that criteria.

6.1.1 Our Co-occurrence Dataset

Description: We developed a co-occurrence based time slicing dataset using the procedure outlined by Yetisgen-Yilidz and Pratt [116]. We describe the dataset's construction in detail in Section 5.3, but briefly, the dataset consists of co-occurring term pairs collected from MEDLINE. We select 200 start terms by randomly selecting 50 terms from each of the semantic types of: Clinical Drug (T200, clnd), Pharmacologic Substance (T121, phsu), Disease or Syndrome (T047, dsyn), Sign or Symptom (T184, sosy). A set of target terms is defined as all terms in the vocabulary, and start-target term pairs are generated for all possible start-target term pairs. Term pairs that exist in the pre-cutoff segment are removed, and the result is a set of all possible start-target term pairs for which no direct co-occurrence exists in the pre-cutoff dataset.

Analysis: This dataset is *universal*, it can evaluate all LBD hypothesis generation components. Specifically, since all possible target terms are used to generate term pairs, it can evaluate the term generation step. This dataset is *somewhat representative* of LBD data, since using all terms in the vocabulary mimics how LBD is performed, the distribution of true and false samples should be representative of real LBD data. It, however, relies of randomly selecting 200 start terms, and there is no guarantee these terms are representative samples. The dataset has *low precision* since using co-occurrence information over-generates relationships [75]. This dataset

has *high recall*, since using co-occurrences will capture nearly all true relationships in the data. This dataset is *generalizable* since it uses concept-concept pairs, and can therefore be used by nearly all LBD systems.

Results: We previously evaluated indirect relatedness measures using this dataset in Chapter 5 for estimating future relatedness, and we show the results again here in Figure 25. Each colored line corresponds to a different indirect relatedness measure. The legend shows the area under the ROC curve (AUROC) in parentheses. The best performing measures are LTC, Dir Cos, and SBC. LTA performs OK, and MWA, LSA, and Emb Cos which perform the worst.

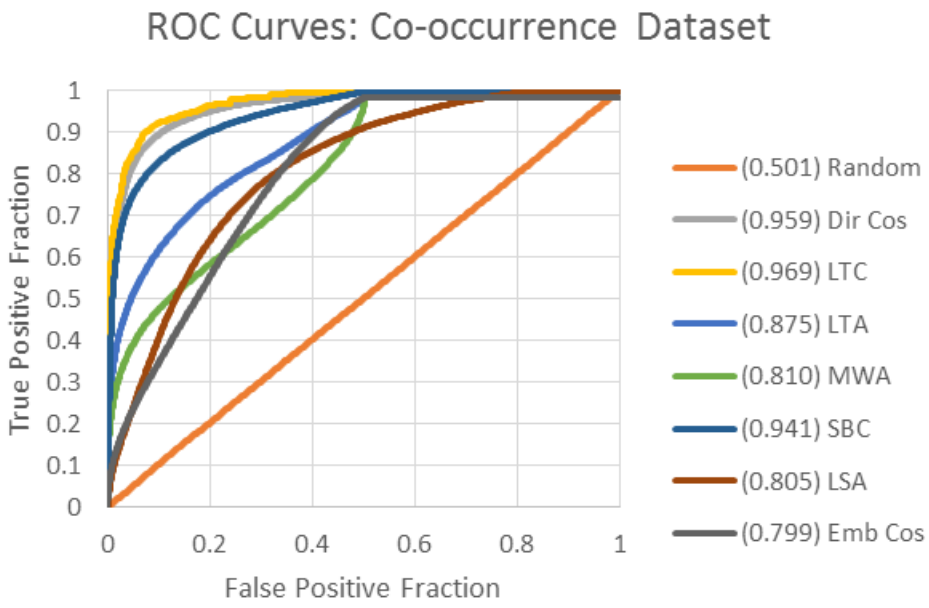

Fig. 25. This ROC curve shows the ability of each ranking method to distinguish between term pairs that co-occur in the future and those that do not using MEDLINE co-occurrences to represent a relationship. The AUROC of each method is shown in parentheses.

6.1.2 Sybrandt Dataset

Description: Sybrandt, et al. create two time-slicing datasets of SemMedDB predications (available online[1]). These datasets are the *published* dataset, which consists of a non-exhaustive (it is unclear how the predications were selected) set of predications that are present in the post-cutoff segment and absent from the pre-cutoff segment, and the *highly cited* dataset, which consists of predications selected from the *published* dataset which occur in papers that are cited at least 100 times. They use a cutoff date of January 1, 2010, and create a gold standard of 4319 published predications, and 1448 highly-cited predications. These predications are represented as start-target term pairs. To add false pairs to their dataset, they create a *noise* set by randomly generating term pairs that do not occur in either the pre- or post-cutoff segments. Pairs are randomly selected from the noise set to create datasets with a balanced number of true and false samples. The end result is their published dataset of 4319 true and 4319 false term pairs, and their highly-cited dataset contains 1448 true and 1448 false term pairs.

Analysis: Since the false term pairs are randomly generated and selected, this dataset is *not universal*, it cannot evaluate all components of LBD. Evaluating term generation requires that for each start term, all possible start-target term pairs be defined as either true, false, or known. If we input a start term into the term generation step, we may output a start-target term pair that is not defined in our dataset, and we therefore cannot determine a truth value for it. Since this dataset cannot evaluate term generation, it therefore cannot evaluate the hypothesis generation step as a whole. This dataset can, however evaluate the term filtering and term ranking steps.

[1] `https://github.com/JSybrandt/Moliere_Validation_Data`

This dataset is *not representative* of LBD data. There are two reasons for this: (1) Sybrandt, et al. artificially produce a balanced dataset, but since most term pairs are not future discoveries, LBD evaluation datasets should have a high class imbalance; (2) using co-occurrence counts collected from the pre-cutoff segment of MEDLINE, we found a distinct difference in term occurrence rates for the term in true pairs versus the term in false pairs. This difference is with respect to the number of terms the terms occur with, and the count of occurrences of the terms. Table 24 summarizes our findings. It shows the average number of co-occurring terms, and average number of occurrences for the start (A) and target (C) terms in the *highly-cited*, *published*, and *noise* datasets.

Difference in Co-occurrences Rates between Terms in each Dataset

Term Set	mean co-occurring terms	mean occurrences
highly-cited A	13,587	987,086
highly-cited C	9,065	607,984
published A	10,312	627,894
published C	7,109	398,202
noise A	2,152	82,555
noise C	1,770	76,213

Table 23. ROC dataset co-occurrence means. Average number of co-occurring terms and average occurrence count for each A and C term in the highly-cited, published, and noise datasets.

Table 23 shows that on average, both the start (A) and target (C) terms in the true pair sets (*highly-cited* and *published*) occur much more frequently and co-occur with many more terms than the terms in the false (*noise*) dataset. This difference in occurrence rates is understandable, since highly cited term pairs may come from more

popular research areas than just any published term pair, and noise term pairs that never co-occur together likely consist of rarely used terms. This difference though, creates a bias in the dataset and is not representative of real-world LBD data.

Sybrandt, et al.'s dataset is, however *somewhat precise*, since using relationships rather than co-occurrences greatly increases the precision of the extracted relationships. SemRep (which is used to generate SemMedDB) has precision rates between 73% and 96% [15] depending on the relationship type. This increase in precision from using SemMeDB also means a decreased recall; SemRep recall rates are between 55-70% depending on the relationship type. Furthermore, the set of *published* pairs is not exhaustive, and it is not clear how the discoveries were sampled, meaning there is *low recall* overall. Lastly, this dataset is only *somewhat generalizable*. It uses concept-concept pairs so it can be used by nearly all LBD systems, but it relies solely on SemMedDB to create the pre- and post cutoff segments. This is problematic for systems that use co-occurrence information from MEDLINE instead of, or in addition to SemMedDB information. Although the true concept pairs (*highly-cited* and *published* sets) may be absent from the pre-cutoff segment of SemMedDB, they may co-occur together in a pre-cutoff version of MEDLINE. We found that over half of these true pairs directly co-occur in the pre-cutoff portion of MEDLINE. This means we are treating known knowledge as true discoveries, which makes the informativeness of any results (including our own) questionable.

Results: We evaluated indirect relatedness measures using the *highly-cited* versus *noise* and *published* versus *noise* datasets. To generate the results, we use the parameters described in Section 6.4. Figures 26 and 27 show the ROC curves generated for the *highly-cited* versus *noise*, and *published* versus *noise* datasets respectively.

Results are very similar for both the *highly-cited* versus *noise* and *published* versus *noise* datasets. The order of performance from best to worst for both datasets

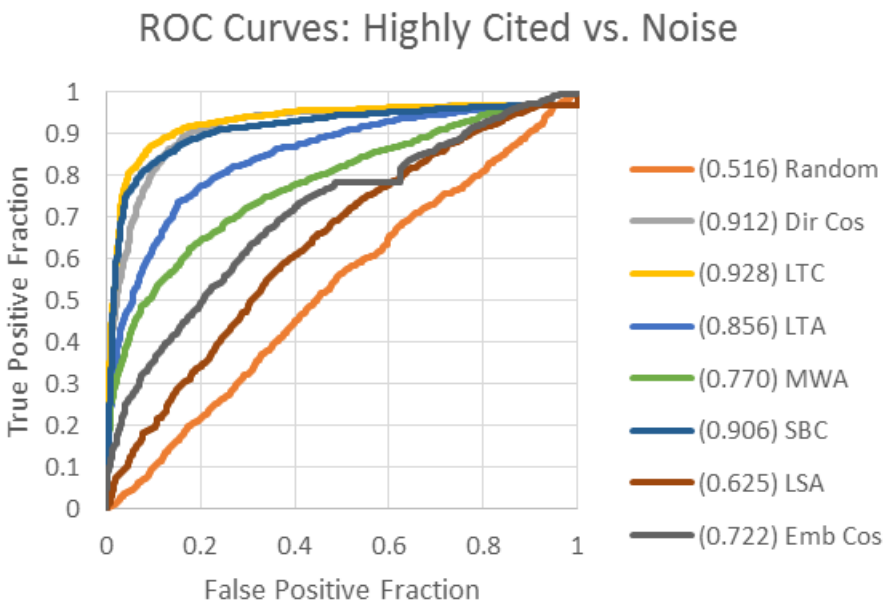

Fig. 26. This ROC curve shows the ability of each method to distinguish between *highly-cited* term pairs and *noise*. *Highly-cited* pairs appear in papers with over 100 citations after the cutoff date. Each measure's AUROC is shown in parentheses.

is LTC, Dir Cos, SBC, LTA, MWA, Emb Cos, and LSA. The ROC curves of LTC, SBC, and Dir Cos are very similar, and all have AUROC values greater than 0.9; this is excellent performance. LTA performs the next best, with good performance, followed by similar ROC curves for MWA and embeddings cosine. LSA performs similar to random, indicating very poor performance. In both datasets, there is a straight line portion of the ROC curve for Emb Cos that is caused by having many tied terms. In our ROC curve implementation (see Section 5.2.1), we penalize tied terms to produce the worst case scenario results, such that any false terms are ranked higher than true terms.

Interestingly, the results are very similar for our co-occurrence based dataset and for Sybrandt, et al.'s dataset. The order of performance is the same for all but the worst two performing methods, Emb Cos and LSA, which perform the worst on our

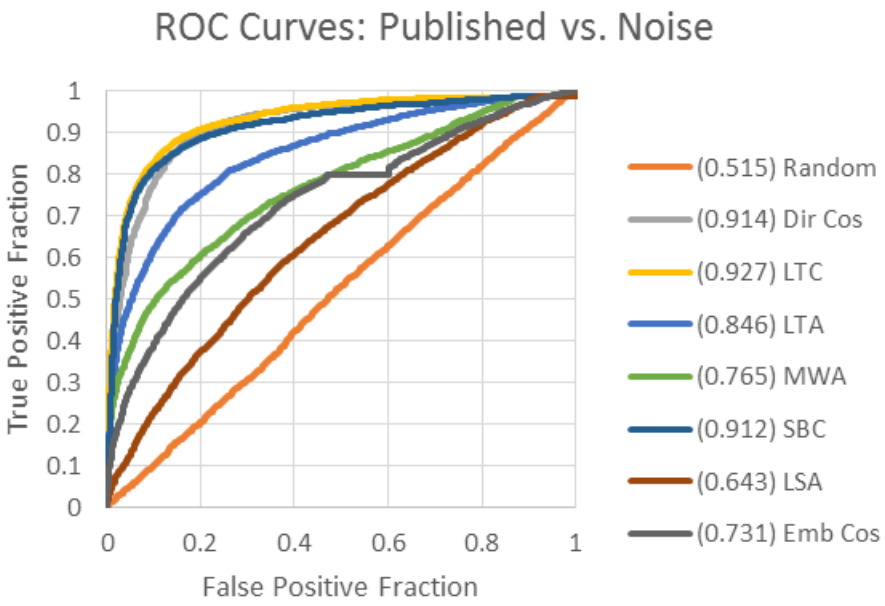

Fig. 27. This ROC curve shows the ability of each method to distinguish between *published* term pairs and *noise*. *Published* pairs appear in at least one paper after the cutoff date. Each measure's AUROC is shown in parentheses.

co-occurrence dataset and Sybrandt, et al.'s datasets respectively. Additionally, the groupings of performance are similar. LTC, Dir Cos, and SBC all clearly perform better than the other methods, and LTA performs somewhere in between. MWA, Emb Cos, LSA remain the worst performing metrics. The similarity in results is surprising, since the datasets are quite different from each other. The co-occurrence based dataset is much larger, and likely contains many false positives. Sybrandt, et al.'s dataset is smaller and more precise, but is not representative of actual LBD data. In the next section, we present our hybrid dataset which addresses problems with these datasets, and we evaluate the performance of indirect relatedness using it.

6.2 Our Hybrid Dataset

Description: In this section we describe our hybrid evaluation dataset (available online[2]). This dataset addresses the problems of our co-occurrence based dataset and of Sybrandt, et al.'s relationship-based dataset by using both relationship (SemMedDB predication) and MEDLINE co-occurrence data. Specifically, we address problems with:

1. Representative - we use all possible term pairs in a restricted vocabulary to more accurately model the distribution of actual LBD data. Specifically, this addresses problems caused by randomly selecting start terms (as is the case with our co-occurrence dataset), and sampling true term and randomly generating false term pairs (as is the case with Sybrandt, et al.'s dataset).

2. Universal - since we use all possible term pairs in this restricted vocabulary we can evaluate all components of the hypothesis generation process, including the term generation step.

3. High Precision - we use SemMedDB data rather than co-occurrence data to generate a set of future discoveries. This has increased precision over co-occurrence data.

4. Generalizable - our set of known knowledge has high recall since we generate it using both SemMedDB and co-occurrence data. This means that our dataset can be nearly all systems regardless of relation extraction method, since known term pairs will not occur as future discoveries (which creates a bias).

We represent current (known) and future knowledge as start-target concept pairs

[2] `https://github.com/henryst57/LBD_Evaluation/releases/tag/v0.01`

in pre- and post-cutoff segments. Known concept pairs are generated from both co-occurrence information in a pre-cutoff versions of MEDLINE and from predications in SemMedDB. Future concept pairs are generated using post-cutoff SemMedDB predications only. The set of true future discoveries is created by removing all pre-cutoff pairs from the post-cutoff pairs. We create false concept pairs as all possible concept pairs that are not in the known or true future datasets. Heuristics are used to reduce the vocabulary, which reduces the size of true and false pairs to a manageable number. The heuristics are semantic type and relationship type filters, restriction to concepts pairs of different semantic types, and co-occurrence based thresholds. Details are explained in Section 6.2.1.

Analysis: Since our hybrid dataset uses all possible term pairs in a restricted vocabulary, it is *universal*, it cannot evaluate all components of the LBD hypothesis generation process. Using all possible term pairs in a restricted vocabulary also means that our hybrid dataset is *representative*; we do not select a small, possibly biased sample of start terms, and we do not generate random, possibly biased false pairs. Unfortunately, since using all term pairs creates a dataset that is too large to be practical, we must apply co-occurrence thresholds to reduce its size which could potentially introduce a bias. Our hybrid dataset uses SemMedDB predications as a gold standard, so it has *moderately high precision* (73% and 96%) and *acceptable recall*(55-70%). Since we use concept pairs to represent relationships it can be used by nearly all LBD systems, and since we use MEDLINE and SemMedDB data for our pre-cutoff segment, it can be used with LBD systems that use either SemMedDB or MEDLINE data.

Results: Here, we evaluate indirect relatedness measures using our hybrid dataset. To generate the results, we use the parameters described in Section 6.4. Figure 28 shows the results of each indirect relatedness measure on our hybrid dataset.

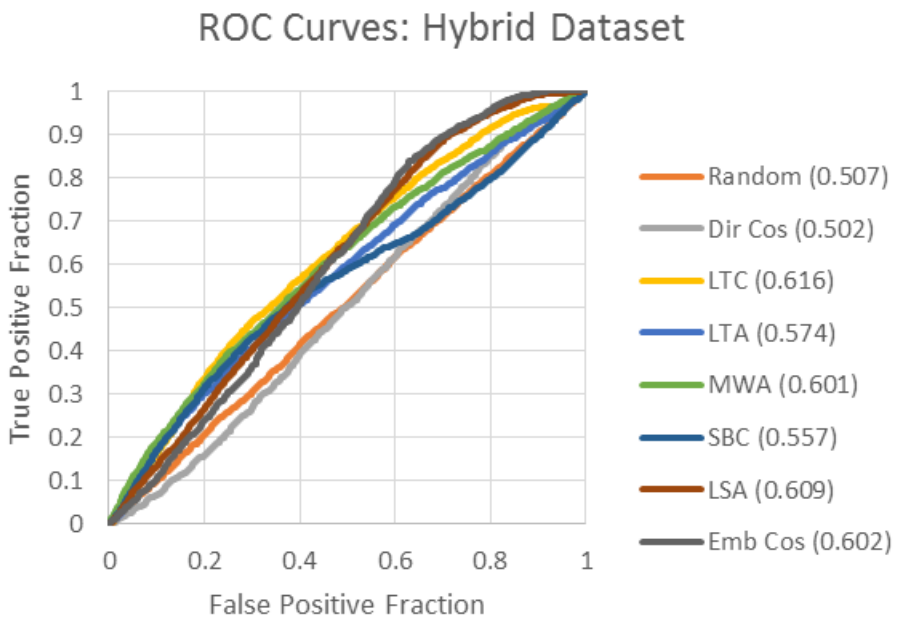

Fig. 28. This ROC curve shows the performance of indirect relatedness measures on our hybrid dataset. Each measure's AUROC is shown in parentheses.

The results using this dataset are much different than the results of the other datasets. All indirect relatedness methods perform poorly, but most perform better than random. These poor results may seem discouraging or confounding, but we believe that our hybrid dataset better models the difficulty of LBD, and that the good performance of the measures on other datasets is caused by biases in those datasets. Our co-occurrence dataset, and Sybrandt, et al.'s dataset introduce a similar bias in two different ways. In our co-occurrence dataset, we use co-occurrence information to construct a gold standard which produces a true set with unacceptably low precision rates. The set of true discoveries is more likely to contain frequently occurring terms, because those terms are used more used often, and are therefore more likely to co-occur with something new in the future. Sybrandt, et al. construct a dataset with a randomly created false discovery set. The majority of terms in a vocabulary occur very infrequently (due to the Zipfian nature of language), so by randomly choosing

terms, they are more likely to select infrequently occurring terms. These terms are therefore more likely to be a part of false discovery (see Table 23). Therefore, for both of these datasets terms that occur frequently are more likely to be a part of discoveries labeled as true. This is problematic, and we addressed it in creating our hybrid dataset. We create true discoveries with high precision (in the same manner as Sybrandt, et al.) by using SemMedDB predications, and we create a comprehensive set of false discoveries (in the same manner as our co-occurrence dataset) by using all terms in the vocabulary. This prevents occurrence rate biases in the both the true and false discovery sets, and better models real-world LBD data.

The order of performance of the indirect relatedness measures on our hybrid dataset (based on AUROC scores) from best to worst is LTC, LSA, Emb Cos, MWA, LTA, SBC, and Dir Cos. LTC performs the best for all datasets, but Dir Cos, and SBC which, for other datasets, were the second and third best performing measures perform the worst with this dataset. Dir Cos is the only method that performs noticeably worse than other methods; it performs no better than random.

In creating the results shown in Figure 28, we observed an interesting phenomenon; results improve if the order of rankings are reversed for LTA, MWA, LSA, Dir Cos, and Emb Cos. For other datasets, ranking is performed in descending order (a pair with a high relatedness score ranks higher than one with a low score), but for these measures ranking in ascending order improves performance on this dataset. Therefore, in Figure 28 we report LTA, MWA, LSA, Dir Cos, and Emb Cos ranked in ascending order, and LTC and SBC in descending order. Reversing these orders results in performance worse than random for all measures except for Dir Cos, which performs on par with random regardless of ranking order.

Ranking in ascending order is counter-intuitive, since we hypothesized that the degree to which two terms are indirectly related indicates the likelihood they are a

future discovery. These results indicate the opposite is true; that terms that are less related are more likely to be future discovery. This is an important finding, and supports the idea that likely relationships are uninteresting, too obvious, or produce only incremental discoveries, and that surprising, unlikely relationships constitute meaningful discoveries and scientific progress [61]. This idea of surprising relations is incorporated into several existing LBD systems [70, 91, 90, 92, 71]. Estimating relatedness is still important though, since it can be reversed (as we do here) to determine interestingness, or a system can be built using multiple filters, where a user can toggle between the types of relationships they are searching for.

6.2.1 Construction Details

In this section, we describe details of the construction of our hybrid dataset. Since LBD is often used to find new treatments (chemicals and drugs) for diseases or disorders, we restrict our dataset to treatment-disease, and disease-treatment pairs which we believe are of high interest to the LBD community. Additionally, since order of co-occurrence or order of subject-object in a relationship does not matter, we do not consider term order during construction. The dataset is constructed in several steps: First, relevant SemMedDB relationship pairs are generated. Next, relevant MEDLINE co-occurrence pairs are generated. Third, time-slicing is performed, and the pairs are divided into pre- and post-cutoff segments. Fourth, the time-sliced co-occurrence and relationship pairs are merged to create sets of known pairs and true future discoveries. Fifth, false concept pairs are generated, and last co-occurrence thresholds are applied to reduce the dataset to a manageable size.

SemMedDB Pair Generation: SemMedDB pairs are generated by first extracting all predications from SemMedDB version 31_R processed up to June 30, 2018. We restrict the subjects and objects of the predications such that they are *treatments*

(semantic types 'Clinical Drug' ('clnd') or 'Pharmacologic Substance' ('phsu')) or *problems* (semantic types 'disease or syndrome'('dsyn') or 'sign or symptom' ('sosy')). Concepts that are too general are also removed by choosing only *novel* concepts, indicated by the subject_novelty and object_novelty flags in SemMedDB. The result is vocabulary of *novel treatment* and *problem* concepts. Next, we restrict the relationships to predication types of: isa, location_of, part_of, uses, causes, process_of, treats, diagnoses, associated_with, coexists_with, method_of, affects, interacts_with, occurs_in, precedes, complicates, prevents, administered_to, disrupts, manifestation_of, compared_with, predisposes, augments, higher_than, inhibits, lower_than, same_as, stimulates, converts_to, than_as. We treat the remaining predications as term pairs in a set, and remove all but *problem-treatment* or *treatment-problem* pairs, causing the set of predication types to narrow further.

MEDLINE Pair Generation: Co-occurrence term pairs are generated using co-occurrence information from MEDLINE. We run UMLS::Association version 1.3's CUI Collector tool[3] over titles and abstracts of the 2015 MetaMapped MEDLINE Baseline, with sentence boundaries ignored, and use a window size of 8 and default values for all other parameters.

Time-Slicing: We divide both the term pairs extracted from SemMedDB and from MEDLINE into pre- and post-cutoff segments using a cutoff date of January 1, 2010. We use the post-cutoff term pairs extracted from SemMedDB predications as our future term pair set. We use the pre-cutoff term pairs extracted from SemMedDB predications and from MEDLINE co-occurrences as our known term pair set. Oddly, when combining the term pairs from SemMedDB and MEDLINE, we found 508 concepts which occur in SemMedDB, but not in MEDLINE. Many of these concepts are

[3] `https://metacpan.org/release/UMLS-Association`

synonyms of those that occur in MEDLINE, or are very general terms. It is unclear why this difference in vocabulary exists, but we suspect it is due to differences in MetaMap versions or UMLS releases used to process the data. Regardless of the reason, this is a small number of concepts, so we removed them from the vocabulary, such that any pairs containing them are removed from the dataset. Next, we ensure the order of term pairs is ignored by adding $B - A$ term pairs for all $A - B$ term pairs in a set. We do this for both the known and future term pair sets. Next, we create our preliminary silver standard set of future discoveries by removing all known term pairs from the future term pair set, and last we limit the vocabulary of the silver standard to that of the pre-cutoff dataset, since we cannot predict unknown concepts.

False Pair Generation: At this point we have a preliminary set of silver standard future discoveries represented as start-target term pairs extracted from SemMedDB predications. The vocabulary of our silver standard is limited to pre-cutoff terms of specific semantic types. To evaluate LBD we must also have a set of false pairs, which we generate as all possible term pair combinations within the silver standard vocabulary. Next, we remove all known and future pairs from this false set, and as with the silver standard pairs restrict them to problem-treatment or treatment-problem term pairs. We add these false term pairs to the silver standard dataset to result in a set of labeled, true and false term pairs. These term pairs are all problem-treatment, treatment-problem term pairs that do not occur in the known term pair set.

Thresholding: At this point, the silver standard dataset is impractically large. Many terms in the vocabulary co-occur with just a few other terms, therefore we reduce the size of vocabulary by removing terms that occur with only small number of subjects or objects. In effect, this removes terms that have few discoveries associated with them. Using a threshold, t, we remove terms that occur with $\leq t$ unique sub-

jects and $\leq t$ unique objects. This is an iterative process, since removing a concept as a subject can affect the number of objects other terms co-occur with. Terms are removed from the vocabulary until the entire silver standard consists of terms with greater than t subjects and t objects associated with them as new discoveries. Limiting vocabulary in this manner greatly decreases the number of concept pairs that are in our silver standard dataset, and at $t = 7$, no terms remain. We use the smallest set ($t = 6$) as our silver standard. Applying this threshold skews the class distribution of true and false samples, but we believe this is acceptable, since the class distribution remains highly imbalanced. Table 24 shows the vocabulary size and class balance at each threshold.

Hybrid Dataset Statistics at with Different Thresholds (t)

Threshold	Vocab Size	True Pairs	False Pairs	True/False Ratio	Mean True Pairs per Term
$t = 0$	6,488	29,878	42,064,266	0.0007	4.61
$t = 1$	3,778	24,662	14,248,622	0.0017	6.53
$t = 2$	2,436	19,506	5,914,590	0.0033	8.01
$t = 3$	1,594	14,726	2,526,110	0.0058	9.24
$t = 4$	960	9,860	695,281	0.0142	10.27
$t = 5$	502	5,606	171,494	0.0327	11.16
$t = 6$	204	2,306	24,664	0.0935	11.30
$t = 7$	0	0	0	0	0

Table 24. Hybrid dataset statistics at different thresholds. A threshold of 6 is used as our final hybrid dataset.

6.3 Literature Based Discovery Evaluation

In this section, we present standard evaluation methods for LBD, which along with our hybrid dataset creates meaningful LBD evaluation methods and datasets. We first describe the evaluation methods, then describe how they are used to evaluate different LBD components.

6.3.1 Evaluation Methods

The LBD components presented in Chapter 3 and shown in Figure 29 have different inputs, outputs, and goals. Evaluating different components in isolation and in combination allows for a more meaningful understanding of the evaluation results, comparison between other systems, and sharing components between them. We propose using three evaluation methods, which along with our hybrid dataset allow all the components of LBD to be evaluated. The evaluation methods are:

1. Precision and Recall (PR) Curves - which show the trade-off between precision and recall.

2. Precision at K graphs - which shows the precision at each rank (K) averaged over all start terms.

3. User-studies - which show how a system is used, users thoughts on a system, and how easily results are interpreted.

In Chapter 3, we compared existing evaluation methods. Their strengths and weaknesses, along with those of PR curves and precision at K graphs are summarized in Table 25.

Evaluation Methods

	Automated	Replicable	Quantitative	Informative	Modular
Discovery Replication	X	X	X		
User Studies				X	X
New Discovery Proposal					
Time-Slicing	X	X	X	X	X
ROC Curves	X	X	X	X	X
Established MI Graphs	X	X	X	X	X
PR Curves	X	X	X	X	X
Precision at K Graphs	X	X	X	X	X

Table 25. Evaluation methods and which of the ideal evaluation criteria they meet as described in Chapter 3

Precision and Recall (PR) Curves: Although most previous work [85, 89, 87] uses ROC curves as an evaluation method for LBD, PR curves have also been used [62]. PR curves and ROC curves show similar information. An ROC curve shows the true and false positive fractions on each axis, where as a PR curve shows the precision and recall on each axis. Recall and true positive fraction are different terms for the same measurement. They measure the percentage of total true samples identified. Precision and false positive fraction differ. Precision measures the percentage of samples identified as true that are actually true. False positive fraction measures the percentage of total samples identified as false that are truly false. As with ROC curves, we can report the area under the curve (AUC) as a single number to quantify performance.

Since PR curves are closely related to ROC curves, they meet the same evaluation method criteria for the same reasons as ROC curves (summarized in Table 25). However, PR curves have been show to be more informative for tasks with a severe class imbalance [99], such as LBD. since our evaluation dataset has a true/false ratio of 0.0935, we recommend using PR curves for evaluation.

Precision at K: Yetisgen-Yildiz and Pratt [62] use precision at K graphs for the top 100 ranked terms in evaluating their LBD system. Precision at K graphs are created by averaging the precision at each rank (K) over all start terms. In target term ranking, we rank a list of start-target term pairs for a single start term. The output is a single set of ranked start-target term pairs. Using this set of ranked pairs, we can calculate the precision at each rank (K) and plot it to create a precision at K graph. In our evaluation we create a precision at K graphs averaged over all start terms as an evaluation method. An ideal system would have perfect precision until there are no remaining true term pairs. We use precision at K graphs for all ranked terms in our LBD evaluation. We calculate the mean average precisions at each K

(average precision at K) to quantify performance with a single value.

Precision at K graphs are a time-slicing technique, and therefore meet the criteria outline in Chapter 3. They are *automated*, *replicable*, and they are *quantifiable* since the average precision at K can be reported as a single number to quantify performance. They are *informative* since we can see how precision changes as rank increases. Additionally, with the inclusion of our hybrid dataset, they are *modular*, since they can be used to evaluate the term ranking step, or the hypothesis generation step of LBD.

Here, we decide to use precision at K graphs rather than established mutual information (MI) graphs (Section 3.3.2.6) because established MI graphs only evaluate directly co-occur terms and their suitability for LBD is therefore questionable. We do not use cumulative relatedness graphs (Section 5.2.1), since our results (Section 6.2) indicate that relatedness may not be a good predictor of future discoveries. We use precision at K graphs in addition to PR curves, because PR curves are designed for binary classifiers and do not explicitly take rank into account.

As we did with our ROC curve construction (Section 5.2.1), when creating both our PR curves and precision at K graphs, we penalize tied terms. That is, when a tie occurs, we create the worst case scenario by ranking false terms higher than true terms. This rewards methods that produce few ties.

User Studies: Although user-studies may not meet all of our ideal evaluation method criteria, they are still a necessary evaluation method. LBD systems are designed to be aids to human discovery and the human component is inescapable, particularly for evaluating how the effectiveness of user interfaces and how results are displayed (a focus of Chapter 7).

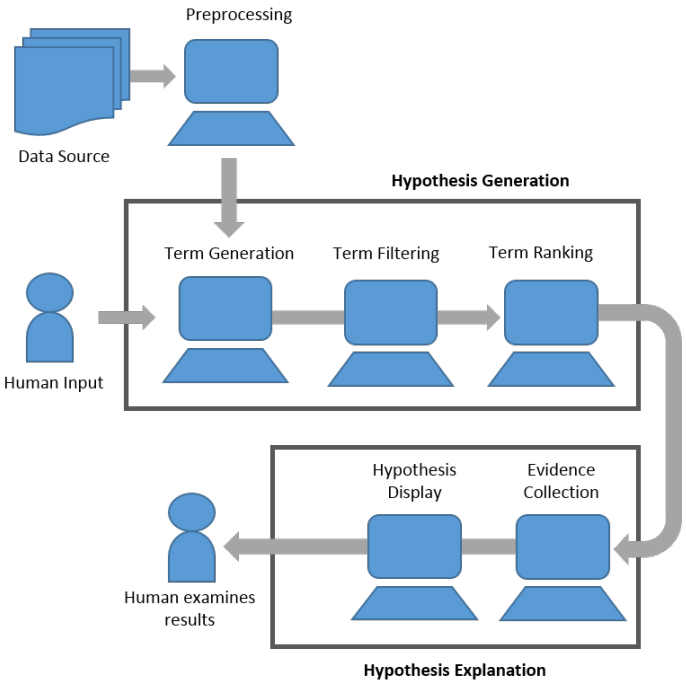

Fig. 29. The LBD process as a series of generalized steps. First data is preprocessed, then, hypothesis generation is performed, in which target terms are generated, filtered and ranked. Lastly, evidence to support the discovery is collected and the hypotheses and their evidence is displayed to the user.

6.3.2 Literature Based Discovery Component Evaluation

In the Chapter 3, we described a generalized framework for LBD. This framework is summarized in Figure 29. It consists of three major steps, and six sub-steps. First data is preprocessed, and along with a starting term(s) input into a hypothesis generation step, where hypotheses in the form of term-term pairs are generated, filtered, and ranked. Next, an explanation for the remaining hypotheses are created via evidence collection and displayed to the user who analyzes the results. In this section, we describe evaluation procedures for each of these steps and sub-steps. Our work focuses on term filtering and ranking, so a more in depth explanation for those components is given in the next subsections.

Preprocessing: Preprocessing techniques are generally not unique to LBD, and standard evaluation datasets and techniques often exist. For instance, preprocessing may consist of named entity recognition or relation extraction. These can be evaluated using precision, recall, or F-measure on tasks such as the 2014 i2b2 De-identification and Heart Disease Risk Factors Challenge dataset [118] or 2010 i2b2 Relations Challenge dataset [119], the ChemDNER dataset [120], or BioCreative V CDR dataset [121]. Extrinsically evaluating the effect of a different data source or preprocessing method with respect to their effect on LBD may be performed by evaluating their effects on the hypothesis generation step of LBD. For instance, more accurate relationship extraction should produce more accurate hypotheses.

Term Generation: Given a vocabulary and a start term, term generation outputs a list of target terms representative of future discoveries. Term generation is a prediction task which can be evaluated as a binary classification task which outputs only the terms it thinks are true discoveries. Using this idea, previous works [85, 89, 87] used ROC curves to evaluate LBD, however LBD has a severe class imbalance, where there are many more terms than true discoveries. We use PR curves which are better for classification tasks with class imbalances [99], and quantify performance with a single number using the area under the curve. Term generation and the downstream tasks of filtering and ranking are all often very dependent on one another, and term generation may be best analyzed based on its impact to these downstream tasks.

Term Filtering: As with term generation, term filtering is a binary classification task. Each term input from the term generation step is labeled as a true or false future discovery. Again, there is often a severe class imbalance so we use PR curves for evaluation, and the area under the curve for a single number quantification.

Target Term Ranking: Term ranking takes a list of terms as input and ranks

them by some measure of interestingness, but most often by how likely they are to be a true discovery. Since only a relatively small number of terms can be analyzed by a user, the correctness of the top few terms, and therefore the ability of the method to rank is critical. PR curves do not explicitly take rank into account. A system that incorrectly classifies the first 100 terms, and correctly classifies the next 5000 may have better precision than a system that correctly ranks the first 100 and incorrectly ranks the next 5000, but that system makes a bad ranking method for LBD. We use precision at K graphs to evaluate the target term ranking step, and the average precision at K to quantify performance with a single value.

Hypothesis Generation: The hypothesis generation step is a product of its three subsets, and can be evaluated in its entirety by the final output of it. We can use average precision at K graphs to evaluate the entire hypothesis generation step. Good term generation methods will produce a list of mainly true terms, good term filtering steps will remove most of these false terms, and good term ranking steps will then rank the true terms before any remaining false terms. The interaction of these components produces this final precision at K graph.

Hypothesis Explanation: User studies are the best method for evaluating hypothesis explanation because it is inherently human-oriented and subjective. What is understandable to one user may not be for another. For more complex methods, steps within hypothesis explanation may independently evaluated, but this is often system specific. For instance, Cameron, et al. [10] generate explanatory subgraphs, and develops a quantitative evaluation of the graph's complexity, and Cohen, et al. [90] use a method to compare the explanatory power of their vector spaces.

6.4 Results

In this section, we evaluate our indirect relatedness measures of linking term association (LTA), minimum weight association (MWA), shared B to C association (SBC), and linking set association (LSA) against the baselines of concept embeddings cosine (Emb Cos), direct co-occurrence vector cosine (Dir Cos) and linking term count (LTC). We evaluate the term filtering and term ranking components of LBD on our hybrid dataset reporting PR curves and Precision at K graphs. We use the same experimental details as described in Section 5 except a cutoff date of January 1, 2010 is used instead of January 1, 2000. That is, co-occurrence matrices and concept embeddings are generated using data published from January 1, 1975 to December 31, 2009 as a training dataset. Additionally, a frequency cutoff of 0 instead of 5 was used for concept embeddings cosine to ensure vectors were created for all terms in the vocabulary.

6.4.1 Results: Term Filtering

Figure 30 shows the PR curves of each indirect relatedness measure on our hybrid dataset. As expected, results are similar to those of the ROC curve analysis shown in Figure 28. All methods perform poorly, but the order of performance is different. Using the AUC as an indicator the order of performance from best to worst is: MWA, LTC, LTA, SBC and LSA are tied, Emb Cos, and Dir Cos. Again, we tried ranking in both descending and ascending order, and again the best results were achieved using ascending order for LTA, MWA, LSA, Emb Cos, and Dir Cos, which are sorted in ascending order in Figure 28. LTC and SBC achieved the best results with and are displayed in descending order. Again, Dir Cos performs on par with random.

Using a PR curve rather than ROC curve we can see finer details into each

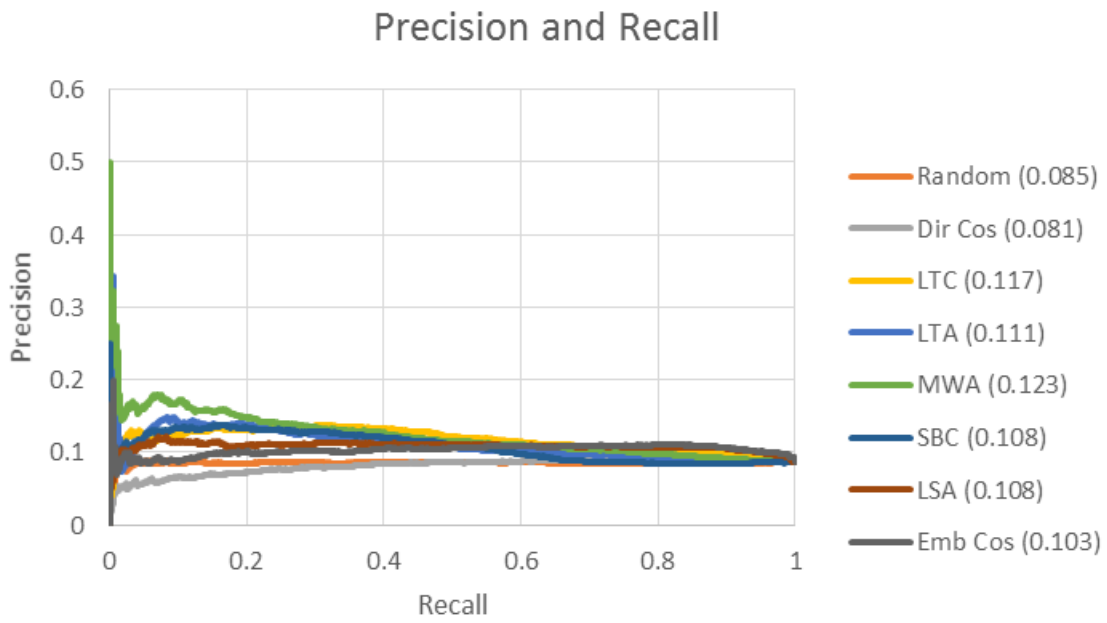

Fig. 30. Precision and recall of indirect relatedness measures on our hybrid dataset. Each measure's AUC is shown in parentheses.

indirect relatedness measure's performance. All measures converge to 9.35% precision, which is the percentage of true to false samples in the hybrid dataset. MWA performs noticeably better than other measures at low levels of recall, and Dir Cos performs worse than random at low levels of recall. All measures perform similarly as recall increases.

6.4.2 Results: Term Ranking

Figure 31 shows the precision at K graphs of each indirect relatedness measure on our hybrid dataset. These results show LTC as the best performing term ranking method; it achieves the highest average precision at K and much higher average precision for the first one hundred ranked terms. MWA is the next best performing measure achieving the second highest average precision at K, and performs visibly better for the first 25 ranked terms. Dir Cos and SBC performs the worst. They

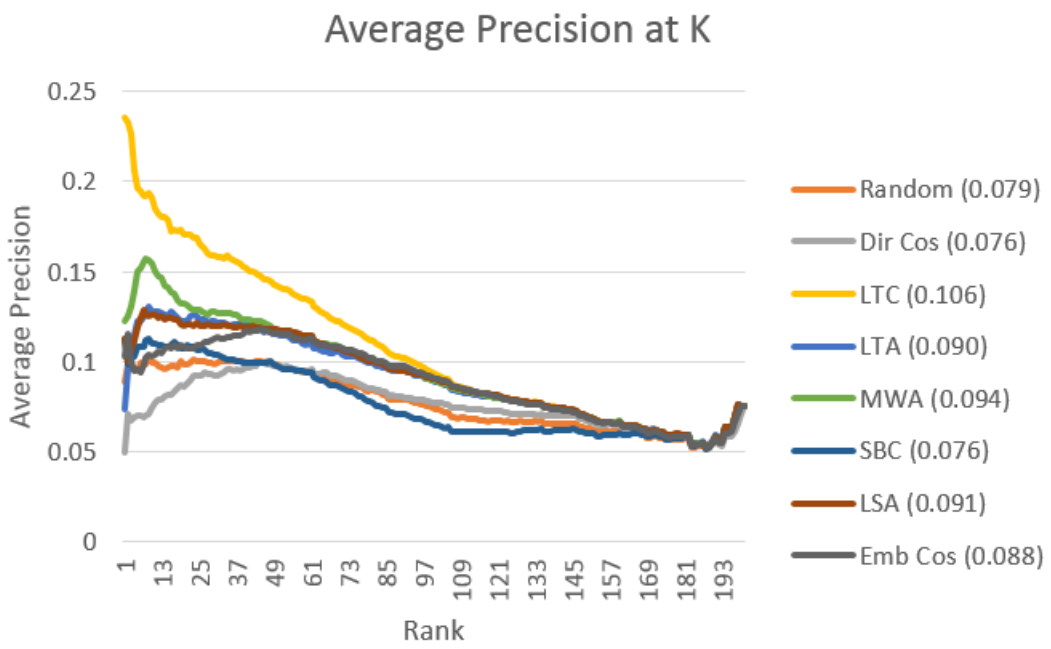

Fig. 31. Precision at K graphs of indirect relatedness measures on our hybrid dataset. Each measure's average precision at K is shown in parentheses.

have the same average precision at K, but Dir Cos performs worse than random for the top ranked samples, and SBC performs better than random for the top ranked samples. The other methods have similar performance. The order of performance based on average precision at K is LTC, MWA, LSA, LTA, Emb Cos, SBC, and Dir Cos.

The evaluation methods of PR curves and Precision at K graphs measure different properties of term filtering and ranking methods, and therefore show different results. MWA is the best performing method for term filtering, and LTC is the best performing method for term ranking. These could therefore be used in together for the term filtering and term ranking tasks of LBD. All methods perform poorly for both term filtering and term ranking which leaves room for improvement with future work in this area.

6.5 Conclusions

In this chapter, we developed a hybrid evaluation dataset and assigned evaluation methods for each component of LBD. We compared a MEDLINE co-occurrence-based dataset, a SemMedDB relationship-based dataset, and our newly created hybrid dataset using ROC curves. This revealed biases in both our co-occurrence, and Sybrandt, et al.'s evaluation datasets which we corrected in our hybrid dataset. The results with our hybrid dataset indicate that relatedness may not be the best indicator of future discoveries, but instead surprising or unexpected relationships may be better predictors. Using this knowledge, we can reverse the sorting order of our indirect relatedness measures, or use them in conjunction with other measures. We proposed using PR curves, precision at K graphs, and user-studies for LBD evaluation, and evaluated the indirect relatedness measures for the term filtering and term ranking steps of LBD. No evaluated method performed well for either task, however MWA performed the best for term filtering, and LTC performed the best for term ranking, indicating that these can be used together to create an effective LBD system.

CHAPTER 7

VISUALIZATION

In this chapter, we address our third critical problem of LBD, difficulty interpreting LBD output. We propose a method of visually summarizing LBD target term list output. We create an interactive graphical environment that allows a user to explore LBD output in its entirety. Our system is concerned primarily with visualizing the output of the open-discovery portion of LBD, which relates the hypothesis generation step in our framework (Section 3.2). We take as input an LBD target term list and output a weighted hierarchical cluster tree. The clustering algorithm groups the most semantically related terms first, resulting in a tree, where the leaf nodes are individual terms, and ascendant nodes are sets of increasingly less related terms. This produces a tree in which the nodes near the root consist of broad groupings of LBD output that identify thematic sets of related terms, and nodes near the leaves consist of sets of closely related terms that indicate potential discoveries. Displaying results in this manner means that the number of terms that need to be analyzed simultaneously is greatly reduced, promoting a comprehensive understanding of LBD output as a whole. The tree structure allows the user to ignore uninteresting, too obvious, or too unlikely branches of the tree entirely, and the interactive interface encourages the user to explore branches of the hierarchy they find most interesting or surprising.

To help in the user's exploration, we weight the tree using our indirect relatedness measure, linking set association (LSA). Specifically, we scale the thickness of edges, scale the size of nodes, and color-code edges to identify and distinguish the most interesting paths and sets of terms. This method of displaying and interacting with

data is intuitive, and facilitates a greater comprehension of LBD output as a whole, helping a user to quickly identify the most promising discoveries. The development of this system yields the following novel contributions:

1. Automatic Functional Group Discovery - we apply hierarchical clustering algorithms to an LBD target term list to automatically identify groups of closely related terms at varying levels of specificity. In effect, this automatically identifies functional groups in an LBD target term list.

2. Functional Group Ranking - we estimate the interestingness of the functional groups using indirect relatedness measures.

3. Comprehensive LBD Visualization - we develop a visualization system that combines these contributions to create an interactive user interface in which LBD target term output is displayed.

We begin the chapter by describing the overall architecture of the system, then by describing hierarchical clustering and visualization. This is followed by experimental details and results of replicating the historic Raynaud's Disease - Fish Oil Discovery. Lastly, conclusions, limitations, and future work are presented.

7.1 Contributions

In this section, we describe the novel contributions of this chapter, and how they are integrated into a comprehensive LBD visualization system. First, we briefly describe the architecture of our overall system. Then, we discuss each novel contribution in detail.

Figure 32 shows the overall architecture of our system. We use our ABC co-occurrence based LBD system (described in section 7.2) to generate a set of target

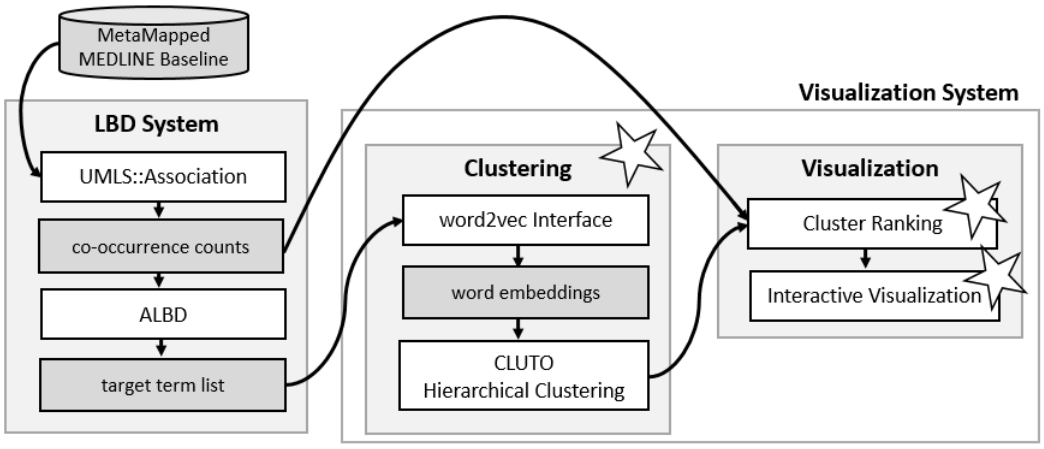

Fig. 32. An overview of the system presented in this work. Processes are in white, and data is in darker gray. Stars indicate areas where novel contributions are integrated.

terms. These target terms are input into our visualization system which contains a clustering and a visualization component. The clustering system performs *automatic functional group discovery* and finds and organizes functional groups of varying specificity into a hierarchical cluster tree. Each node of the tree contains a set of terms indicating a functional group. The hierarchical cluster tree is input into our visualization system, where each functional group is ranked in the *cluster ranking* step using LSA, and lastly the tree is displayed in an *interactive visualization*. We describe the automatic functional group discovery and visualization systems in the next sections.

7.1.1 Automatic Functional Group Discovery

We input an LBD target term list into a hierarchical clustering algorithm to automatically identify functional groups. The algorithm constructs a binary hierarchical cluster tree, for which each target term is a leaf node, and the root is a cluster containing all target terms. Intermediate nodes represent functional groups and are sets that contain all descendant target term leaf nodes. In the hierarchical tree, the

most related terms and clusters are grouped first, as the node's depth increases, so does its specificity. Similarly, as the depth of a cluster increases, the number of terms in the cluster decreases. This means that nodes containing a few highly related terms are near the leaves, and nodes containing many, more loosely related terms are near the root. Figure 33 shows an example. Each node in the tree is a cluster. Single term clusters are at the leaf nodes, and are indicated by an asterisk. All other nodes contain the set of all descendant leaf node terms. For instance, the "fish oil related terms" node contains all terms indicated by asterisks. Descendant from that is the "fish oils" node, which contains the terms *cod liver oil*, *menhaden oil*, and *fish oil*.

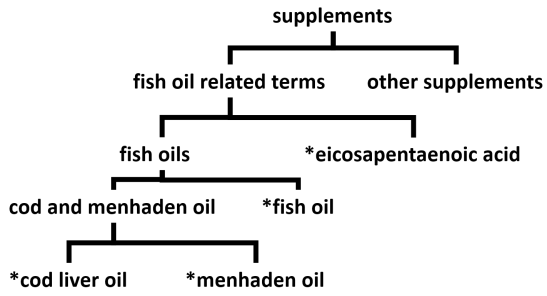

Fig. 33. An example of hierarchical clustering, in which the root node is all supplements, single terms are leaf nodes and are indicated by asterisks.

In our method, we use pre-existing hierarchical clustering algorithms for the following reasons: (1) Hierarchical clustering methods avoid the problem of defining the "number of clusters", or a cluster stopping metric; both of which are difficult to define and are often subjective. Hierarchical clustering finds clusters of varying specificity automatically, by effectively finding clustering solutions at every possible "number of clusters". (2) Using hierarchical clustering produces the unique opportunity to display the hierarchical cluster tree to the user. This cluster tree allows the user to visually explore the target terms output by an LBD system in a structured and meaningful manner. (3) Many pre-existing clustering techniques are designed for

large, high dimensional datasets, and are therefore scalable and efficient. (4) Most pre-existing clustering techniques allow any distance metric to be defined, meaning we can use distance metrics that are effective for this task, and we can redefine the distance metric as more advanced methods are developed.

7.1.2 Comprehensive Visualization

The entire hierarchical cluster tree of LBD target terms is displayed at once in an interactive environment that allows a user to explore branches of the tree they find interesting. Our intended use case is for the user to begin exploration at the root node, and descend the tree by following paths they find most interesting or surprising. In this scenario, the user first encounters broad groups of loosely related terms, and as they descend the tree, they discover sets of increasingly specific sets of closely related terms, until arriving at single term leaf nodes. The hierarchical tree is a binary tree, meaning that at each node, the user may select one of two different paths. While backtracking, and exploring multiple paths is possible and encouraged, the size of the tree makes exploring all possible paths or leaf nodes impractical. Therefore, our goal is to help the user find the most specific and most interesting terms while examining as few nodes as possible. We do this by visually indicating our estimates of interestingness of individual paths and nodes to guide the user down the hierarchy.

The interestingness of nodes is estimated by the our indirect relatedness measure, LSA (see Chapter 5) between the terms of that cluster and the starting term set. We scale the size of the displayed nodes, such that more interesting nodes are larger, and less interesting nodes are smaller. Nodes are labeled to give an estimate of the types of terms descendant from it at a glance. The label is generated by selecting the term in that cluster whose vector cosine distance is closest to the cluster centroid. The interestingness of edges, and therefore paths is estimated as the sum of all descendant

nodes' estimated interestingness. We scale the thickness of edges such that more interesting edges are thicker, and less interesting edges are thinner. This indicates the amount of interestingness a user may encounter by following that path. Since it may be difficult to distinguish between edges with similar thicknesses, and at each node a user may choose between two possible paths, we color the thinner edge red, and the thicker edge blue.

7.2 Experimental Details

We use our own implementations and several pre-existing components to form a complete LBD system, and evaluate its performance via open discovery replication of Swanson's Raynaud's Disease - Fish Oil Discovery. As shown in Figure 32, the system architecture consists of a data source, and three primary components, the LBD system, the clustering system, and the visualization system. Details for the results shown are described here.

Data Source: As a data source, we use titles and abstracts published between January 1, 1983 to December 31, 1985 [80] from the 2015 *MetaMapped MEDLINE baseline*[1].

LBD System: We collect co-occurrence count using UMLS::Association version 1.3's CUI Collector tool[2] with a window size of 8. These counts are used later to rank clusters, and here as input to *Association Literature Based Discovery (ALBD)* version 0.05[3]. ALBD is a Perl implementation of an ABC co-occurrence model of LBD. We use the UMLS concepts of Raynaud's Disease (`C0034734`) and Raynaud's Phenomenon (`C0034735`) as A terms, since they are listed as synonyms within the

[1] `https://ii.nlm.nih.gov/MMBaseline/index.shtml`

[2] `https://metacpan.org/release/UMLS-Association`

[3] `https://metacpan.org/release/ALBD`

UMLS [75]. We apply a semantic type filter at both the A to B and the B to C linking steps. The filters are purposefully broad to avoid overfitting [62], specifically: B terms are restricted to the UMLS semantic groups (using UMLS 2016AA) of "Chemicals and Drugs (CHEM)", "Disorders (DISO)", "Genes and Molecular Sequence (GENE)", "Physiology (PHYS)", and "Anatomy (ANAT)", and C terms are restricted to semantic groups of "Chemicals and Drugs (CHEM)" and "Genes and Molecular Sequence (GENE)". We apply a linking term count (LTC) threshold to the target term list output by the LBD system to limit the number of target terms to 3000, however the terms ranked 2995 through 3016 were tied, resulting in a total of 3016 to be clustered. This threshold is applied due to memory constraints of the current implementation of clustering algorithms.

Clustering System: We perform clustering on word embeddings of the target terms output by the LBD system. We generate UMLS concept embeddings using co-occurrences information between UMLS concepts rather than individual terms [117] using *word2vec-interface* version 0.03[4], a Perl interface to the *word2vec* package [23]. Continuous bag of words (CBOW) embedding model, a window size of 8, a frequency cutoff of 5, and default settings for all other parameters are used. Hierarchical clustering is performed using *CLUTO* version 2.1.1 [122], a clustering software package designed for large sample size, high dimensionality datasets. The *vcluster* function with repeated bisection clustering algorithm with the Full Tree option enabled, and default settings for all other parameters are used.

Visualization System: The hierarchical cluster tree of target terms is input into the visualization system. We use our implementation of LSA in UMLS::Association version 1.7 using Pearson's Chi-Squared, and all other settings as default to compute

[4] `http://search.cpan.org/dist/Word2vec-Interface/`

the association scores between each cluster and the start term set. The visualization is generated by yEd Graph Editor[5], a graph visualization program using the tree-balloon layout. Nodes sizes are displayed as their set-association score linearly scaled, relative to all other nodes to a value between 1 and 100, and edge widths as $log(1 + weight)^3$ of the edge. These scalings are used (rather than linear) to better display the range of values generated by our system.

7.3 Results

In this section, we describe the results of replication of Swanson's Raynaud's Disease - Fish Oil Discovery. We use the system described in section 7.2 to generate the system output shown in Figure 34. Figure 34 shows our generated hierarchical cluster tree in its entirety. The labels tell us the types of nodes we can expect to see descendant from that branch. By clicking on a few nodes, and visually inspecting the tree, we quickly get a general idea of the tree's overall structure. Exploring the tree in more detail, and starting at the root node, marked by a star, there are two edges. The edges appear to be similar widths, but the blue edge leads to far fewer nodes. This is significant, and tells us that the nodes in the direction of the blue path are much more interesting, since their sum of interestingness is similar to a much larger set of nodes. The target nodes of these two edges are labeled "Enzymes" and "Clonidine". This indicates that the red edge likely leads to terms loosely related to "Enzymes" (but due to the label's generality and number of descendant nodes may contain several different categories of terms), and the blue edge leads to terms associated with "Clonidine", a drug used to treat hypertension, indicating that the descendant terms are likely associated with effects on blood. This is exciting, since

[5] `https://www.yworks.com/products/yed`

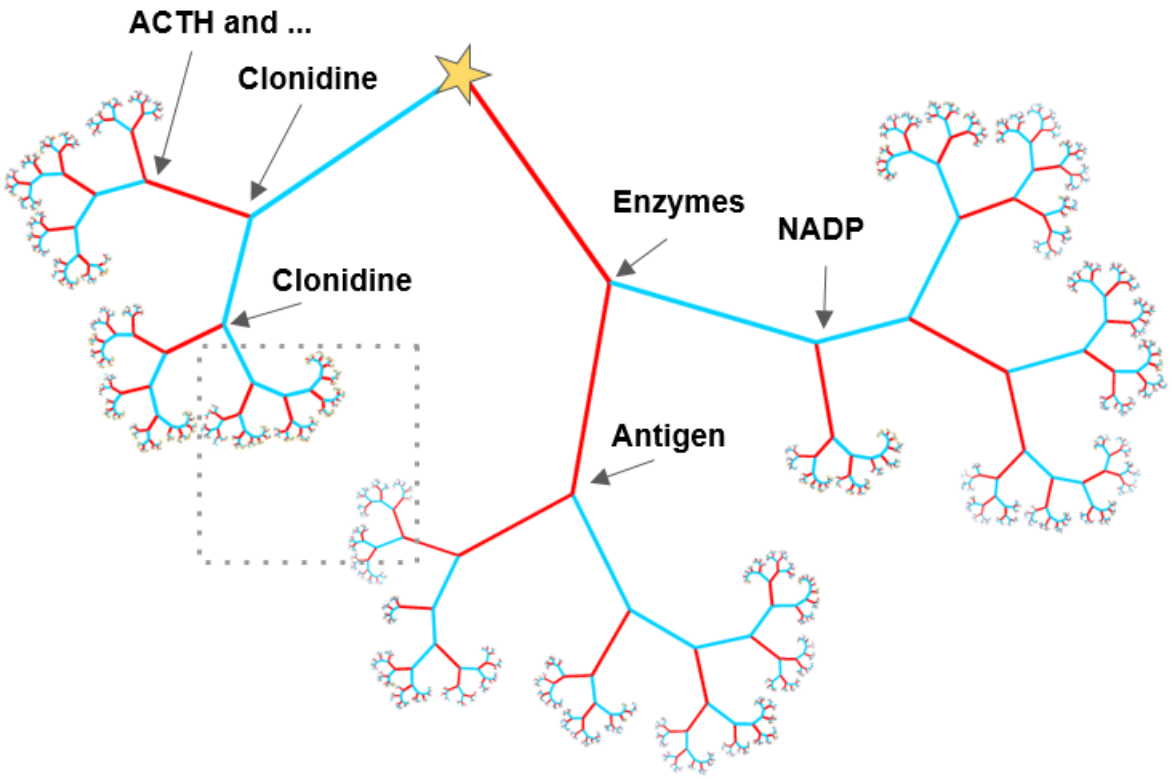

Fig. 34. Fully zoomed out visualization of the Raynaud's Disease - Fish Oil Discovery replication. Labels were manually added, because at this zoom level nodes must be manually inspected to show labels. Figure 35 shows a zoomed in version of the dotted rectangle.

fish oil's effects on blood viscosity, platelet aggregation, and vascular reactivity have been identified as the primary functional pathways for which fish oil treats Raynaud's Disease [59].

Descending the tree in the blue edge direction, we reach the next junction. Here, the edge weights appear similar, and the descendant tree appears more balanced, with similar numbers of nodes in either direction. The red edge leads to a node labeled "ACTH and synthetic analog preparations", and the blue path to a different node, labeled once again "Clonidine". ACTH can induce hypertension and Raynaud's disease is often associated with hypertension, but since the path to Clonidine is blue we choose to descend the tree in the blue direction. In a real-world environment though, at this point, we have reduced the number of terms to a manageable level, where a user could further explore both paths in more detail. Following the blue path, at our next junction, we must decide between two paths, one leading to "aspirins" and one to "morphines". The blue path leads to "aspirins", and its descendant tree is shown in Figure 35.

The tree in Figure 35 is sufficiently small to explore the branches in more detail. By clicking on each node we see its label, or we can zoom in even further to see sets of labels next to the nodes. This manual analysis shows that these terms are mostly calcium channel blockers and anti-hypertensive drugs, listed among them are Felodipine and Nisoldipine. Interestingly, present-day recommended pharmacological treatments for Raynaud's Disease are calcium channel blockers and anti-hypertensive drugs, and two of the recommended drugs are Felodipine and Nisoldipine (although these were unknown within our time-sliced dataset). Eicosapentaenoic acid, the active ingredient in fish oil is also found in this sub-tree.

By following the most interesting path indicated by our system, we immediately eliminated about two-thirds of the LBD system output. This narrowed our discovery

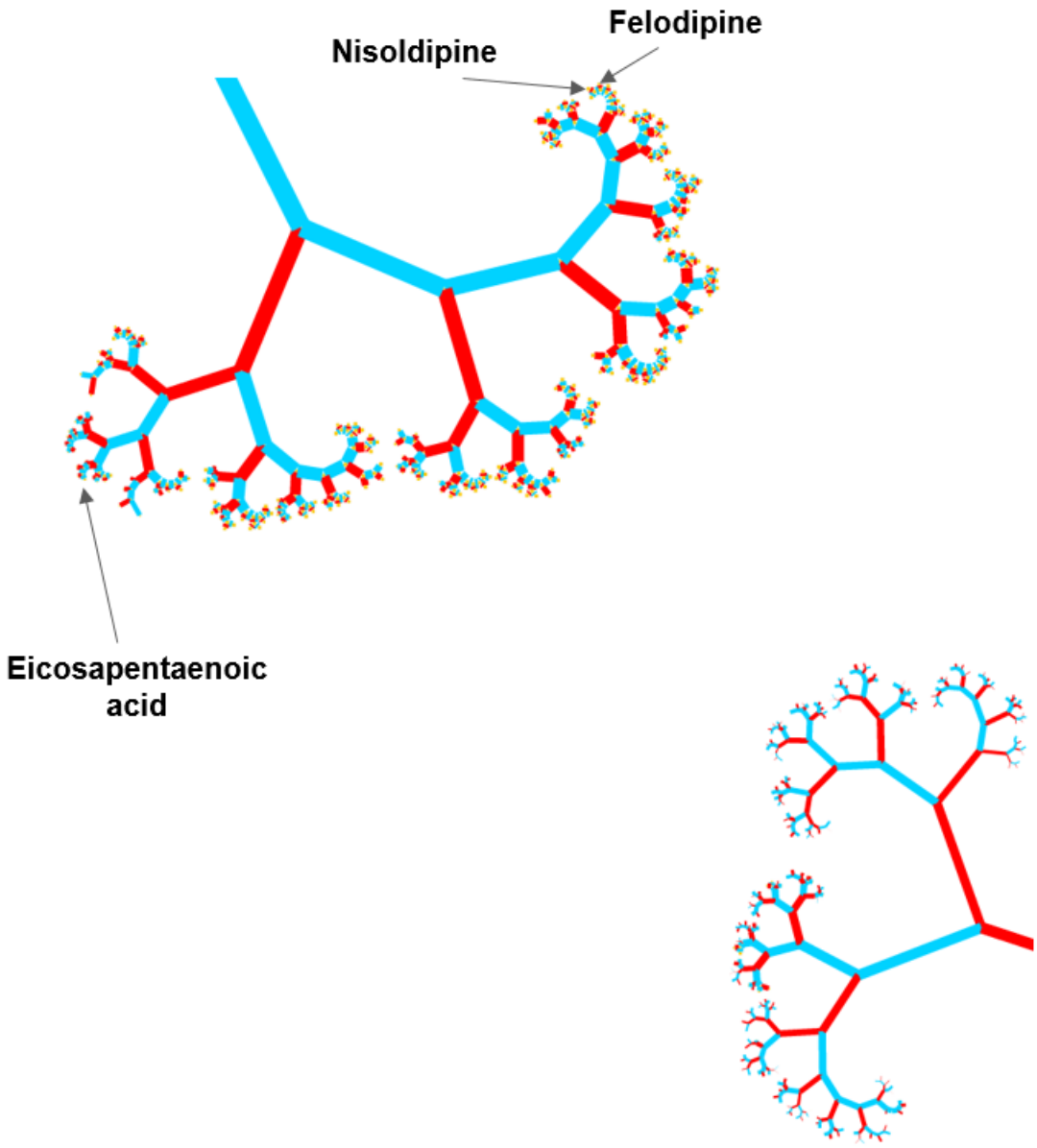

Fig. 35. Portion of Figure 34 found by following blue paths. Labels were manually added, because at this zoom level nodes must be manually inspected to show labels. The top tree shows a sub-tree of general DNA related terms, which our system deemed less interesting. "ACTH and ..." abbreviates "ACTH and synthetic analog preparations"

search to drugs and compounds that are related by their effects on blood, specifically hypertension. Further descending the path in the recommended directions, we arrive at calcium channel blockers and anti-hypertensive drugs, which are a class of drugs that are the present-day recommended treatment for Raynaud's Disease. Our system gives the user an overall understanding of the LBD output, and using linking set association to estimate interestingness, we lead the user directly to treatments for Raynaud's Disease, including present-day drugs used to treat the disease, Eicosapentaenoic Acid, the primary active ingredient in fish oil.

7.4 Conclusions

In this chapter, we presented a novel method of visualizing and understanding LBD target term output. This method required the novel application of hierarchical cluster algorithms to LBD target term output, the use of indirect relatedness measures, and their integration novel LBD visualization system. We showed the efficacy of our visualization system by replicating the historic Raynaud's Disease - Fish Oil Discovery. Doing so, we found that our system quickly eliminates the majority of the target terms, and leads the user to blood-related terms, the primary pathway in which fish oil can treat Raynaud's Disease. Descending the tree, our system leads the user to a class of drugs used in present day to treat Raynaud's Disease, and to Eicosapentaenoic Acid, the active ingredient in fish oil.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions:

In this dissertation, we presented our work at developing more effective LBD systems. We identified and addressed three areas of LBD in critical need of improvement, these include:

Problem 1: Over-generation of knowledge - LBD systems tend to create too many hypotheses, causing promising and meaningful hypotheses to be buried within false, uninteresting, or too obvious ones.

Problem 2: Lack of meaningful evaluation methods - LBD is difficult to evaluate, and evaluation methods are often ad-hoc and system specific. Without standard evaluation methods, LBD systems and components cannot be quantitatively compared or objectively improved.

Problem 3: Difficulty interpreting output - LBD systems often output hypotheses as a simple list of terms. This combined with their over-generation of knowledge means a user is presented with a list of hundreds or thousands of often unrelated terms, which are meaningless without extensive manual review.

We addressed Problem 1, the over-generation of knowledge by developing indirect association measures for hypothesis ranking and filtering in LBD. Indirect relatedness is not a well studied field, so to develop indirect relatedness measures, we first studied methods to estimate direct relatedness. Estimating direct relatedness is a well established field, but not directly applicable to LBD, since terms in LBD hypotheses are

not directly related. We present our study of direct relatedness in Chapter 4, where we presented the following contributions:

1. Concept Association and Expansion - we use association measures between concepts to estimate direct relatedness, and introduce concept expansion, which in combination with concept associations is a novel methodology that accounts for lexical variation at both the synonymous and hyponymous levels.

2. Set Associations - we extend association measures to quantify relatedness between sets of terms rather than individual term-term pairs.

3. An analysis of association measures for direct relatedness, for which we achieve state of the art results.

4. An analysis of vector measures for direct relatedness, for which we achieve state of the art results.

This study on direct relatedness informed our development of indirect relatedness measures by indicating that both association measures and vector measures were good candidates for application to indirect relatedness, and provided an understanding of how parameters affect their performance. In Chapter 5, we develop indirect association measures and apply vector measures to estimating indirect relatedness. Specific contributions are:

1. Indirect Association Measures - we develop four indirect association measures, including linking term association (LTA), minimum weight association (MWA), shared B to C association (SBC), and linking set association (LSA).

2. The development of a dataset and method for evaluating the ability to estimate future relatedness.

3. An analysis of indirect association measures and vector-based measures on their ability to estimate direct and future relatedness.

In Chapter 6 we addressed Problem 2, the lack of meaningful evaluation methods. We developed a standard evaluation dataset and methods for evaluating each component of the LBD process. Specific contributions included:

1. Development of a standard evaluation dataset (our hybrid dataset).

2. Assignment of evaluation methods for LBD components in isolation and in combination.

3. Evaluation of indirect relatedness measures for term filtering and term rankings steps of LBD.

Using these standard evaluation methods and dataset, we evaluated indirect relatedness measures on the tasks of the term filtering and term ranking for LBD. The results indicated that unlikely or surprising future associations may be better indicators of future discoveries than highly likely future associations. This finding is exciting for future research, but does not reduce the usefulness of our indirect relatedness measures, which can be used on conjunction with measures of surprise, or be used as an indicator of unlikeliness (by reversing the rankings).

In chapter 7 we addressed Problem 3, the difficulty in interpreting LBD output by developing an interactive visualization environment to explore LBD output. This visual environment displays LBD system output as an interactive hierarchical cluster tree of automatically discovered functional groups. Visual cues are added to aid in interpretation. Its development required the following contributions:

1. Automatic Functional Group Discovery - we apply hierarchical clustering algorithms to automatically find functional groups in the LBD output.

2. Functional Group Ranking - we estimate the interestingness of the functional groups using indirect relatedness measures.

3. Interactive Visualization - we use the clustering hierarchy and cluster rankings to create an interactive visualization of our system's LBD output. We use visual cues to aid the user in exploring cluster tree interactively.

We showed the effectiveness of the visual cues and hierarchical structure of our visualization by re-discovering present day treatments for Raynaud's disease.

In addition to the contributions listed above, we broke the LBD process into a series of discrete components, where each component has well defined inputs, outputs, and goals. We developed an evaluation framework which combines our new evaluation dataset with evaluation methods for each of the components. With these evaluation methods, we created quantifiable and comparable evaluation standards and techniques. This make it possible for individual components to be developed and compared in isolation from, and used interchangeably with different LBD systems and paradigms. This reduces the complexity of developing an entire LBD system. It allows researchers to focus on smaller, well defined tasks for which measurable improvements can be definitively made. Just as many complex natural language processing systems are broken into well defined, compartmentalized pipelines, and systems are built from publicly available components, we hope the same for LBD. We hope these smaller, quantifiable goals lead to an increased interest and advancement of the field, and that application specific LBD systems can be made largely from pre-existing components, rather than by writing all the code from scratch.

8.1.1 Future Work

This dissertation creates several immediate areas of future work and research opportunities. Our visualization system has several limitations which could be addressed in future research. These include:

1. Visualization limitations - thickness of the edges is difficult to distinguish, and labels are only displayed when zoomed in. Better visualization software which allows dynamic edge thickness and flexibility when displaying labels could be developed.

2. Evaluation - an evaluation of our visualization system with user studies would provide insights into how it is used and what improvements could be made.

3. Develop more efficient clustering methods - the clustering algorithms implemented in CLUTO cannot produce hierarchical cluster trees with all terms output by LBD. This is caused by memory constraints; developing more efficient clustering algorithms could alleviate this.

4. Cluster naming and summarization - naming clusters as their cluster centroid is simplistic. A better method assign a cluster label, set of labels, or even a cluster summary could be developed.

We applied our set associations measures to LBD, but we believe they are applicable to a variety of problems, such as: (1) sentence or phrase similarity, where the set similarity between all terms of two sentences could be calculated to quantify the two sentence's similarity; (2) automatic keyword identification, where the set similarity of all terms in an abstract to a list of keywords could be used to automatically assign keywords to documents; (3) word sense disambiguation, where the set association

between an ambiguous term and the terms of its surrounding context could indicate its sense.

Our indirect association measures performed well, but we believe this performance could be improved by "cleaning" the linking term sets. Term that don't contribute valuable information relative to the context in which the terms are linked may decrease performance. By using only informative terms in SBC or LSA, we believe their performance would increase. Elimination of terms could be performed as simply as applying occurrence count filters, information content, or more complex methods could be developed.

The work of this dissertation focused primarily on the open-discovery (one-node search) model of LBD, which corresponds to the hypothesis generation step (Section 3.2). A large area of work, that is largely untouched in this dissertation is hypothesis explanation step of LBD. This step is similar to the closed discovery model (two node search), which can help a user better understand how the start and target terms of a hypothesis relate to one another. We envision a system in which hypotheses are generated and visualized, and a user can select a node (corresponding to a functional group) in our visualization tree. Next, hypothesis explanation is performed for the selected functional group to explain how the terms in the group relate to the start terms. Promising research in this domain includes Cameron, et al. [10], who create visual explanatory subgraphs which connect the start and target terms. Their system is impressive, but is computationally intensive and currently requires manual intervention.

8.1.2 The Future of LBD

LBD is still primarily a theoretical field that is only being used in a handful of laboratories. As the overwhelming number of publications continues to grow, LBD

must make the jump from theory to application. However, most researchers aren't using LBD, and many aren't even aware it exists. This is a major problem, and while creating more effective LBD systems and components is necessary for creating useful tools, we must also ask the question, what is the role of LBD in research?

LBD as a fully automated hypothesis generation system is still just science fiction, and will likely remain so for the near future. Humans are an integral part of scientific research. Machines don't understand the documents they digest, and they can't make truly intelligent discoveries from them. Instead, machines extract information from huge amounts of data. They assemble this information, display it, and can even infer new information from it, but the ability to make an intelligent discovery still requires a human. Machines cannot identify what is truly interesting, meaningful, or relevant. They have no common sense, but more importantly, they have no real world context in a discovery's domain. Understanding context helps to understand the relevance of information, and to decide if a hypothesis is too obvious, too far fetched, or if it is actually novel and interesting. Integrating context into LBD systems, is therefore necessary to create truly intelligent discoveries. We see a great opportunity to get LBD into the hands of researchers, and a way to provide context by integrating LBD systems into content management and recommendation systems.

Researchers read scientific publications on a regular basis, and there is a need to manage the library of papers they have read, find new papers to read, and assemble knowledge from those papers. This problem is shared across scientific disciplines, and by creating a tool that is immediately useful for researchers, it will become an integral part of the research process. This means we can get immediate feedback to inform the development of new components, and as our system becomes an integral part of their research process, the context in which they work can be estimated by the publications they read, and by those that they publish. LBD was first proposed as

a tool to bridge disciplines, and to break researchers out of their research silos. The content researchers read and publish is what makes up their silo, and what better way to understand a research silo, than to serve as the system that provides information to the user.

We envision an LBD system that estimates a researcher's knowledge and the context in which they work, through their own publications, and the content they read. Information extracted from these documents can be represented as a knowledge graph, and gaps in this graph can be filled both by recommending papers to read, and by offering information from papers outside of their discipline. This is similar to link prediction, however the links are derived from information known to science, but new to the scientist. Exposing users to information that is new to them, but not necessarily new to science has been shown to be one of the most useful aspects of LBD [123].

Collaboration is a vital part of scientific research, and by understanding a researcher's context a collaboration recommendation system could be implemented. In this system, researchers with complementary knowledge and goals (estimated by the papers they read and publish) can be automatically found and recommended (LBD as a collaboration finding system has been proposed [124]).

Knowing a researcher's context also helps in the discovery of new knowledge, particularly in the hypothesis generation process. The context in which a researcher works is a critical component in estimating interestingness of any automatically generated hypotheses. The problem of over-generation of knowledge during hypothesis generation can be alleviated by presenting only hypotheses relevant to a researcher's context. By narrowing the context in which LBD is performed, LBD becomes a much more tractable problem. Knowledge graphs can be built based on a researcher's readings, and new hypotheses can be suggested and offered for further research. This is

personalized LBD; LBD tailored to the research of a specific scientist, and the context in which they work.

To summarize, we see several immediate areas of future research that expand upon the contributions presented in this paper, these include improvements to the clustering and visualization components of our system, new applications of set association, and improvements to indirect association measures. Hypothesis explanation is a field largely left untouched by this dissertation, but is critical for LBD, and deserves future research. Looking beyond these immediate improvements, we think the next critical step in developing useful LBD systems is to get LBD into the hands of scientists, and to integrate a scientist's specific research context into the LBD process. This context can be estimated by the articles a scientist reads and publishes, and we can get our systems into their hands by producing tools that are immediately useful to researchers; by integrating LBD systems with content management systems. By understanding the context in which a researcher works, the scope of LBD is narrowed, and only hypotheses relevant to their research are then generated. By getting scientists to use our systems, we receive feedback into how to improve them, and hypotheses can be presented within a framework already familiar to the user.

Appendix A

ABBREVIATIONS

AMW	average minimum weight
CBOW	continuous bag of words
CRG	cumulative relatedness graph
CUI	Concept Unique Identifier
LBD	literature based discovery
LSA	linking set association
LTA	linking term association
LTC	linking term count
MAP	mean average precision
MeSH	Medical Subject Headings
MWA	minimum weight association
PR	precision and recall
ROC	receiver operating characteristic
SBC	shared B to C association
SG	skip gram
SNOMED CT	Systematized Nomenclature of Medicine - Clinical Terms
SVD	singular value decomposition
UMLS	Unified Medical Language System
WSD	word sense disambiguation

REFERENCES

[1] Lawrence Hunter and K Bretonnel Cohen. "Biomedical language processing: What's beyond PubMed?" In: *Molecular cell* 21.5 (2006), pp. 589–594.

[2] Lutz Bornmann and Rüdiger Mutz. "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references". In: *Journal of the Association for Information Science and Technology* 66.11 (2015), pp. 2215–2222.

[3] Don R Swanson. "Fish oil, Raynaud's syndrome, and undiscovered public knowledge". In: *Perspectives in biology and medicine* 30.1 (1986), pp. 7–18.

[4] Raoul Frijters et al. "Literature mining for the discovery of hidden connections between drugs, genes and diseases". In: *PLoS Comput Biol* 6.9 (2010), e1000943.

[5] T Cohen et al. "Predicting High-Throughput Screening Results With Scalable Literature-Based Discovery Methods". In: *CPT: Pharmacometrics & Systems Pharmacology* 3.10 (2014), pp. 1–9.

[6] Jonathan D Wren et al. "Knowledge discovery by automated identification and ranking of implicit relationships". In: *Bioinformatics* 20.3 (2004), pp. 389–398.

[7] Eftychia Lekka et al. "Literature analysis for systematic drug repurposing: A case study from Biovista". In: *Drug Discovery Today: Therapeutic Strategies* 8.3 (2012), pp. 103–108.

[8] Dimitar Hristovski et al. “Combining semantic relations and DNA microarray data for novel hypotheses generation”. In: *Linking literature, information, and knowledge for biology*. Springer, 2010, pp. 53–61.

[9] Yanhui Hu et al. “Analysis of genomic and proteomic data using advanced literature mining”. In: *Journal of Proteome Research* 2.4 (2003), pp. 405–412.

[10] Delroy Cameron et al. “Context-driven automatic subgraph creation for literature-based discovery”. In: *Journal of Biomedical Informatics* 54 (2015), pp. 141–157.

[11] Nancy C Baker, Denis Fourches, and Alexander Tropsha. “Drug side effect profiles as molecular descriptors for predictive modeling of target bioactivity”. In: *Molecular Informatics* 34.2-3 (2015), pp. 160–170.

[12] Alexa T McCray, Anita Burgun, and Olivier Bodenreider. “Aggregating UMLS semantic types for reducing conceptual complexity”. In: *Studies in health technology and informatics* 84.0 1 (2001), p. 216.

[13] Alan R Aronson and François-Michel Lang. “An overview of MetaMap: historical perspective and recent advances”. In: *Journal of the American Medical Informatics Association* 17.3 (2010), pp. 229–236.

[14] Thomas C Rindflesch and Marcelo Fiszman. “The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text”. In: *Journal of Biomedical Informatics* 36.6 (2003), pp. 462–477.

[15] Halil Kilicoglu et al. “SemMedDB: a PubMed-scale repository of biomedical semantic predications”. In: *Bioinformatics* 28.23 (2012), pp. 3158–3160.

[16] Dimitar Hristovski et al. “Using Literature-Based Discovery to Explain Adverse Drug Effects”. In: *Journal of medical systems* 40.8 (2016), pp. 1–5.

[17] Graciela Rosemblat et al. “A methodology for extending domain coverage in SemRep”. In: *Journal of Biomedical Informatics* 46.6 (2013), pp. 1099–1107.

[18] T. Pedersen et al. “The Ngram statistics package (Text::NSP): A flexible tool for identifying ngrams, collocations, and word associations”. In: *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*. Association for Computational Linguistics. 2011, pp. 131–133.

[19] John R Firth. “A synopsis of linguistic theory, 1930-1955”. In: *Studies in linguistic analysis* (1957).

[20] T. Pedersen. “Unsupervised corpus-based methods for WSD”. In: *Word sense disambiguation: algorithms and applications* (2006), pp. 133–166.

[21] S. Deerwester et al. “Indexing by latent semantic analysis”. In: *Journal of the American society for information science* 41.6 (1990), p. 391.

[22] AKM Sabbir, A.J. Yepes, and R. Kavuluru. “Knowledge-Based Biomedical Word Sense Disambiguation with Neural Concept Embeddings and Distant Supervision”. In: *arXiv preprint arXiv:1610.08557* (2016).

[23] T. Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[24] T. Pedersen et al. “Measures of semantic similarity and relatedness in the biomedical domain”. In: *Journal of Biomedical Informatics* 40.3 (2007), pp. 288–299.

[25] R. Rada et al. “Development and application of a metric on semantic nets”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 19.1 (1989), pp. 17–30.

[26] D. Lin. “An information-theoretic definition of similarity”. In: *Proceedingins of the International Conference on ML*. 1998, pp. 296–304. URL: `citeseer.ist.psu.edu/95071.html`.

[27] P. Resnik. “Using information content to evaluate semantic similarity in a taxonomy”. In: *Proceedings of the 14th International Joint Conference on AI*. 1995, pp. 448–453.

[28] Zhibiao Wu and Martha Palmer. “Verbs semantics and lexical selection”. In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 1994, pp. 133–138.

[29] Alexander Maedche and Steffen Staab. *Comparing ontologies-similarity measures and a comparison study*. AIFB, 2001.

[30] Montserrat Batet, David Sánchez, and Aida Valls. “An ontology-based measure to compute semantic similarity in biomedicine”. In: *Journal of biomedical informatics* 44.1 (2011), pp. 118–125.

[31] S. Pakhomov et al. “Semantic Similarity and Relatedness between Clinical Terms: An Experimental Study”. In: *Proceedings of the American Medical Informatics Association (AMIA) Symposium*. Washington, DC, Nov. 2010, pp. 572–576.

[32] Ronald A Fisher. “Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population”. In: *Biometrika* (1915), pp. 507–521.

[33] Ted Dunning. "Accurate methods for the statistics of surprise and coincidence". In: *Computational linguistics* 19.1 (1993), pp. 61–74.

[34] Kenneth Ward Church and Patrick Hanks. "Word association norms, mutual information, and lexicography". In: *Computational linguistics* 16.1 (1990), pp. 22–29.

[35] Kenneth W Church. "Empirical estimates of adaptation: the chance of two noriegas is closer to p/2 than p 2". In: *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics. 2000, pp. 180–186.

[36] Ted Pedersen. "Fishing for Exactness". In: *In Proceedings of the South-Central SAS Users Group Conference*. Citeseer. 1996.

[37] Kenneth W Church and William A Gale. "Concordances for parallel text". In: *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*. 1991, pp. 40–62.

[38] Kenneth Church et al. "Using statistics in lexical analysis". In: *Lexical acquisition: exploiting on-line resources to build a lexicon* (1991), p. 115.

[39] Frank Smadja. "Retrieving collocations from text: Xtract". In: *Computational linguistics* 19.1 (1993), pp. 143–177.

[40] Don Blaheta and Mark Johnson. "Unsupervised learning of multi-word verbs". In: *Proc. of the ACL/EACL 2001 Workshop on the Computational Extraction, Analysis and Exploitation of Collocations*. 2001, pp. 54–60.

[41] Ronald N Kostoff. "Literature-related discovery (LRD): Potential treatments for cataracts". In: *Technological Forecasting and Social Change* 75.2 (2008), pp. 215–225.

[42] Ronald N Kostoff, Michael B Briggs, and Terence J Lyons. "Literature-related discovery (LRD): Potential treatments for multiple sclerosis". In: *Technological Forecasting and Social Change* 75.2 (2008), pp. 239–255.

[43] Ronald N Kostoff and Michael B Briggs. "Literature-Related Discovery (LRD): potential treatments for Parkinson's disease". In: *Technological Forecasting and Social Change* 75.2 (2008), pp. 226–238.

[44] Padmini Srinivasan and Bisharah Libbus. "Mining MEDLINE for implicit links between dietary substances and diseases". In: *Bioinformatics* 20.suppl 1 (2004), pp. i290–i296.

[45] Caroline B Ahlers et al. "Using the literature-based discovery paradigm to investigate drug mechanisms." In: *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*. 2007.

[46] Christopher M Miller et al. "A closed literature-based discovery technique finds a mechanistic link between hypogonadism and diminished sleep quality in aging men". In: *Sleep* 35.2 (2012), pp. 279–285.

[47] Michael J Cairelli et al. "Semantic MEDLINE for discovery browsing: using semantic predications and the literature-based discovery paradigm to elucidate a mechanism for the obesity paradox". In: *AMIA Annual Symposium Proceedings*. Vol. 2013. American Medical Informatics Association. 2013, p. 164.

[48] Rui Zhang et al. "Exploiting literature-derived knowledge and semantics to identify potential prostate cancer drugs". In: *Cancer informatics* Suppl. 1 (2014).

[49] Spyros N Deftereos et al. "Drug repurposing and adverse event prediction using high-throughput literature analysis". In: *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 3.3 (2011), pp. 323–334.

[50] Hsih-Te Yang et al. "Literature-based discovery of new candidates for drug repurposing". In: *Briefings in bioinformatics* 18.3 (2017), pp. 488–497.

[51] Majid Rastegar-Mojarad et al. "A new method for prioritizing drug repositioning candidates extracted by literature-based discovery". In: *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*. IEEE. 2015, pp. 669–674.

[52] Majid Rastegar-Mojarad et al. "Prioritizing Adverse Drug Reaction and Drug Repositioning Candidates Generated by Literature-Based Discovery". In: *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM. 2016, pp. 289–296.

[53] Nancy C Baker. "Methods in literature-based drug discovery". PhD thesis. University of North Carolina at Chapel Hill, 2010.

[54] Ritwik Banerjee et al. "Automated suggestion of tests for identifying likelihood of adverse drug events". In: *IEEE International Conference on Healthcare Informatics*. Citeseer. 2014, pp. 170–176.

[55] Ning Shang et al. "Identifying plausible adverse drug reactions using knowledge extracted from the literature". In: *Journal of Biomedical Informatics* 52 (2014), pp. 293–310.

[56] Justin Mower et al. "Classification-by-Analogy: Using Vector Representations of Implicit Relationships to Identify Plausibly Causal Drug/Side-Effect Pre-

diction". In: *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*. 2016.

[57] Jonathan D Wren. "The 'open discovery' challenge". In: *Literature-based discovery*. Springer, 2008, pp. 39–55.

[58] Edwin J Matthews and Anna A Frid. "Prediction of drug-related cardiac adverse effects in humans - A Creation of a database of effects and identification of factors affecting their occurrence". In: *Regulatory Toxicology and Pharmacology* 56.3 (2010), pp. 247–275.

[59] Marc Weeber et al. "Using concepts in literature-based discovery: Simulating Swanson's Raynaud–fish oil and migraine–magnesium discoveries". In: *Journal of the American Society for Information Science and Technology* 52.7 (2001), pp. 548–557.

[60] Don R Swanson and Neil R Smalheiser. "An interactive system for finding complementary literatures: a stimulus to scientific discovery". In: *Artificial intelligence* 91.2 (1997), pp. 183–203.

[61] Neil R Smalheiser. "Literature-based discovery: Beyond the ABCs". In: *Journal of the American Society for Information Science and Technology* 63.2 (2012), pp. 218–224.

[62] Meliha Yetisgen-Yildiz and Wanda Pratt. "A new evaluation methodology for literature-based discovery systems". In: *Journal of Biomedical Informatics* 42.4 (2009), pp. 633–643.

[63] Jonathan D Wren. "Extending the mutual information measure to rank inferred literature relationships". In: *BMC bioinformatics* 5.1 (2004), p. 1.

[64] Don R Swanson, Neil R Smalheiser, and Vetle I Torvik. "Ranking indirect connections in literature-based discovery: The role of medical subject headings". In: *Journal of the American Society for Information Science and Technology* 57.11 (2006), pp. 1427–1439.

[65] Trevor Cohen et al. "Logical Leaps and Quantum Connectives: Forging Paths through Predication Space." In: *AAAI Fall Symposium: Quantum Informatics for Cognitive, Social, and Semantic Processes*. 2010.

[66] Dimitar Hristovski et al. "Exploiting semantic relations for literature-based discovery". In: *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*. 2006.

[67] Peter Bruza et al. "Towards operational abduction from a cognitive perspective". In: *Logic Journal of IGPL* 14.2 (2006), pp. 161–177.

[68] Trevor Cohen, Roger W Schvaneveldt, and Thomas C Rindflesch. "Predication-based semantic indexing: permutations as a means to encode predications in semantic space." In: *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*. 2009.

[69] Trevor Cohen et al. "Finding Schizophrenia's prozac emergent relational similarity in predication space". In: *International Symposium on Quantum Interaction*. Springer. 2011, pp. 48–59.

[70] Ronald N Kostoff et al. "Literature-related discovery (LRD): Methodology". In: *Technological Forecasting and Social Change* 75.2 (2008), pp. 186–202.

[71] T Elizabeth Workman et al. "Spark, an application based on Serendipitous Knowledge Discovery". In: *Journal of Biomedical Informatics* 60 (2016), pp. 23–37.

[72] Vetle I Torvik and Neil R Smalheiser. "A quantitative model for linking two disparate sets of articles in MEDLINE". In: *Bioinformatics* 23.13 (2007), pp. 1658–1665.

[73] Trevor Cohen et al. "Discovering discovery patterns with predication-based semantic indexing". In: *Journal of Biomedical Informatics* 45.6 (2012), pp. 1049–1065.

[74] Wanda Pratt and Meliha Yetisgen-Yildiz. "LitLinker: capturing connections across the biomedical literature". In: *Proceedings of the 2nd international conference on Knowledge capture*. ACM. 2003, pp. 105–112.

[75] Judita Preiss, Mark Stevenson, and Robert Gaizauskas. "Exploring relation types for literature-based discovery". In: *Journal of the American Medical Informatics Association* (2015), pp. 987–992.

[76] Bartłomiej Wilkowski et al. "Graph-based methods for discovery browsing with semantic predications". In: *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*. Vol. 2011. American Medical Informatics Association. 2011, p. 1514.

[77] Judita Preiss and Mark Stevenson. "The Effect of Word Sense Disambiguation Accuracy on Literature Based Discovery". In: *Proceedings of the ACM Ninth International Workshop on Data and Text Mining in Biomedical Informatics*. ACM. 2015, p. 57.

[78] Xiaohua Hu et al. "A Semantic Approach for Mining Hidden Links from Complementary and Non-interactive Biomedical Literature". In: *SDM*. SIAM. 2006, pp. 200–209.

[79] Meliha Yetisgen-Yildiz and Wanda Pratt. "Using statistical and knowledge-based approaches for literature-based discovery". In: *Journal of Biomedical Informatics* 39.6 (2006), pp. 600–611.

[80] Michael D Gordon and Robert K Lindsay. "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". In: *Journal of the American Society for Information Science* 47.2 (1996), pp. 116–128.

[81] Robert K Lindsay and Michael D Gordon. "Literature-based discovery by lexical statistics". In: *Journal of the Association for Information Science and Technology* 50.7 (1999), p. 574.

[82] Dimitar Hristovski et al. "Using literature-based discovery to identify disease candidate genes". In: *International Journal of Medical Informatics* 74.2 (2005), pp. 289–298.

[83] Michael D Gordon and Susan Dumais. "Using latent semantic indexing for literature based discovery". In: *Journal of the American Society for Information Science* 49.8 (1998), pp. 674–685.

[84] Peter Bruza, Dawei Song, and Robert McArthur. "Abduction in semantic space: Towards a logic of discovery". In: *Logic Journal of IGPL* 12.2 (2004), pp. 97–109.

[85] Lauri Eronen and Hannu Toivonen. "Biomine: predicting links between biological entities using network models of heterogeneous databases". In: *BMC bioinformatics* 13.1 (2012), p. 119.

[86] Justin Sybrandt, Michael Shtutman, and Ilya Safro. "MOLIERE: Automatic Biomedical Hypothesis Generation System". In: *Proceedings of the 23rd ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. 2017, pp. 1633–1642.

[87] Justin Sybrandt and Ilya Safro. "Validation and Topic-driven Ranking for Biomedical Hypothesis Generation Systems". In: *bioRxiv* (2018). DOI: 10.1101/263897. eprint: https://www.biorxiv.org/content/early/2018/02/11/263897.full.pdf. URL: https://www.biorxiv.org/content/early/2018/02/11/263897.

[88] Armand Joulin et al. "Fasttext. zip: Compressing text classification models". In: *arXiv preprint arXiv:1612.03651* (2016).

[89] Andrej Kastrin, Thomas C Rindflesch, Dimitar Hristovski, et al. "Link prediction on a network of co-occurring mesh terms: towards literature-based discovery". In: *Methods of information in medicine* 55.4 (2016), pp. 340–346.

[90] Trevor Cohen et al. "EpiphaNet: An interactive tool to support biomedical discoveries". In: *Journal of biomedical discovery and collaboration* 5 (2010), pp. 21–49.

[91] Ingrid Petrič et al. "Literature mining method RaJoLink for uncovering relations between biomedical concepts". In: *Journal of Biomedical Informatics* 42.2 (2009), pp. 219–227.

[92] Ingrid Petrič et al. "Bisociative knowledge discovery by literature outlier detection". In: *Bisociative Knowledge Discovery*. Springer, 2012, pp. 313–324.

[93] Delroy Cameron et al. "A graph-based recovery and decomposition of Swanson's hypothesis using semantic predications". In: *Journal of Biomedical Informatics* 46.2 (2013), pp. 238–251.

[94] Neil R Smalheiser. "The Arrowsmith project: 2005 status report". In: *International Conference on Discovery Science*. Springer. 2005, pp. 26–43.

[95] Neil R Smalheiser et al. "Collaborative development of the Arrowsmith two node search interface designed for laboratory investigators". In: *Journal of biomedical discovery and collaboration* 1.1 (2006), p. 8.

[96] Ralph A DiGiacomo, Joel M Kremer, and Dhiraj M Shah. "Fish-oil dietary supplementation in patients with Raynaud's phenomenon: a double-blind, controlled, prospective study". In: *The American journal of medicine* 86.2 (1989), pp. 158–164.

[97] Dimitar Hristovski et al. "Supporting discovery in medicine by association rule mining of bibliographic databases". In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer. 2000, pp. 446–451.

[98] Neil R Smalheiser. "Rediscovering don swanson: The past, present and future of literature-based discovery". In: *Journal of Data and Information Science* 2.4 (2017), pp. 43–64.

[99] Takaya Saito and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS one* 10.3 (2015), e0118432.

[100] J Caleb Goodwin, Trevor Cohen, and Thomas Rindflesch. "Discovery by scent: Discovery browsing system based on the Information Foraging Theory". In: *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference*. IEEE. 2012, pp. 232–239.

[101] T Elizabeth Workman et al. "Framing serendipitous information-seeking behavior for facilitating literature-based discovery: A proposed model". In: *Jour-*

nal of the Association for Information Science and Technology 65.3 (2014), pp. 501–512.

[102] Wen-tau Yih and Vahed Qazvinian. "Measuring word relatedness using heterogeneous vector space models". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2012, pp. 616–620.

[103] Joseph Reisinger and Raymond J Mooney. "Multi-prototype vector-space models of word meaning". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 109–117.

[104] Kira Radinsky et al. "A word at a time: computing word relatedness using temporal semantic analysis". In: *Proceedings of the 20th international conference on World wide web*. ACM. 2011, pp. 337–346.

[105] TH Muneeb, Sunil Kumar Sahu, and Ashish Anand. "Evaluating distributed word representations for capturing semantics of biomedical concepts". In: *Proceedings of ACL-IJCNLP* (2015), p. 158.

[106] B. Chiu et al. "How to Train Good Word Embeddings for Biomedical NLP". In: *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. 2016, pp. 166–174.

[107] S.V. Pakhomov et al. "Corpus domain effects on distributional semantic modeling of medical terms". In: *Bioinformatics* 32 (2016), pp. 3635–3644.

[108] A. Sajadi et al. “Domain-Specific Semantic Relatedness from Wikipedia Structure: A Case Study in Biomedical Text”. In: *Computational Linguistics and Intelligent Text Processing*. Vol. 9041. Springer International Publishing, 2015, pp. 347–360.

[109] Zhiguo Yu et al. “Retrofitting Word Vectors of MeSH Terms to Improve Semantic Similarity Measures”. In: *EMNLP 2016* (2016), p. 43.

[110] T Elizabeth Workman et al. “A literature-based assessment of concept pairs as a measure of semantic relatedness”. In: *AMIA Annual Symposium Proceedings*. Vol. 2013. American Medical Informatics Association. 2013, p. 1512.

[111] B.T. McInnes, T. Pedersen, and S.V. Pakhomov. “UMLS-Interface and UMLS-Similarity : Open Source Software for Measuring Paths and Semantic Similarity”. In: *Proceedings of the American Medical Informatics Association (AMIA) Symposium*. San Fransico, CA, Nov. 2009.

[112] M. Lesk. “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone”. In: *Proceedings of the 5th Annual International Conference on Systems Documentation*. 1986, pp. 24–26.

[113] S. Patwardhan and T. Pedersen. “Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts”. In: *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*. Trento, Italy, Apr. 2006, pp. 1–8.

[114] S.V.S. Pakhomov et al. “Towards a Framework for Developing Semantic Relatedness Reference Standards”. In: *Journal of Biomedical Informatics* 44.2 (2011), pp. 251–265.

[115] Bridget T McInnes and Ted Pedersen. "Evaluating semantic similarity and relatedness over the semantic grouping of clinical term pairs". In: *Journal of biomedical informatics* 54 (2015), pp. 329–336.

[116] Meliha Yetisgen-Yildiz and Wanda Pratt. "Evaluation of literature-based discovery systems". In: *Literature-based discovery*. Springer, 2008, pp. 101–113.

[117] Sam Henry, Clint Cuffy, and Bridget T McInnes. "Vector representations of multi-word terms for semantic relatedness". In: *Journal of biomedical informatics* 77 (2018), pp. 111–119.

[118] Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. "Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1". In: *Journal of biomedical informatics* 58 (2015), S11–S19.

[119] Özlem Uzuner et al. "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 552–556.

[120] Martin Krallinger et al. "The CHEMDNER corpus of chemicals and drugs and its annotation principles". In: *Journal of cheminformatics* 7.1 (2015), S2.

[121] Chih-Hsuan Wei et al. "Overview of the BioCreative V chemical disease relation (CDR) task". In: *Proceedings of the fifth BioCreative challenge evaluation workshop*. 2015, pp. 154–166.

[122] Ying Zhao, George Karypis, and Usama Fayyad. "Hierarchical clustering algorithms for document datasets". In: *Data mining and knowledge discovery* 10.2 (2005), pp. 141–168.

[123] Dimitar Hristovski, Thomas Rindflesch, and Borut Peterlin. "Using literature-based discovery to identify novel therapeutic approaches". In: *Cardiovascular & Hematological Agents in Medicinal Chemistry (Formerly Current Medicinal Chemistry-Cardiovascular & Hematological Agents)* 11.1 (2013), pp. 14–24.

[124] Dimitar Hristovski, Andrej Kastrin, and Thomas C Rindflesch. "Semantics-based cross-domain collaboration recommendation in the life sciences: Preliminary results". In: *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2015, pp. 805–806.