BIOMEDICAL TEXT MINING FROM CONTEXT

A DISSERTATION

SUBMITTED TO THE PROGRAM IN BIOMEDICAL INFORMATICS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Bethany L. Percha

December 2016

This dissertation is online at: http://purl.stanford.edu/ft877bj7559

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Russ Altman, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Art Owen

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Christopher Potts

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Many profound insights from biomedical research and clinical practice remain hidden within the unstructured text of scientific articles and electronic medical records. Extracting structured information from biomedical text could dramatically accelerate the pace of biomedical research, but due to the high variability of natural language, it hinges on our ability to recognize when different-looking statements are saying the same thing. Unfortunately, attempts to address this problem in the biomedical domain usually involve structured lexicons and ontologies, which are expensive and time-consuming to produce. In recent years, a subdomain of natural language processing called distributional semantics has approached normalization in a different way: by learning mathematical representations of words, phrases, and relationships based on their usage patterns in large corpora. These methods can detect that two different strings are semantically related based on how they are used in context, and require little or no human effort.

This dissertation illustrates how distributional approaches can be applied to several important biomedical text mining tasks, including gene, drug and disease name normalization, ontology building, and the construction of a structured radiology lexicon from clinical notes. I describe a novel distributional algorithm (EBC) for extracting relationships among biomedical entities, such as chemicals, genes and diseases, and show how it can be applied to learn the structure of chemical-gene, chemical-disease, gene-disease, and gene-gene relationships from contextual usage patterns. Finally, I apply distributional relationship extraction to two inferential tasks: curating pharmacogenomic pathways, and uncovering the mechanisms behind drug-drug interactions.

Acknowledgments

I have never felt more indebted to more people than I have writing this dissertation. The past six years have seen many ups and downs, promising results and failed experiments, shifts of focus, waxing and waning of my belief in what I was doing, and one significant leave of absence to join a startup. Were it not for the people in my life who consistently welcomed me into their homes, helped me through tough times, made me laugh, and shared their wisdom and experience with me, I never would have made it through. Thank you to everyone listed here, and to everyone I'm sure I forgot. You will all be in my heart forever.

First, to my adviser, Russ Altman: thank you for your unsurpassed commitment, both in time and energy, to me and to all of your students. I have never met another professor, at Stanford or elsewhere, who takes this role as seriously as you do. Thank you for opening my eyes to new opportunities, for all of the late night meetings at Cafe Borrone, and for giving me the space and freedom to pursue my own ideas. You were always able to find ways to help me along on my scientific journey, even though my goals were constantly evolving and, in many cases, different from your own. I can think of no better definition of the word "mentor". I will miss you.

To my committee, Nigam Shah, Curt Langlotz, Pentti Kanerva, Art Owen, and Chris Potts: thank you for all of your wisdom and guidance over the years. Thank you to Nigam for serving as part-time psychotherapist during the more personally challenging parts of my PhD, for giving me the opportunity to help design and teach a new course at Stanford, and for always focusing on practical utility and not being afraid to ask the hard questions. Thanks to Curt for teaching me about radiology and helping me expand the scope of my work to clinical text. Every time I enter that tiny,

windowless hospital office, I know I'm going to laugh and come away more excited about NLP than when I came in; I'm excited to continue our collaboration this year. Pentti, I'm so glad that Trevor introduced us back in 2013. Your work on random indexing inspired me before I knew you, and is one of the reasons I became interested in distributional semantics. Thank you also for your friendship, for the many dinners with Anne, and for your kindness toward my family. Art, thanks for the lunches and the technical advice on EBC, as well as for sharing your life experiences in academia with me. I now understand why you're such a highly sought-after committee member in BMI. :) Chris, CS224U was one of my favorite classes at Stanford, and was where I first got to explore distributional semantics ideas in more detail. It's been incredibly valuable to have a linguist on board with this project, even if I did run out of time before I could fix all the Oxford commas in this dissertation (forgive me). Thank you for all of your advice and help over the years. I have learned so much from you.

Art and Chris: a special thank you for serving as readers for this dissertation. The end of my PhD was marked by a great personal tragedy, as my beloved grandmother ("Grams") passed away two weeks before the dissertation deadline. Russ, Art and Chris made it possible for me to spend her last days with her, a gift it would be impossible to repay. Thank you all so much.

To the Helix Group, past and present, especially Emily Mallory, Emily Flynn, Stefano Rensi, Roxana Daneshjou, Olga Sazonova, Binbin Chen, Weizhuang Zhou, Grace Tang, Assaf Gottlieb, Lichy Han, Yong Li, Tianyun Liu, Wen Torng, Nick Tatonetti, Yuhao Zhang, Sara Hillenmeyer, and Yael Garten: thanks for creating such a welcoming and fun lab environment. I would also like to thank the staff of PharmGKB, especially Teri Klein, Michelle Carrillo, Mark Woon and Ryan Whaley, for all of their help and technical advice.

Special thanks go to my PhD partner-in-crime, Sara Hillenmeyer, without whose example I never would have completed a single form or milestone. There is a reason all my talks were exactly one month after hers.

Mary Jeanne Oliva, Nancy Lennartsson, Steve Bagley, Larry Fagan, and the students and staff of the BMI program have created the most unique and supportive academic environment I have ever experienced. For the past six years, I have felt like

I was part of an extended family. I wish you all the best of luck, and look forward to seeing you again over the years.

Before coming to Stanford, I was fortunate to learn from some wise and patient mentors at Walled Lake Central and the University of Michigan. Thanks especially to David Darnton, Michal Zochowski, Carl Simon, Mark Newman, Betsy Foxman, and Joe Eisenberg. Thanks to Chris Manning, Percy Liang, Dan Jurafsky and all the members of the Stanford NLP group for graciously allowing me to sit in on their group meetings for several years. Thanks to Daniel Rubin for supervising my first rotation at Stanford, and for all of his help with the radiology portion of this thesis.

My PhD was supported financially by a Morgridge Family BioX Stanford Interdisciplinary Graduate Fellowship, as well as a grant from Oracle Corporation and a training grant from the National Library of Medicine. I am grateful and deeply humbled that so many people believed in my research enough to pay for it.

One of my most valuable experiences during my PhD years was consulting for several startups. Thank you to the teams at ElationEMR, SmartPatients, Kyron, DigiSight and HealthPals for all you have taught me. Special thanks to Roni Zeiger, Nitin Karandikar, Alison Polkinhorne, Sushant Shankar, and Kunal Sethy for their words of wisdom on business and life.

In the final year of my PhD, I participated in the Accel Innovation Scholars program at Stanford, which I cannot recommend highly enough. Thanks to Tina Seelig and Fern Mandelbaum, and to my fellow scholars (and friends) Natalie Burkhard, Chen-Ming Chang, Liliana De La Paz, Henning Roedel, and Joy Xiang.

To my friends, in Michigan, D.C., California and elsewhere: I value you more than you could ever know. Whether I see you once a week or once a year, I am so grateful to have you in my life. Thanks to: Jenn Lahti Andersen, Ed Baskerville and Robin Goldman, Elsa Birch and Alex Morgan, Selen Bozkurt, Rob Bruggner, Sarah Cherng, Ryan and Elizabeth Conroy, Roxana Daneshjou and Omid Azizi, Sumon Dantiki and Kate Hadley, Nathan Diehl, Lynn Eckert and Andrea Knittel, Nik Engineer, Bill Ferrell, Francisco Gimenez and Lisa Fugere, Rachel Goldfeder, Shama Cash Goldwasser, Peyton Greenside, Darla and Mike Hewett, Ken Jung, Anne and Pentti Kanerva, Peter Kang, Daniel Kim, Yi Liu and Linda Fang, Elizabeth

Leicht and Michael Busha, Brian Maples and Erika Strandberg, Sara Hillenmeyer and Casey Marks, Emily Mallory, Greg McInnes and Christina Tsutsumi, Catherine Morris and Jarryd Smith, Jonathan Mortensen, David Moskowitz and Sarah Kim, Evan and Breanne Minty, Laura Neville, Jon Palma, Marshall Pierce, Katie Planey, Tanya Podchiyska, Sarah Poole, David Poznik and Emily Tsang, Olga Sazonova and Noah Goodman, Sushant Shankar and Julia Ling, Ariel Shaw and Brian Karrer, Greg Valiant and Janara Christensen, Kerrie and Jon VerLee, Stephanie Wang, Luke Yancy, Darvin Yi, Marek Zapletal, and Noah and Veronica Zimmerman.

My deepest thanks and love go to my family, especially Grams, Aunt Gail and Uncle Don, Uncle Rick and Aunt Cheryl, Alyssa, Alexis, Ryan, Uncle Russ and Aunt Lynn, Cody, Uncle Dave and Aunt Laura, Miranda, David, and my wonderful extended family in west Michigan. Thanks especially to Aunt Gail and to Grams for being such a huge help to my parents and brother while I've been in California.

Finally, to Mom, Dad and Steve: even though you live far away, you have been with me every step of the way. Thank you for your constant belief in me throughout my life. Your love, support and encouragement has sustained me through everything I've ever done, and everything I've tried and failed to do. Thank you for teaching me, right from the beginning, what it means to do my best. I love you.

For my grandfather
Royal N. Perkins
(1930-2004)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Biomedical research generates text at an incredible rate. Each year, several hundred thousand new articles enter Medline from over 5,500 unique journals (Figure 1.1) [98, 99]. The literature's rapid growth and the rise of interdisciplinary domains like bioinformatics and systems biology are changing how the scientific community interacts with this important resource. Knowledge bases like OMIM [40], DrugBank [152] and PharmGKB [150] manually curate and restructure information from the literature to increase its accessibility to researchers and clinicians. These knowledge bases capture cross-sectional "slices" of the literature, drawing connections among facts reported in different journals, at different times, and in different research domains. Often, they examine the literature in ways not easily captured by current indexing strategies, such as MeSH terms or key words.

As the literature grows and the information we need to extract increases in complexity, full manual curation of these knowledge bases is rapidly becoming infeasible. Progress in natural language processing (NLP) has encouraged the development of automated and semi-automated methods for enabling more efficient curation of biomedical text [50, 81, 134], especially as biomedical research begins to explore even larger text-based resources, such as electronic medical records (EMRs) [57, 96].

However, tasks that are simple for human readers, such as recognizing when two different-looking statements mean the same thing, or when one statement is a more general version of another statement, are often extremely challenging for NLP

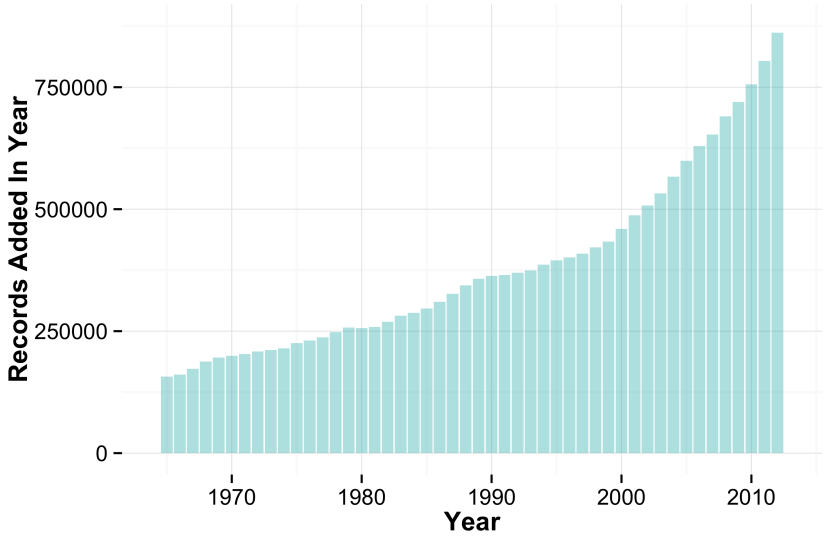


Figure 1.1. New citations added to Medline per year, 1965-2012. Data from the National Library of Medicine.

algorithms. Recognizing when two different strings map to the same concept (at some defined level of granularity) is sometimes called "normalization". In the biomedical domain, normalization is usually accomplished using structured lexicons and ontologies, which are difficult and time-consuming to produce and do not translate easily to new domains.

One way around the problem of normalization, and one that has been explored extensively in the computer science and linguistics literature, is to infer whether two segments of text (words, phrases, sentences, etc.) are semantically related by examining similarities and differences in their usage patterns over large, unlabeled text corpora. Methodologically, these approaches fall under an umbrella term: "distributional semantics". If two statements are used in similar contexts, this idea goes, they are likely to be semantically related. If distributional approaches are successful, they could obviate or greatly reduce the need for manually curated lexicons and ontologies.

1.1 Research Hypothesis and Specific Aims

Hypothesis Distributional techniques can partially or completely replace human-curated lexicons and ontologies for normalization in biomedical text mining.

Specific Aims

1. Develop distributional methods that extract and normalize biomedical entities and relationships in text using minimal labeled training data. In cases where existing distributional techniques may apply, evaluate them on relevant biomedical tasks.
2. Evaluate these methods against human curated data from PharmGKB, DrugBank, and other biomedical databases to assess their potential for assisting curation.
3. Apply to four related tasks:
   (a) Constructing more complete lexicons of drugs, genes, and phenotypes
   (b) Expanding PharmGKB's coverage of drug-gene, gene-gene and gene-phenotype relationships
   (c) Predicting new drug-drug interactions based on mechanisms of drug action as described in text
   (d) Automating (or partially automating) the curation of pharmacogenomic pathways from research articles

1.2 Summary of the Dissertation

Chapter 2 describes a simple technique for learning whether two drugs are likely to interact based on descriptions of their interactions with genes. The method depends on PHARE, a manually constructed ontology, for normalization. The limitations of this work, discussed at the end of the chapter, inspired the rest of this dissertation. Chapter 2 also provides important background information on drug-drug interactions (DDIs) and why text mining might be relevant for predicting them.

Chapter 3 is a survey of the research field of distributional semantics: how, from

an algorithmic standpoint, do we efficiently learn how words, phrases and relationships are similar, based only on their contextual usage patterns?

In Chapter 4, I show how the methods from Chapter 3 can be applied to expand the coverage of an existing biomedical lexicon, normalize disease, drug, and gene names (with varying degrees of success), and build a domain specific lexicon of radiology terms based on their usage in clinical notes.

Chapter 5 introduces a new algorithm, Ensemble Biclustering for Classification (EBC) that is similar to the algorithms from Chapter 3 but addresses some of their shortcomings, specifically for the task of relationship extraction. Relationship extraction has taken a backseat to word and phrase similarity assessments in the distributional semantics literature, but it is a particularly important task for the curation of biomedical knowledge bases.

Chapter 6 applies EBC to the task of drug-gene relationship extraction from Medline text. I describe how EBC can be used to expand existing knowledge bases (DrugBank and PharmGKB), and how even in the absence of any training data, EBC can be applied in an unsupervised manner to learn the structure of drug-gene relationships entirely from contextual usage patterns.

In Chapter 7, I show how unsupervised EBC can be applied to learn relationship classes of four different types - chemical-gene, chemical-disease, gene-disease, and gene-gene. I build an annotated network of relationship types and show how its structure can be used to infer new biomedical relationships from old ones.

Chapters 8 and 9 describe the application of EBC, the relationship classes learned in Chapter 7, and the ideas about inference discussed therein to two important biomedical problems: the automated curation of pharmacogenomic pathways from text, and the mechanistic explanation of drug-drug interactions.

Chapter 2

A Motivating Example: Drug-Drug Interactions

Drug-drug interactions (DDIs) are an emerging threat to public health. Recent estimates indicate that DDIs cause hundreds of thousands of emergency room visits and hospitalizations each year in the United States. Current approaches to DDI discovery, which include Phase IV clinical trials and post-marketing surveillance, are insufficient for detecting many DDIs and do not alert the public to potentially dangerous DDIs before a drug enters the market.

Here we describe a novel approach to the early detection of DDIs that is based on mining the biomedical literature for drug-gene interactions and connecting them in series to predict drug-drug interactions. The method is simple, predicts DDIs with high accuracy, and could be applied to perform surveillance of the biomedical literature for likely DDIs in real time, as new discoveries about drugs, genes, and their relationships are made. Unfortunately it also illustrates many of the pitfalls associated with information extraction from biomedical text – pitfalls that inspired the rest of this dissertation.

The text in this chapter is borrowed mainly from our published paper [107]. Much of the background information about DDIs is taken from our subsequent review article [105].

2.1 Drug-Drug Interactions: Incidence and Impact

In 2007, a meta-analysis of 23 clinical studies from around the world revealed that drug-drug interactions (DDIs) cause approximately 0.054% of emergency room visits, 0.57% of hospital admissions, and 0.12% of rehospitalizations [3]. There are 136.1 million emergency room visits [32] and 34.1 million hospital discharges [39] in the USA alone each year. If these percentages are correct, Americans experience DDI events serious enough to send them to the emergency room almost 74,000 times per year, and hospitals admit nearly 195,000 patients per year because of DDIs. Unsurprisingly, DDIs also contribute to increased cost and duration of hospital stays [94].

We should expect DDI incidence to increase as the simultaneous use of multiple drugs becomes more common. According to the Centers for Disease Control (CDC), the percentage of the US population taking at least one prescription drug within the last 30 days increased from 39.1% in 1988-1994 to 47.5% in 2007-2010. During that same period, the percentage of Americans taking three or more prescription drugs rose from 11.8% to 20.8%, and the percentage taking five or more drugs increased from 4.0% to 10.1% (Figure 2.1a) [33]. Polypharmacy is particularly common among the elderly, making them especially susceptible to DDIs (Figure 2.1b). In the 2007 study described above, DDIs caused 4.8% of hospital admissions among the elderly, increasing their risk nearly 8.5-fold relative to the general population.

2.2 Why DDIs are Difficult to Study

Many known DDIs involve common medications such as antihypertensives, anti-inflammatories, and anticoagulants (Table 2.1), so a reasonable question is why so many DDIs go undetected for so long. Drugs have occasionally been pulled from the market because of DDIs, but even in such cases the drugs were usually available to the public for years before withdrawal (see [10] and [116]). Although in-vitro laboratory studies, such as DDI assays, can help to alert drug manufacturers and the scientific community to the presence of new DDIs, the difficulty of recognizing DDIs in the clinic, the dose dependence of many DDIs, the nature of the drug approval process,

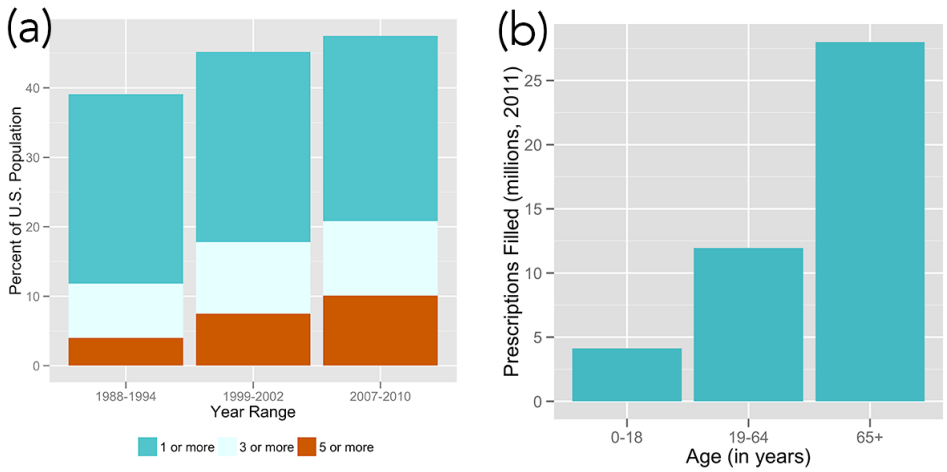


Figure 2.1. (a) Number of prescription drugs used in the past 30 days by percentage of the USA population (age-adjusted estimates). Source: National Center for Health Statistics. Health, United States, 2011: With Special Feature on Socioeconomic Status and Health. Hyattsville, MD. 2012. Table 99: Prescription drug use in the past 30 days, by sex, age, race, and Hispanic origin: United States, selected years 1988-1994 through 2007-2010. (b) Average number of prescriptions filled in 2011 in the USA by age. The data include both new prescriptions and refills, brand name, and generic drugs. Source: The Kaiser Family Foundation, statehealthfacts.org, accessed September 14, 2012. Data source: SDI Health, LLC: Special Data Request, 2012. Calculations based on 2011 population estimates from the US Census Bureau.

Table 2.1. Examples of known drug interactions. Mibefradil and astemizole were pulled from the market because of their interactions with other drugs, even though both were considered safe on their own.

Year reported	Drug combination	Effect	Refs
1990	Ketoconazole Terfenadine	Ventricular arrhythmias	[92]
1993	Astemizole CYP3A4 inhibitors (e.g. grapefruit juice)	Prolonged QTc interval, arrhythmias	[10]
1997	Sorivudine Fluorouracil	Fatal toxicity	[100]
1998	Mibefradil Various cardioactive drugs including β-blockers and statins	Bradycardia, rhabdomyolysis	[116, 128]
1998	Clozapine Fluoxetine	Death due to depression of nervous, respiratory, and cardiovascular systems	[30]
2005	Clarithromycin Colchicine	Death in patients with renal insufficiency	[49]
2008	Warfarin Antibiotics	Gastrointestinal bleeding	[127]
2008	Propofol, methylprednisolone, cyclosporine Colchicine, simvastatin	Fatal toxic myopathy	[34]
2011	Pravastatin Paroxetine	Increased blood glucose	[143]

and natural genetic and demographic variation can all delay DDI recognition.

For example, we cannot realistically expect practicing physicians to notice and document most DDIs on their own. Patients who take multiple drugs are often afflicted with multiple comorbidities, and it is difficult to determine whether adverse events are the result of side effects from a single drug, interactions between two or more drugs, or exacerbations of the patient's underlying disease(s). In addition, the number of patients on a particular drug combination, especially within a single practice or hospital, may be small, preventing physicians from recognizing patterns of interactions within their own patient cohorts. Some DDIs are also dose-dependent, which means that a DDI may be unrecognizable unless a patient is dosed at the high end of the approved range for one or both drugs.

In addition, DDIs are often difficult to observe within the environment of a pre-market clinical trial. Phase III clinical trials, the last stage of investigation before a

drug enters the market, usually enroll only 1000-3000 individuals. If the test agent interacts with a drug that is not typically prescribed among members of the study population, or if the interaction is weak or rare, few if any study subjects will experience DDI-related symptoms over the course of the study. It comes as no surprise, therefore, that the DDIs we know about are those that cause the most consistent and serious side effects, or those that occur between very commonly prescribed classes of drugs.

Finally, natural human variability can potentially mask the effects of many DDIs. We already know that individuals with particular genetic complements are more sensitive to the effects of certain drugs [17, 136] and have a greater chance of experiencing adverse side effects [97] than others. We might therefore expect DDIs to appear more frequently among certain genetic subpopulations, such as people with transporter gene mutations, for example, or those who are "fast" or "slow" metabolizers. We also know that demographic variation – differences in age, gender, race/ethnicity, weight, and height, among other factors – explains much of the variation in dosing requirements for many drugs [13, 17, 153], so we might expect those factors to confound DDI detection as well. The US Food and Drug Administration (FDA) regularly publishes documents containing advice about the design of DDI studies that addresses many of these complicating factors, summaries of which can be found in [48] and [156].

2.3 Mechanistic Prediction of DDIs from Text

Biologically, many DDIs are the result of conflicting or synergistic interactions between a pair of drugs and similar genes or molecular pathways within the human body [2, 58]. Therefore, what we observe as drug-drug interactions might often be considered drug-gene-drug interactions. Unfortunately, while lists of known DDIs are widely available and commonly used in clinical practice, drug-gene interactions are not as widely known. Genes and drugs can interact in a variety of ways, and it is unclear which interaction types are most predictive of a drug's tendency to interact with other drugs. No complete databases exist that concisely describe the exact mechanisms by which drugs and genes interact; most of these interactions are only described in papers buried deep within the scientific literature.

For these reasons, text mining presents an intriguing possible solution to the problem of uncovering novel DDIs [15, 37, 109]. A growing body of research has sought to classify DDIs and better understand gene-drug relationships using text mining; for example, Tari *et al* [142] developed a method that combined text mining and automated reasoning to extract novel DDIs. Other authors have built text-based networks of biological entities and used reasoning techniques to uncover new biologically relevant relationships among them [12, 108]. Previous work from our own group [18, 20] has established methods for using a syntactical parser to identify and characterize drug-gene relationships. The end result was a semantic network of drug-gene relationships in which the edges consisted of several hundred interaction types and subject/object context terms normalized to concepts in an ontology. All of these approaches have sought to infer novel relationships among biological entities by combining known facts expressed in scientific text.

The work described here extends this line of research by using our semantic network - in particular, paths through the network that connect pairs of drugs - to infer the types of drug-gene relationships that can predict drug-drug interactions. An advantage of our method is the fact that it makes almost no a priori assumptions about the nature of these relationships, instead using a machine learning algorithm (a random forest) to identify the kinds of gene-drug relationships that best predict DDIs. Besides learning which textual features are most relevant for predicting DDIs, the method can also be used to predict novel DDIs and to explain these predictions through suggested mechanisms of interaction.

2.3.1 Methods: Extraction Pipeline for Drug-Gene Relations

This project built on an earlier method for text mining Medline to extract drug-gene interactions [20]. That method worked as follows:

1. *Create two lexicons of terms, one for gene names and one for drug names.* We used two custom lexicons. The first consisted of a set of 731 known pharmacodynamic and pharmacokinetic genes identified by the PharmGKB database curators [62]. The second consisted of 2,910 unique drug and drug-class names,

also from PharmGKB. The gene lexicon also included all common synonyms for each gene; we required the drug name to be in its generic form (rather than a brand name) to be included[1].

2. *Obtain a corpus of Medline article abstracts.* The Helix Group at Stanford University maintained a corpus of all Medline abstracts published before 2009[2]. The corpus contained about 17.5 million abstracts and 88 million sentences.
3. *Retrieve all sentences in Medline that mention both a drug and a gene of interest.* The drug and gene entities of interest are known as *seeds*. We accomplished this using the two aforementioned lexicons and running 100 search processes in parallel on Stanford's BioX2 cluster[3].
4. *Represent sentences as dependency graphs using the Stanford Parser [61].* The dependency graphs are rooted, oriented, and labeled graphs, where the nodes are words and the edges are grammatical relationships between words [23]. If two seeds were not located in the same sentence clause, that sentence was removed from consideration. In addition, if a graph contained more than one clause and there was a clause that did not contain either seed, that clause was removed from consideration. A sample dependency graph for one Medline sentence of interest is shown in Figure 2.2.
5. *Identify and normalize composite entities.* A seed does not usually occur in isolation in a sentence, but as part of a larger composite entity that includes the surrounding context. For example, a gene name like CYP3A4 will usually occur as part of a larger entity, such as *CYP3A4 degradation* or *CYP3A4 elimination*. We used a previously-established algorithm [18] to identify the context terms surrounding each seed and normalize them. The normalization process involved mapping context terms with similar semantics but different syntax, such as *degradation of CYP3A4* and *CYP3A4 degradation*, to the same concept (*Elimination*) using a previously-constructed ontology [20].

[1]Note that throughout this analysis, we use the term "gene" interchangeably with "gene product" or "protein"; it is actually the protein product of a gene that interacts with a drug.

[2]Work on this project began in 2010, which is why this corpus is so old.

[3]The Bio-X2 cluster at Stanford University was funded by the National Science Foundation, NSF award CNS-0619926.

6. *Extract relations between composite entities.* Relations describe the nature of the interaction between the two entities in a given sentence. They take the form $R(a, b)$, where a and b represent the locations of the two entities in the dependency graph, and R is a node that connects a and b and indicates the nature of their relationship. For a sentence to progress past this stage of the analysis, the relation connecting the gene and drug entities must have been a verb (e.g. associated) or a nominalized verb (e.g. association).
7. *Normalize relations.* The extracted relations, like the context terms surrounding each seed, were normalized. During normalization, the raw relations were mapped onto a much smaller set of normalized relationships taken from the ontology. For example, the verbs *associated* and *related* both map to the ontological entity *isAssociatedWith*. In addition, less-common terms like *augment* were mapped to more common synonyms, like *increase*.

The overall goal of the normalization process for both composite entities and relations was to collapse statements with the same semantic meaning but different word choice or syntax to the same basic relationship, reducing the number of features that needed to be considered later when building the DDI classifier. When tested on a smaller set of drug-gene relationships extracted from Medline, our ontology was able to properly normalize approximately 80% of all relation types mentioned in the literature. Nonetheless, by including only those sentences where the relation could be normalized, we necessarily excluded some true facts about drug-gene relationships from our network. It is important to note that only sentences for which the relation could not be normalized were thrown out; sentences for which the context terms could not be normalized were still included - the context was simply normalized to *Thing*, as described further in Section 2.3.4.

2.3.2 Methods: Creating a Semantic Network

When applied to the entire Medline 2009 corpus, the relation extraction and normalization process yielded 76,784 different normalized drug-gene relationships of the form shown in Figure 2.3. We eliminated all relationships in which the verb could not be

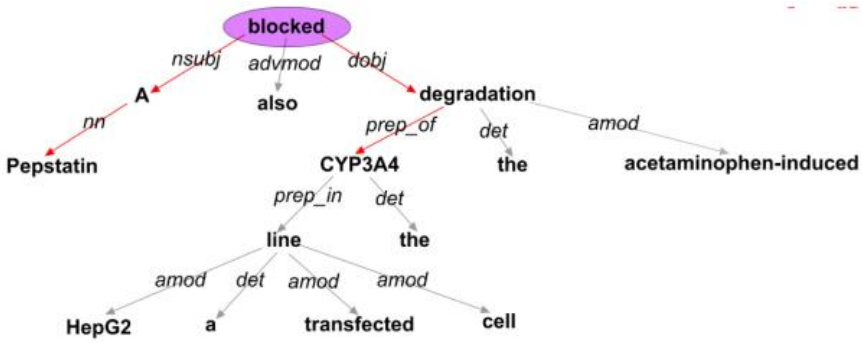


Figure 2.2. Dependency graph for the sentence "Pepstatin A also blocked the acetaminophen-induced degradation of the CYP3A4 in a transfected HepG2 cell line" (PMID: 15078344). The red arrows show the path through the graph that connects the seeds Pepstatin A (a drug) and CYP3A4 (a gene). Because this path contains a verb - in this case, *blocked* - this is a sentence of interest.

normalized (i.e. was not one of the relations contained in the ontology), which left us with 53,208 relationships[4]. We then put all relations in active voice, collapsing passive/active pairs of normalized verbs such as *isMetabolizedBy* and *metabolizes* into a single feature. This left 49,021 normalized relations. However, many of these normalized relations were duplicates of each other because a given drug-gene relationship could be reported in similar ways many different times throughout the biomedical literature. We chose to eliminate duplicate paths of this nature. After collapsing the duplicate edges, we were left with 24,155 unique edges, which we used to construct a semantic network, a subset of which is shown in Figure 2.4. Each edge in the semantic network had the form shown in Figure 2.3 (top), but is simplified in Figure 2.3 (bottom) for clarity.

[4]In our article describing this work [107], we stated, "Examples of relations that could not be normalized included protects, mimicked, oxidize, encode, and seen. We are in the process of expanding our ontology to include some of these less common relations."

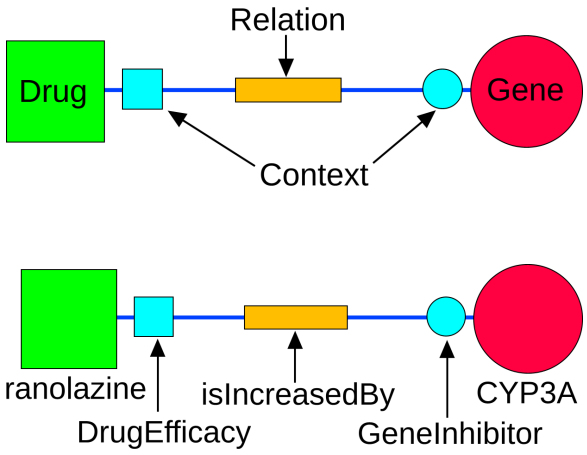


Figure 2.3. A single drug-gene edge in the semantic network. A composite entity consists of a drug or gene and its surrounding [normalized] context terms. (a) The general form of an edge. (b) A specific example.

2.3.3 Methods: Building A Classifier for DDIs

Feature Extraction

The feature extraction phase of this project relied on one central assumption: that the shortest textual path linking two drugs in the network represented the simplest explanatory mechanism of their interaction (if any such mechanism existed). The set of relevant features then consisted of all the genes, relations, and context terms found on the shortest path. To find the shortest path between any two drugs D_1 and D_2, we performed a breadth-first search for D_2, starting at D_1. Breadth-first search is guaranteed to yield the optimal (shortest) path between two points on a graph [119]. The shortest possible path between any two drugs in the network has the form shown in Figure 2.5. For the purposes of building our training set, we considered only drug pairs that had one or more shortest paths of the form in Figure 2.5; if the shortest path was longer than this, the drug pair was not included in the training set. We made this decision because we wanted to explore only those drug pairs for which the mechanistic explanation provided by the shortest path could be interpreted easily.

By assigning each feature a numeric index, we could easily convert the lists of normalized terms found on the shortest paths into a matrix of numbers, with each

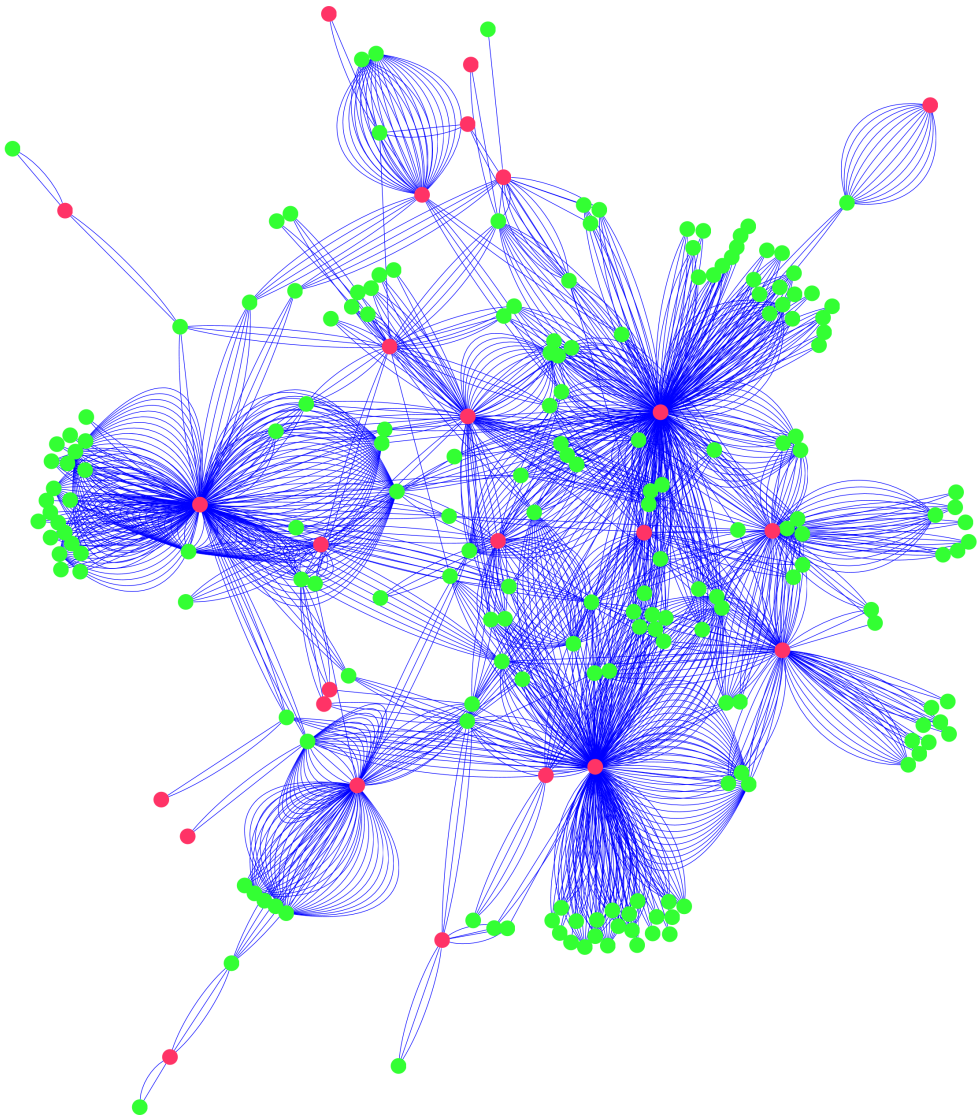


Figure 2.4. A subset of the semantic network (selected to enhance visual clarity), including only the 43 most pharmacogenomically-important genes from PharmGKB and 600 drugs that were known to interact with at least one other drug. The green nodes represent drugs and the pink nodes genes. The context terms and relations are not shown in this picture, but are present on every edge, as shown in Figure 2.3. Multiple edges between the same gene-drug pair in this figure represent different textual relationships found between that gene and drug in the literature.

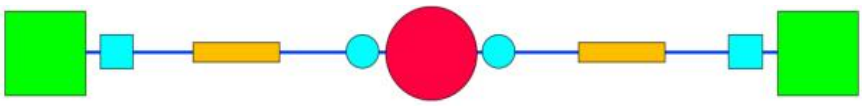


Figure 2.5. The minimum-length path between two drugs in the network. It is two edges long. The colors and symbols in this figure are identical to those in Figure 2.3: green squares represent drugs, and the pink circle represents a gene. The yellow rectangles represent relations, and the blue circles and squares represent context terms for genes and drugs, respectively.

row corresponding to a single path and the columns corresponding to the number of occurrences of each feature on the path. If multiple shortest paths were found, we included a separate row in our training matrix for each unique path.

Classification

The next step was to train a supervised machine learning classifier to recognize interacting drug pairs based on the textual features of their connecting paths. We randomly sampled 5000 drug pairs from a list of known interacting pairs provided by DrugBank [152], then selected 5000 additional drug pairs randomly from the drug lexicon, ensuring that none of them were on DrugBank's list of interactions. For each of the 5000 pairs in our positive and negative training sets, we found all of the paths between the two drugs in the pair that took the form shown in Figure 2.5 and recorded the features observed along the paths. Each path between an interacting drug pair became a positive training example, and each path between a noninteracting drug pair became a negative training example.

We used a random forest [43], specifically the implementation found in the R library `randomForest`, to perform the final classification for all of the drug pairs in our training set. The random forest is an ensemble method in which many uncorrelated decision trees vote to classify data points; it outperformed both logistic regression and a support vector machine (SVM) classifier used in the early stages of this project. Each tree in the random forest uses only a subset of the features for classification, which ensures that votes from different trees are uncorrelated. We found that the overall classification error stabilized when approximately 200 trees were included in

the forest.

Performance Evaluation

The standard metric of performance for the random forest is the out-of-bag (OOB) estimate of the error, which is similar to leave-one-out cross-validation [43]. Each tree in the random forest is constructed using only about 2/3 of the available training data; the rest of the data points are referred to as the "out-of-bag" data for that tree. Thus it is possible to build the entire forest, then reclassify each training example using only those trees for which it was OOB. The generally-accepted rule is to use a voting cutoff of 50% to classify a training point as positive; this means that for the random forest to assign the label "interacting" to a path, 50% or more of the trees in the forest had to classify that path as interacting. We used the standard OOB estimate of the error to evaluate the random forest's performance on our training data.

One interesting feature of the random forest is that it provides a natural measure of its classification certainty for each training example - namely, the fraction of trees that voted "interact" for that example. By ranking the paths for a particular drug pair based on the number of "yes" votes each received from the random forest, we can determine which path(s) represent the most likely mechanism(s) of interaction for that pair.

2.3.4 Results: Ranking of Important Features

A total of 1806 entities were represented in our network: 1061 drugs, 532 genes, 172 context terms, and 41 relations. There were 172,271 negative training examples (paths between 5000 noninteracting drug pairs) and 182,534 positive training examples (paths between 5000 interacting drug pairs) in our training set, all of which had the form shown in Figure 2.5.

The random forest uses a permutation method to provide an estimate of which textual features were most important to the classification process; the 50 most important features are shown in Figure 2.6. Among the most important features are the genes *ABCB1, IL28B, TNF, CYP3A4, EGF, CAMP,* and *CYP2D6*, the context terms

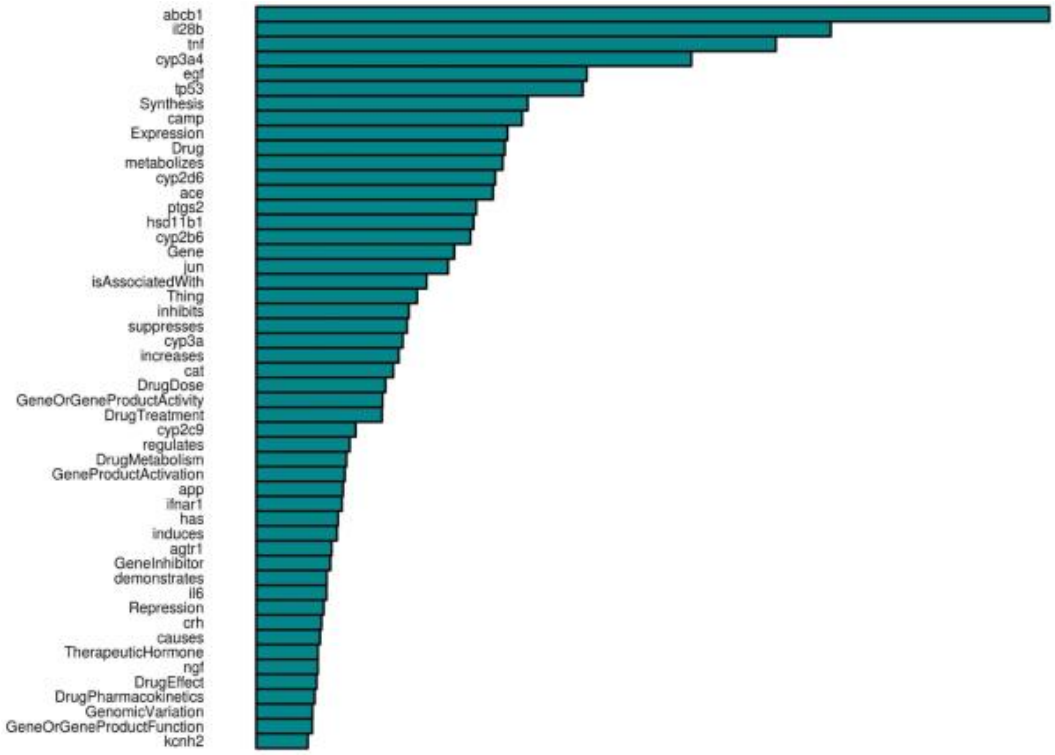


Figure 2.6. The 50 most important features to the random forest classifier, ordered according to a permutation metric [43]; the numeric values of importance are not as informative as the relative sizes of the bars.

Synthesis, Expression, DrugDose, GeneOrGeneProductActivity, DrugTreatment, DrugMetabolism, GeneProductActivation, GeneInhibitor, Repression, and *DrugEffect*, and the relations *metabolizes, isAssociatedWith, inhibits, suppresses, increases, regulates*, and *induces*.

Three context terms on this list appear strange at first glance: *Drug, Gene,* and *Thing*. Context is normalized to the term *Drug* or *Gene* if the drug/gene seed is itself the subject or object of the verb or nominalized verb in the sentence, as in the example sentence *CYP2C9 metabolizes warfarin*. In this sentence, the gene *CYP2C9* would be normalized to *CYP2C9 Gene* and the drug warfarin would be normalized to *warfarin Drug*. Context is normalized to the term *Thing* if the real context is some property of the drug or gene that cannot be otherwise normalized. For example, in one sentence,

Table 2.2. The final contingency table for the random forest classifier.

	Random Forest Classification		
True Class	**0**	**1**	**Class-wide Error**
No Interaction	135,842	36,429	0.211
Interaction	36,915	145,619	0.202

the authors used the term "polymorphism" incorrectly as a modifier of a drug name, referring to "polymorphisms in (drug name) drugdose. . . ". Because the drug context in that sentence was *polymorphisms* but the seed was a drug name and not a gene name, *polymorphisms* was not found in the ontology among the acceptable context terms for the drug and the context was normalized to *Thing*. One can therefore think of *Thing* as a marker for cases where normalization of a context term was not possible (using the current version of the ontology), but normalization of the relation proceeded normally.

2.3.5 Results: Predicting Drug-Drug Interaction Mechanisms

The final contingency table for the random forest classifier is shown in Table 2.2. The random forest correctly assigned 281,461 out of 354,805 training paths (79.3%; 135,842 non-interacting and 145,619 interacting paths) to the correct class. It said 36,429 paths represented interactions when the drug pair involved did not appear on the list from DrugBank (false positives), and claimed that 36,915 paths did not represent interactions when in fact the drug pair did appear on the list from DrugBank (false negatives).

We can get a sense of the significance of this result by considering what would happen if we simply flipped a coin to assign the label "interacting" or "noninteracting" to each path. Roughly 50% of the paths in our training set corresponded to interacting drug pairs, and the other 50% to noninteracting drug pairs. Therefore, if we assigned the labels "interacting" and "noninteracting" entirely at random, we would expect to

correctly classify about 50% of paths (with the false positive error rate approximately equal to the false negative error rate). Our method thus represents an improvement in accuracy of nearly 30% over simple guessing.

In addition to classifying interacting and noninteracting drug pairs with nearly 80% accuracy, our method provides insight into the possible mechanisms by which drugs interact. By choosing a path from one drug to the other through a particular gene, we obtain one potential mechanism for how the two drugs could interact. For example, Figure 2.7 shows a selection of the highest-ranking paths for a known interacting drug pair: verapamil and atorvastatin. Table 2.3 shows the Medline sentences corresponding to the edges that comprise these paths. All of these paths received at least 90% "yes" votes from the random forest.

Table 2.3. The raw sentences from Medline abstracts that correspond to the edges shown in Figure 2.7. Each path between verapamil and atorvastatin consists of two edges (i.e. two sentences).

PMID	Normalized Relation	Sentence
Relationships involving f2 (thrombin)		
2611956	verapamil Thing inhibits Gene f2	Ilexonin A and verapamil markedly inhibited the thrombin induced Ca2+ influx.
12921859	atorvastatin DrugEfficacy prevents Gene f2	In addition, thrombin induced NF-kappaB translocation and membrane translocation of RhoA in smooth muscle cells which were both prevented by pre-treatment of the cells by atorvastatin.
12921859	atorvastatin Drug decreases Gene f2	How atorvastatin could limit the pro-inflammatory response to thrombin was studied in cultured rat aortic smooth muscle cells.
15792677	atorvastatin Drug decreases Thing f2	Atorvastatin reduces thrombin generation after percutaneous coronary intervention independent of soluble tissue factor.
Relationships involving ABCB1 (P-glycoprotein, MDR1)		
16996216	atorvastatin Drug causes Synthesis ABCB1	Atorvastatin at 10 and 20 microM up-regulated ABCB1 expression resulting in a significant 1.4-fold increase of the protein levels.
16996216	atorvastatin Drug increases Thing ABCB1	Treatment of HepG2 cells with 20 microM atorvastatin caused a 60% reduction on mRNA expression ($p < 0.05$) and a 41% decrease in ABCB1-mediated efflux of Rhodamine123 ($p < 0.01$) by flow cytometry.
9607955	verapamil DrugTreatment induces GeneOrGeneProductActivity ABCB1	Previous drug exposure of the cells showed that verapamil, celiprolol, and vinblastine induced the P-gp expression, while metkephamid (MKA) decreased the P-gp expression level as compared to the control.

9636053	verapamil DrugActivity demonstrates Thing ABCB1	P-gp proteoliposomes from *P. pastoris* showed a strong verapamil- and valinomycin-stimulated ATPase activity, with characteristics (KM, Vmax) similar to those measured in mammalian cells.
9535788	verapamil Drug inhibits Gene ABCB1	In addition, the DNA-damaging agent was found to enhance in a dose-dependent manner cellular efflux of the P-gp substrate rhodamine 123, which was inhibited by the P-gp inhibitor verapamil, thus providing evidence that exposure to MMS led to increased P-gp-related drug transport in rat liver cells.
7769842	verapamil Drug inhibits GeneOrGeneProductActivity ABCB1	When P-gp function was assessed by Rhodamine 123 (Rh123) efflux kinetics, we found that only KG1a and KG1 cells, which have an early (immature) CD34+ CD33- CD38- phenotype, and to a lesser extent TF1, with an intermediate (CD34+ CD33+ CD38+) phenotype, displayed significant P-gp activity which could be inhibited by both verapamil and SDZ PSC 833.
16457995	verapamil DrugAbsorption inhibits Gene ABCB1	While cyclosporine and verapamil significantly increased the absorption of methylprednisolone and vinblastine through potent inhibition of intestinal P-gp, tacrolimus failed to achieve this.
17936633	verapamil Drug regulates GeneOrGeneProductActivity ABCB1	The results displayed that only compound 3c was P-gp inhibitor as Elacridar, while compound 3a and reference compounds Cyclosporin A and Verapamil modulated P-gp activity saturating the efflux pump as substrates.
16260035	verapamil Drug suppresses Gene ABCB1	Depsipeptide-resistant KU812 cells expressed P-glycoprotein (P-gp) and their resistance was abolished by co-treatment with verapamil.
15257901	verapamil DrugDose isAssociatedWith Repression ABCB1	DL-PPMP and verapamil were found to inhibit MDR1 gene expression in KBV(200) cells at the mRNA level, and complete inhibition occurred after a 48-hour DL-PPMP treatment at 25 micromol/L.
15257901	verapamil Drug inhibits Expression ABCB1	The inhibition of GCS and mdr1 gene expressions is positively correlated with the concentrations of DL-PPMP and verapamil, which can reverse MDR by inhibiting synthesis of GCS and mdr1 gene, indicating the positive correlation between the expression of GCS gene and MDR in KBV(200) cells.
7749215	verapamil DrugTreatment decreases Expression ABCB1	The level of mdr1 mRNAs is decreased in the presence of verapamil (with a maximum effect obtained at the 24th hour), which suggests that the mechanism of action of verapamil is transcriptional and/or post-transcriptional.
Relationships involving VEGFA		
14615256	verapamil Drug decreases Synthesis VEGFA	Verapamil (100 microM) decreased IL-6 and VEGF production ($P < 0.03$ and $P < 0.005$, respectively) in central keloid fibroblasts cultures at 72 h.
16701707	atorvastatin Drug induces Gene VEGFA	We observed that atorvastatin significantly stimulated VEGF release in a dose-dependent manner.
17389519	atorvastatin Drug isAssociatedWith Repression VEGFA	Atorvastatin effectively inhibited laser-induced CNV in mice and was associated with downregulation of CCL2/MCP-1 and VEGF and reduced macrophage infiltration into the RPE/choroid.

12084593	atorvastatin Drug decreases Expression VEGFA	Atorvastatin therapy reduced VEGF plasma levels in CAD patients (from 31.1 +/- 6.1 to 19.0 +/- 3.6 pg/ml; $p < 0.05$).
Relationships involving CYP3A4		
15001968	verapamil DrugEfficacy isAssociatedWith Expression CYP3A4	Values for the maximum rate of metabolism (V(max)) of verapamil N-dealkylation (formation of D-617) and N-demethylation (formation of norverapamil) activities correlated with the CYP3A4 protein content in both organs.
11907487	CYP3A4 Gene induces Drug-Metabolite verapamil	Consistent with expression data, formation of verapamil metabolites catalyzed by CYP3A4 and CYP2C was shown.
11005703	CYP3A4 Enzyme metabolizes Drug atorvastatin	Atorvastatin, cerivastatin, lovastatin and simvastatin are predominantly metabolised by the CYP3A4 isozyme.
11061579	CYP3A4 Gene metabolizes Drug atorvastatin	Atorvastatin is metabolized solely by CYP3A4, and pravastatin metabolism is not well defined.
Relationships involving CYP3A		
16513446	verapamil Drug inhibits GeneOrGeneProductActivity CYP3A	Verapamil inhibited CYP3A activity, with a maximum effect occurring within 10 days.
16013069	verapamil DrugMetabolism inhibits Repression CYP3A	The above data suggested that the metabolism of verapamil and the formation of norverapamil was inhibited by naringin possibly by inhibition of CYP3A in rabbits.
14744949	verapamil DrugIsoform inhibits Gene CYP3A	The present study showed that verapamil enantiomers and their major metabolites [norverapamil and N-desalkylverapamil (D617)] inhibited CYP3A in a time- and concentration-dependent manner by using pooled human liver microsomes and the cDNA-expressed CYP3A4 (+b5).
12433810	atorvastatin Drug increases Expression CYP3A	Treatment of 2- to 3-day-old human hepatocyte cultures with $3 \times 10(-5)$ M lovastatin, simvastatin, fluvastatin, or atorvastatin for 24 h increased the amounts of CYP2B6 and CYP3A mRNA by an average of 3.8- to 9.2-fold and 24- to 36-fold, respectively.
16258024	CYP3A Gene metabolizes Drug atorvastatin	Atorvastatin (ATV) is primarily metabolized by CYP3A in the liver to form two active hydroxymetabolites.

Most of the edges connecting verapamil and ABCB1 in Figure 2.7 seem to indicate that verapamil inhibits the activity of ABCB1. The edges connecting atorvastatin and ABCB1 indicate that atorvastatin upregulates the production of P-glycoprotein, the protein product of ABCB1. The two drugs' effects on ABCB1 therefore interfere with each other. Following another path, this time through the gene CYP3A4, we see that CYP3A4 induces the breakdown of verapamil into its metabolites, specifically by N-dealkylation and N-demethylation of the drug. Since CYP3A4 is a major metabolizing enzyme for atorvastatin, we might expect that coadministration of the two drugs could lead to heightened levels of one or both of them in the body, leading to toxicity.

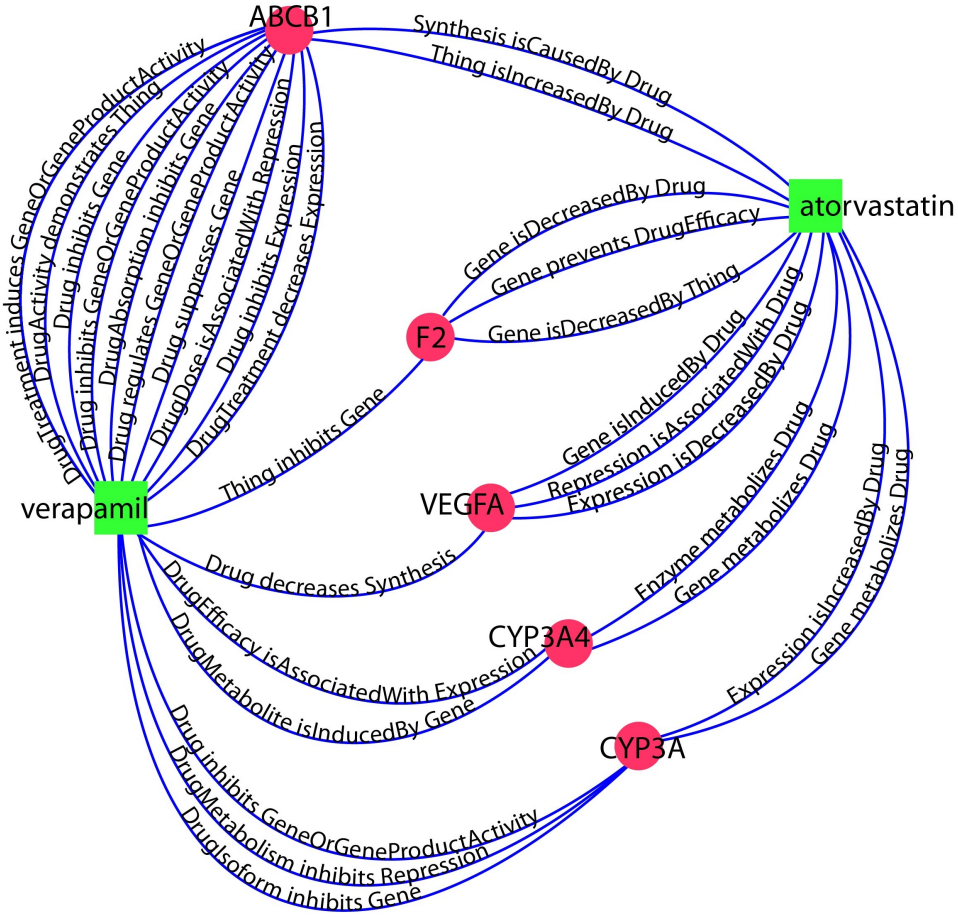


Figure 2.7. A subnetwork representing some of the possible interaction modes between verapamil and atorvastatin.

These represent two different possible mechanisms of interaction.

These suggested mechanisms are useful because they provide summaries of what the scientific community knows about pharmacogenomically-mediated interactions between drug pairs of interest. The drug-gene relationships that form the basis of these mechanisms are all existing knowledge; however, our method provides a novel way to connect disparate facts from across the biomedical literature to provide mechanistic explanations of drug-drug interactions[5]. In the case of drug pairs that are already known to interact, using this approach provides a list of potential mechanisms of interaction, which may help us uncover new mechanisms that are not yet part of common medical knowledge. By looking at known interacting drug pairs with similar mechanisms of interaction, we can also begin to predict what the phenotypic effects of our newly-predicted interactions might be.

2.4 A Retrospective Critique of this Work

This chapter describes one simple way that text mining can address an important biomedical problem: predicting drug-drug interactions based on drug-gene interactions extracted from the literature. In effect, we were attempting to predict novel DDIs based on mechanistic and structural information about the drugs themselves and their interactions with proteins. The advantage of this approach lies in the fact that it relies mainly on chemical and bioactivity data from laboratory studies rather than clinical data. As a result, it could potentially be used to predict DDIs before drugs enter the market.

We encountered a few issues in this project that paved the way for the work in the rest of this thesis. The project also made us aware of some broader challenges that, while not problematic for this particular study, limited its generalizability to other problems in biomedical information extraction.

- *Using a custom ontology introduced a bottleneck.* The construction of the PHARE

[5]Other authors have already pointed out that by finding new ways to connect seemingly-unrelated facts from throughout the scientific literature, we can often generate interesting, and novel, hypotheses. For an example, see [139].

ontology was a separate project [19, 20] that created a black box upon which our work depended, but that was difficult to understand and update. Multiple assumptions were made in the construction of the ontology that, while reasonable, introduced biases in the types of relationships we could extract, as well as their granularity.

- *Normalization using the ontology resulted in low recall.* Due to these assumptions, we estimate that we were able to retrieve and normalize less than half of relevant drug-gene relationships from Medline. Even estimating recall is difficult, since it requires manual analysis of hundreds of sentences; absolute recall numbers were not reported in the original study describing relationship extraction using PHARE [20].
- *Spelling errors or variants of named entities not in PHARE could not be detected.* PHARE depended on the PharmGKB lexicons to recognize and normalize entities. Any variant that was not an exact match to these lexicons could not be detected.
- *Using PHARE limited the approach's generalizability.* Aside from other relationships involving known drugs, genes and phenotypes, this approach could not be used for other similar projects that might involve connecting two relationships in series.

Perhaps most importantly, while normalization using PHARE helped us collapse descriptions of relationships into a smaller space of descriptions, it did not tell us anything about how various descriptions were semantically related to each other. PHARE extracts and normalizes only the connecting verb and context terms from a sentence. It does not normalize the entire description, which is why we resorted to using the verbs and context terms separately as features. This essentially assumed that each component of a path (verbs, gene, context terms) contributed on its own to determining whether the two drugs interacted, which is incorrect (*Gene isDecreasedBy Drug* is not 2/3 similar in meaning to *Gene isIncreasedBy Drug*). This also explains why our most predictive features were all gene names.

In the rest of the dissertation, we describe techniques that address normalization in a different way, and that alleviate many of the issues described above. In Chapter 9, we revisit drug-drug interactions from this new perspective.

Chapter 3

Distributional Semantics

For decades, researchers in natural language processing (NLP) have investigated the idea that human notions of semantic similarity, which derive from our experiences in the world and our sensory associations with various words and grammatical structures, might be replicated by a statistical approach that considers how these words and phrases are used in large corpora. One of the earliest quotes on this subject, by Firth (1957), perhaps summarizes this idea the best: "You shall know a word by the company it keeps" [31]. This idea – that words occurring in similar contexts tend to have similar meanings – is known as the *distributional hypothesis* [26, 147]. Distributional semantics is not a field as much as a collection of techniques that have been developed over the years by researchers in computer science, philosophy and linguistics. An excellent general review can be found in [147]. For a review of distributional semantics in the biomedical domain, please see [16].

This chapter reviews some of the major themes in distributional semantics and takes a closer look at two approaches that make appearances later in this dissertation: random indexing and the skip-gram model.

3.1 Target, Context, Model

Methods in distributional semantics typically involve three components, which I will call the target, context, and model.

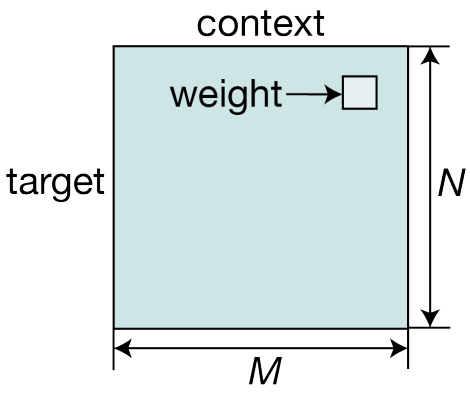


Figure 3.1. A distributional semantics model conceptualized as a matrix in which the rows are the target, the columns are the context, and the model describes how to create the weight (numeric value) that is present in each cell.

- The *target* refers to the entity whose semantics we care about. Examples include words, phrases, pairs of words/phrases, or documents.
- The *context* is the set of textual features that we think should be similar for two targets if the targets themselves are semantically similar. We might assume, for example, that words with similar meaning will be used in the same documents. In that case, the context is the set of all documents.
- The *model* tells us how information about the context is used to produce a *representation* of the target. This representation often takes the form of a vector of numeric weights: one for each context feature. The model also tells us how the representations should be compared to establish how semantically similar two targets are.

Following the exposition in [147], we will picture these three components as a matrix in which the rows represent the target and the columns the context. The matrix elements are numbers ("weights") created using the model (see Figure 3.1). The width of the matrix is the number of possible context features, M, and the height of the matrix is the number of targets, N. This conceptualization is abstract, but it helps us understand very different-looking distributional semantics techniques within a single, unifying framework.

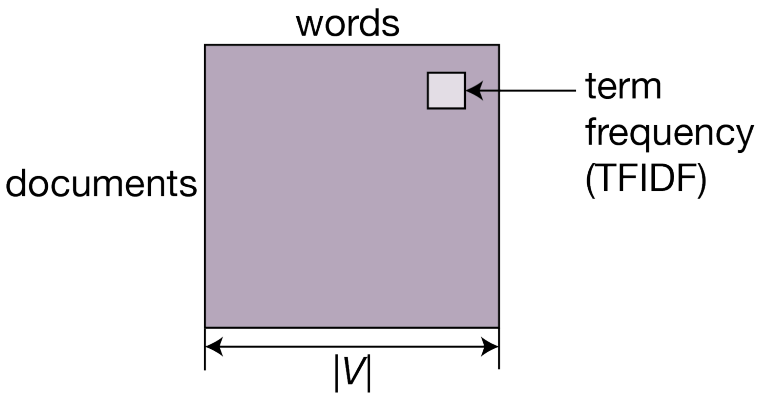


Figure 3.2. The SMART information retrieval system as a matrix. Here the rows are documents and the columns are words in those documents. A cell of the matrix contains some variant of the TFIDF score for a given word.

3.2 Historical Background

Some of the earliest distributional semantics models focused not on the semantics of words and phrases, but the efficient retrieval of documents[1]. In fact, many of these ideas form the conceptual foundation of modern search engines [84, 147]. For example, the SMART information retrieval system [123, 125], developed in the 1960s, represented documents as vectors of length $|V|$, where V is the set of all unique words present in the corpus. A document vector's elements contained TFIDF scores[2] for all of the words for that document. We can draw this system as shown in Figure 3.2.

The goal of the SMART system was to provide an efficient way to retrieve documents from a database: each document would be represented as one of these vectors, and so would a user's query. By ranking the documents based on their vectors' similarity to the query vector, the best matching documents would, theoretically, filter to the top.

[1] For a review of even earlier methods for creating word space models, mainly from linguistics and cognitive psychology, please see [121], Chapter 3: Word Space Models.

[2] TFIDF refers to "term frequency inverse document frequency", a weighting scheme for words in documents in which the score goes up the more times a word occurs in the document, and it decreases with the number of documents in which a term is present. So for example, the term "the" might have a high frequency within a given document, but because it occurs in almost all documents, its TFIDF is low. There are many formulations of TFIDF weighting. A good review is given in [84] Chapter 6: Scoring, Term Weighting and the Vector Space Model and some of the original research on TFIDF for document retrieval can be found in [124].

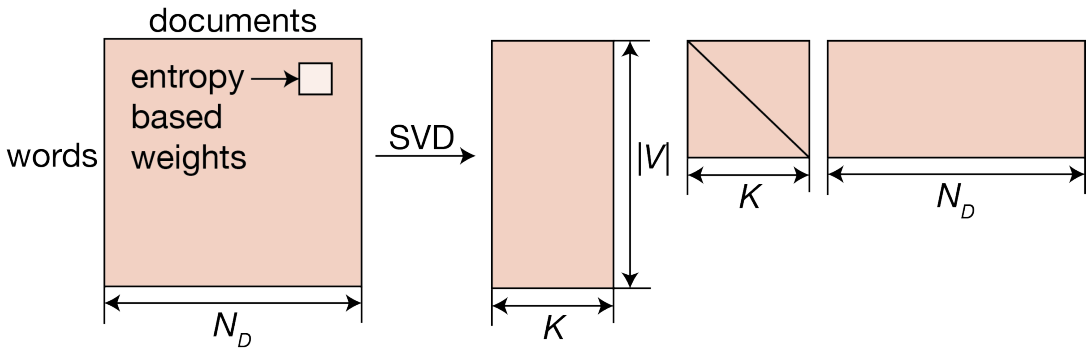


Figure 3.3. Latent Semantic Analysis (a.k.a. Latent Semantic Indexing), a technique whereby a reduced SVD is performed on a term by document matrix. This matrix is the transpose of the one used by the SMART information retrieval system in Figure 3.2, with a slight modification in terms of the weighting scheme.

However, beginning in the 1990s, researchers began to use these vector space models (VSMs) for tasks beyond search. Most of these techniques focused on two areas: measuring the semantic similarity of words and phrases, and measuring the similarity of relationships between entities (relational similarity).

3.2.1 Word and Phrase Similarity

In the 1990s, inspired by the work of the SMART team [125] on document retrieval using VSMs, Deerwester and colleagues developed *Latent Semantic Analysis (LSA)* [26]. LSA essentially works by transposing the matrix in Figure 3.2 and then performing a reduced singular value decomposition (SVD) on it [137], as shown in Figure 3.3. It also uses a slightly different TFIDF weighting scheme, based on entropy. The weight of term i for document j is given by

$$f_{ij} = (\log(\mathrm{TF}_{ij}) + 1) \times \left(1 - \sum_j \frac{p_{ij} \log p_{ij}}{\log N_D}\right)$$

where N_D is the total number of documents in the collection, $p_{ij} = \mathrm{TF}_{ij}/f_i$, and f_i is the frequency of term i in the corpus [28, 121].

By including only the top K singular values and multiplying the first two matrices

in Figure 3.3, one can obtain compressed (K-dimensional) vectors for all of the words in V. One disadvantage of LSA is that K is a free parameter, and LSA's results on various word similarity tasks are dependent on this parameter choice[3] [26, 66].

In LSA, the word vectors are typically compared to each other using *cosine similarity*[4], which produces scores in the range $\sigma \in [-1, 1]$.

Even as LSA was being developed, other authors were building word vectors using different notions of context. In 1992, Schütze described a model similar to LSA in which the context was, instead of documents, windows of words around a target word [129, 130]. In 1995, Lund and Burgess [82] extended these ideas further to create *Hyperspace Analogue to Language (HAL)*, which again uses word windows for context, but introduces a positional weighting scheme for the context words (Figure 3.4). Although not as widely cited as LSA, HAL's definition of context is actually closer to that of many modern distributional semantics techniques, including the popular `word2vec` [88]. Other authors have explored different notions of context, such as grammatical dependencies [76, 101], or incorporated additional information about the target words, such as selectional preferences for argument positions [29].

Two more recent methodological advances in the estimation of word and phrase similarity using distributional semantics are random indexing [120] and neural word embeddings [4, 88], both of which are reviewed in more detail later in this chapter, and both of which make appearances in Chapter 4 of this dissertation.

3.2.2 Relational Similarity

While models of words have dominated the distributional semantics literature, there is nothing about the ideas of target and context that require the target to be a word. In fact, a great deal of recent work has focused on building distributed representations of longer stretches of text such as phrases [14, 89, 102, 159], sentences [60] and documents [68].

[3] The original LSA paper stated, of K: "We have not explored the degree of accuracy needed in these numbers, but we guess that a small integer will probably suffice" [26].

[4] The cosine similarity of two vectors, w and v, is defined as $\sigma(w, v) = w \cdot v / \|w\| \|v\|$, where $\| \cdot \|$ denotes the L_2 (Euclidean) norm.

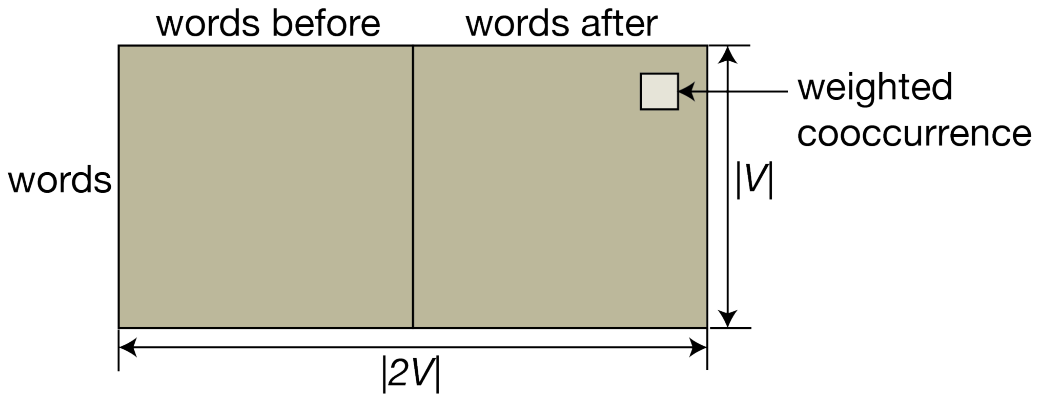


Figure 3.4. Hyperspace analogue to language (HAL). There are some subtleties to this model that are usually ignored. The authors defined context as a narrow window of words on either side of a target word and built a $|V| \times |V|$ cooccurrence matrix where the rows contained cooccurrence information for words appearing before the target word and the columns contained cooccurrence information for the words following it. Cooccurrence frequencies were weighted by $w - x$, where w is the window width and x is the distance between the target word and context word. The rows and columns of this $|V| \times |V|$ matrix were then concatenated to produce a vector of length $2|V|$ for each word, where the first half of the vector corresponded to cooccurrence counts before the target word and the second after the target word. To make HAL consistent with our "target, context, model" formulation, we show the second version of the cooccurrence matrix here. HAL used Euclidean or Manhattan distance to calculate the similarity of different rows in this matrix to each other.

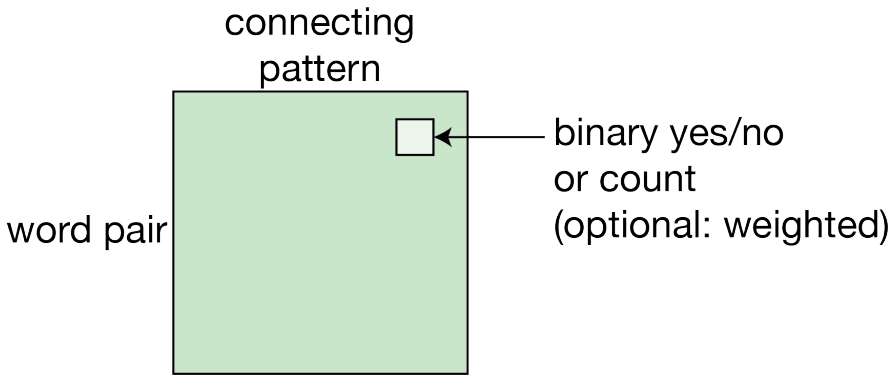


Figure 3.5. A pair-pattern matrix used to establish similarity of relations. Here the rows are pairs of words (or phrases) and the columns are some type of connecting pattern, such as the words in between the pair of words in a sentence, or dependency paths. The ijth element reflects how frequently pair i is connected by pattern j in the corpus.

One type of distributional similarity that is relevant for many important tasks in text mining, especially in biomedical text mining, is *relational similarity*: given two entities, can we obtain a distributed representation of their relationship, and can we assess how similar that relationship is to relationships between other entity pairs? It appears that relational similarity *between* entities cannot simply be derived from properties of the entities themselves – it is something fundamentally different [73, 147].

We can conceptualize the relational similarity task as operating on a matrix where the rows are pairs of words (or phrases) and the columns are patterns that connect these word pairs in a corpus. Popular choices include the span of words occurring between the pair words in sentences, or dependency paths connecting them in dependency graphs. The matrix elements contain measures of association between pairs and patterns, such as counts of how frequently pair i is connected by pattern j in a corpus.

This *pair-pattern matrix* was first introduced by Lin and Pantel in 2001 in their paper describing *DIRT (Discovery of Inference Rules from Text)* [77]. This work was also the first to coin the term *extended distributional hypothesis*, which states that patterns co-occurring with similar pairs have similar meanings. In their 2001 paper, Lin and Pantel applied this idea to paraphrase detection [77]. Subsequently, Turney made a similar analogical leap to the one that turned SMART into LSA. He claimed that if the extended distributional hypothesis is true, its inverse is also true: word pairs connected by similar patterns have similar semantic relations. This is the *latent relation hypothesis*, introduced in Turney's 2005 paper on *Latent Relational Analysis (LRA)*, a technique which uses the SVD on a pair-pattern matrix in a manner similar to LSA [145, 146].

More recently, Riedel *et al* [112, 155] used factorization of a pair-pattern matrix for relation extraction, although their factorization method went beyond word pairs to include parameters related to the similarities of individual words within the pairs and also incorporated structured information from knowledge bases. Several related methods have clustered patterns to discover latent groupings of entity pairs corresponding to distinct relations [41, 117, 135, 157]. Others have clustered both rows and columns of a pair-pattern matrix to discover relations between entities on the web [7] and to

build semantic networks [63]. The issue of textual *entailment*, finding the degree to which one statement implies the existence of another, is a core problem in natural language processing and is also closely related to the ideas described above. It is reviewed extensively in [21] and [22].

3.3 Some Practical Considerations

By this point, many observers will have noticed that the matrix shown in Figure 3.1, regardless of whether it is a word-document, word-word, or pair-pattern matrix, is going to be very large. The number of unique tokens in a reasonably-sized corpus, such as Medline, typically ranges from $|V| \approx 100,000$, where $|V|$ is the vocabulary size, to over 1 million. If we also consider word combinations (phrases) up to length n, the total number of words and phrases will be on the order of $|V|^n$. The numbers of documents, patterns, etc. in a corpus are somewhat arbitrary but can be huge. Medline, for example, contains well over 16 million abstracts as of this writing[5].

Even though techniques like LSA reduce the dimensionality of the final target representations using matrix decompositions, they still necessitate first forming the entire matrix (Figure 3.1) and then decomposing it. Computing the SVD on a giant matrix can be very costly, both with respect to runtime and memory [120].

The two techniques discussed in the remainder of this chapter, random indexing (Section 3.4) and `word2vec`'s skip-gram model (Section 3.5), create low-dimensional representations of word and phrase targets without ever constructing the full co-occurrence matrix for their contexts. As we will see, this dimensionality reduction will come at a cost, which is interpretability - the elements of lower-dimensional representations do not correspond to measurable quantities, such as counts of words within documents. What is gained, however, is computational efficiency.

[5]The total number of citations in Medline is actually much higher, around 24 million, but not all of these have abstracts.

3.4 Random Indexing

In the mid-2000s, Sahlgren and colleagues developed *Random Indexing* [120], an alternative to LSA that is based on Pentti Kanerva's work on sparse distributed memory [55, 56].

3.4.1 How it Works

In random indexing, each word in a corpus is assigned a random, sparse *elemental vector*. The *dimension* of this vector is its length, and the *seed length* is how many of the terms in the vector are nonzero; typical values for dimension and seed length are $100 - 1000$ and $5 - 20$. An elemental vector is built by initializing all of its elements to zero and then randomly assigning the value "1" to $s/2$ elements and "-1" to a different $s/2$ elements, where s is the seed length. After the elemental vectors are assigned, a *context vector* is built for a particular target word by adding together the elemental vectors from words that occur within some pre-specified *window width* of the target word. It is important at this stage to distinguish the elemental from the context vectors: elemental vectors are randomly assigned, and context vectors are built for each target word using the elemental vectors of the other words that surround it. It is the context vectors that will be used to compare word meanings.

The process for building the context vectors is simple: one moves through a corpus with a bin of width $2w + 1$, where w is the window width, and adds elemental vectors for all words within the bin to the context vector for the word in the middle. These added elemental vectors may optionally be weighted according to some predefined scheme, such as their corresponding context word's overall frequency in the corpus or the distance between context and target word. Finally, the context vectors are normalized to unit length. To evaluate the semantic similarity of two words, one calculates the cosine similarity of the context vectors corresponding to those two words.

Different variants of random indexing encode word order for surrounding terms in context vectors in different ways. The most basic version ignores it completely; elemental vectors for all words within the bin are added directly to the context vector

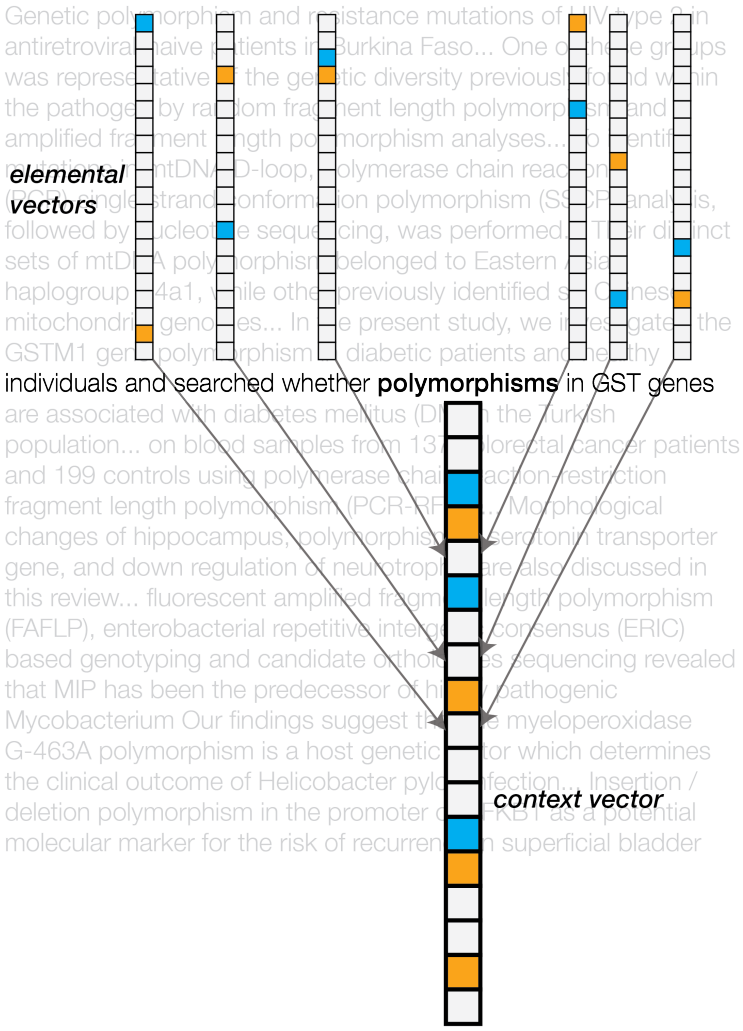


Figure 3.6. Random indexing. Here the context vector for the word *polymorphism* is being assembled from the elemental vectors of the cooccurring words within a window width of 3 (these include: *and, searched, whether, in, GST*, and *genes*). The blue squares represent -1 elements and the orange squares represent $+1$ elements. Note that the final context vector for *polymorphism* will not look like this. The word *polymorphism* occurs dozens of times in the corpus, so many more elemental vectors will be added to create the final context vector.

for the word in the middle, as shown in Figure 3.6. More elaborate versions use convolution [53] or permutations [122] to encode word order[6].

3.4.2 Why it Works

Consider the matrix in Figure 3.7. In this matrix, the rows represent target words, the columns represent words from the surrounding context (within a window width of w), and the ijth element is the raw count of the number of times word j appeared in the context for word i. Call this matrix A.

Now imagine randomly projecting A by multiplying it on the right by a random matrix, R, of size $|V| \times d$, where d might be on the order of $100 - 1000$. This random matrix is generated in a way that ensures its rows and columns are approximately orthogonal, so that $R^T R \approx I$, perhaps by sampling its elements from a Gaussian distribution[7]. The result will be a new matrix, $B = AR$, also of size $|V| \times d$, whose rows look nothing like those of the original matrix. However, as long as d is of sufficiently high dimensionality, the distances between the *rows* of B will be approximately the same as between the corresponding rows of A. This surprising result comes from the *Johnson-Lindenstrauss Lemma* [52], which states that if points in a vector space are projected onto a random subspace of sufficiently high dimension, the distances between the points are approximately preserved.

Now consider the fact that A could be written as the sum of many other matrices. A is a matrix of counts, so it could be assembled by the following procedure:

1. Initialize the $|V| \times |V|$ matrix, A, to all zeros.
2. Proceed through the corpus with a window of fixed size. Assume there are N total cooccurrence events that occur. At cooccurrence event i, context word with index c_i occurs in the context of target word with index t_i. This cooccurrence

[6]The preceding three paragraphs were borrowed, almost verbatim, from the Background section of our previously-published paper [104].

[7]A Gaussian distribution is, indeed, a common choice for producing random matrices with this property. However, it leads to dense, real-valued matrices that may be very large. In 2001, Achlioptas [1] showed that simpler distributions with zero mean and unit variance, such as a discrete distribution where $P(X = 1) = 1/2$ and $P(X = -1) = 1/2$, or one where $P(X = 0) = 2/3$, $P(X = 1) = \sqrt{3}/6$, and $P(X = -1) = \sqrt{3}/6$, also produce mappings that satisfy the Johnson-Lindenstrauss lemma.

is recorded by adding to A a $|V| \times |V|$ matrix, C_i, where all elements are zero except the element (t_i, c_i), which contains "1".

3. Continue until you reach the end of the corpus.

The A that results from this procedure is identical to the A that is in Figure 3.7. It can be written

$$A = \sum_{i=1}^{N} C_i$$

and the projected matrix, B, can be written

$$B = AR = \sum_{i=1}^{N} C_i R.$$

That is, we would have achieved the same result if we had randomly projected all of the C_is and then added them together, as opposed to projecting A. The matrix $C_i R$ contains all zeros except for one row - the row with index $t(i)$. That row contains the $c(i)$th row of R. This means that the $t(i)$th row of A will eventually contain the sum of all of the rows of R that correspond to words found in its context.

The random indexing algorithm described in Section 3.4.1 involves precomputing these rows of R (the elemental vectors) and adding them together to create context vectors. The only requirement is that these rows be orthogonal or nearly orthogonal, and ideally they would also be sparse (to save memory). Hecht-Nielsen [44] showed that in a high-dimensional space, there are many more nearly-orthogonal than purely-orthogonal directions. By simply sampling random directions from a high-dimensional space, we can produce elemental vectors that are nearly orthogonal. Random indexing uses a variant of Achlioptas's [1] idea about how these random elements should be distributed.

3.5 The Skip-Gram Model

In 2013, a team at Google [88] published a paper and associated software package detailing two simple neural network based language models that could be used to learn vector representations of words efficiently over large corpora. The best-performing

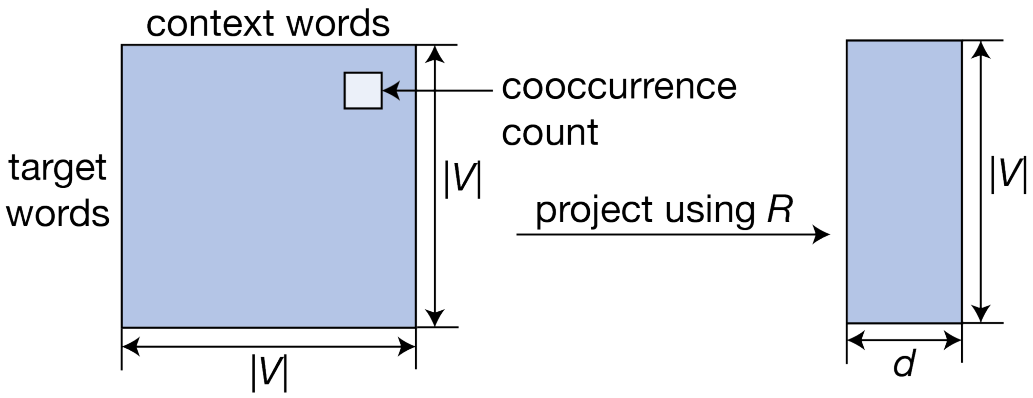


Figure 3.7. Random indexing as matrix projection. On the left is a word × word matrix of cooccurrence counts. This is what would result from random indexing if the elemental vectors were the length of the vocabulary, $|V|$, and each contained only a single "1" element and $|V| - 1$ "0" elements. We can view the random indexing algorithm as projecting this matrix using a sparse random matrix, R, whose rows and columns are orthogonal.

model is called the *skip-gram model*.

3.5.1 How it Works

The mathematical details of the skip-gram model are explained in detail in a tutorial from a researcher at the University of Michigan[8]. The skip-gram model is trained using stochastic gradient descent and backpropagation, and the end result is a matrix of size $|V| \times h$, where h is the size of the model's hidden layer (usually $100 - 1000$). As with random indexing, because there is never a need to represent a giant matrix of cooccurrence counts, and because the model is trained in an online, and not batch, fashion, the training process is fast and quite memory efficient, even on very large corpora. This fact, combined with the fact that the model's authors created a fast and user-friendly software suite (`word2vec`), has contributed to the skip-gram model's rapidly becoming the most popular method for building distributional word vectors.

[8] http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf

3.5.2 Why it Works

At first, it was a bit unclear why the skip-gram model leads to good word embeddings. However, Levy and Goldberg [38, 72] recently showed that word2vec's training process implicitly factorizes a positive pointwise mutual information (PPMI) matrix, as shown in Figure 3.8. Pointwise mutual information (PMI) is defined in the target-context formulation as

$$\text{pmi}(t; c) = \log \frac{p(t, c)}{p(t)p(c)}$$

where t represents the target, $p(t)$ the marginal probability of target t over all contexts, c a context term, and $p(c)$ the marginal probability of that context term over all targets. The *positive* pointwise mutual information (PPMI) is a modified version of the PMI in which negative values are replaced by zero.

In addition to showing that the skip-gram model factorizes a PPMI matrix, Levy and Goldberg also showed that with the right hyperparameter adjustments, the vectors built using the skip-gram model behave much like vectors produced using LSA, GLOVE [103], or other distributional techniques such as random indexing. For capturing different kinds of semantic similarity, what matters most appears not to be the model itself, but rather the type of context one uses in creating the vectors [71].

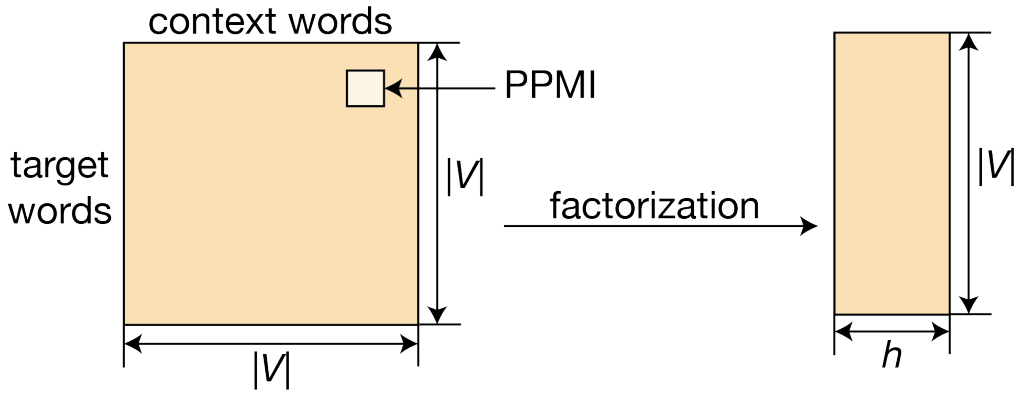


Figure 3.8. The skip-gram model as matrix factorization. Although the details of how the skip-gram model is trained involve stochastic gradient descent and backpropagation and appear very different from how LSA and random indexing work, Levy and Goldberg [72] showed that in fact, the skip-gram model is implicitly factorizing a PPMI matrix.

Chapter 4

Lexicon and Ontology Building

In this chapter, we show how distributional semantics approaches can be applied to the related tasks of biomedical named entity normalization, lexicon learning, and ontology building. The text in the first part of this chapter is drawn from one of our published papers [104].

4.1 Normalization

As we saw in Chapter 2, one of the key first steps in biomedical text mining is *normalization*: mapping the diversity of natural language to a smaller set of concepts, ideally those that are well characterized and understood, such as known drug, gene or disease entities. Normalization is particularly important in the biomedical domain because of the diversity with which even simple ideas can be described. For example, consider the following:

- To identify patients with diabetes in medical records, we need to understand that if the physician writes "diabetes", "diabetes mellitus", "diabotes" (spelling error), "Diabetes" (capitalization variant), or "DM" (abbreviation), all of these terms refer to the disease of interest.
- Protein names often have multiple forms, such as "ABCB1", "MDR1", "P-glycoprotein", and "P-gp", all of which refer to the same protein (multidrug resistance protein 1, mutations in which are responsible for many cases of

antibiotic resistance). Extracting drug-protein relationships from the literature requires knowing when two different-looking strings are referring to the same protein.

- Different authors have different styles and word preferences. One might consistently use "augments" while another uses "increases", yet they are both describing the same change. One radiologist might consistently use "hypodense" to describe a bright white region on an image, while another might use "hypoattenuating". Descriptions vary by institution, geographic region, subject area, journal and individual author.
- Drugs have brand names and generic names. If we want to know if a patient is taking Lipitor, we should also look for the term "atorvastatin".

It is tempting to want to handle normalization by hiring domain experts to build structured lexicons and ontologies. In fact, this has been the dominant normalization strategy throughout the biomedical domain. Millions of dollars and thousands of man hours have been invested in resources like UMLS [6, 78], a collection of dozens of ontologies that aims to catalogue all of the ways different biomedical concepts can be described and how these concepts are related to each other.

There are many good reasons to build a biomedical ontology: standardization of medical billing codes, medical education (teaching students to use consistent terms when describing clinical findings, for example) and solidifying our understanding of entities and their relationships in specific domains. Perhaps more importantly from the perspective of text mining, recent work has shown that for many text mining tasks, string matching to terms in ontologies works just as well as more sophisticated NLP approaches [54]. However, it is important to consider the relative expense of constructing ontologies vs. building NLP systems, especially if the ontologies are domain-specific and the NLP approaches can be applied to multiple domains.

Here I argue that the best approach to normalization is to use NLP as a first step to identify relevant terms and phrases and localize them within ontologies prior to human review. Distributional semantics techniques applied to relevant corpora, such as Medline or clinical notes, represent an efficient and scalable way to accomplish this task. Applying distributional semantics as a first step can greatly reduce the time

needed for ontology building and ensure that ontologies can be continually updated to reflect modern usage patterns.

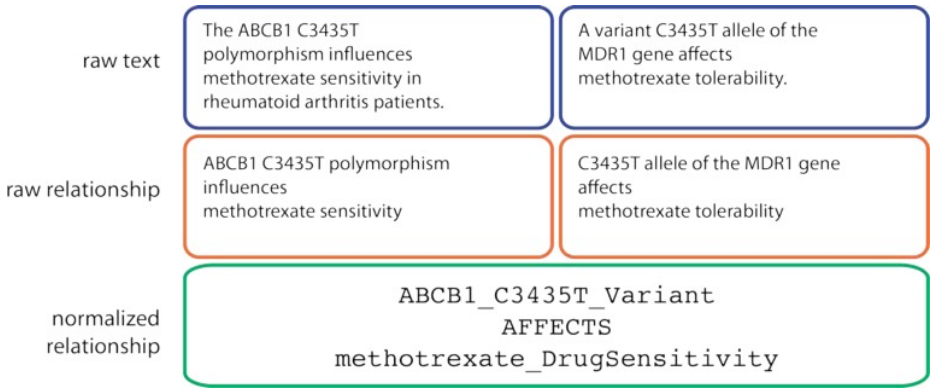
Our work builds on many decades of work in the biomedical domain to recognize relevant entities in text and understand how they are related to each other. Biomedical named entity normalization begins with biomedical named entity recognition, which has a long history within NLP [51,64,115]. Several different approaches exist for discovering named entities within biomedical text, some of which have been developed into full-fledged software solutions that can be downloaded and deployed with ease [69, 132]. Several authors have investigated which features provide the best performance in biomedical NER, and some have included distributional features among these [95, 141]. Our work also draws heavily on a history of efforts in automated biomedical ontology learning [79, 118, 148]. Finally, the problem of normalization was attacked head-on by the biomedical NLP community in the BioCreative competitions [65, 93], which led to a number of novel approaches for handling normalization. Many of these are included in PubTator, a project from the National Center for Biotechnology Information (NCBI), which provides downloadable annotations for all of Medline using a suite of state-of-the-art named entity recognition (NER) tools [149].

4.2 Augmenting an Existing Ontology

Many biomedical ontologies are built for specialized purposes. For example, the PHARE (PHArmacogenomic RElationship) ontology was built for the sole purpose of extracting and normalizing pharmacogenomic relationships from the medical literature (Figure 4.1) [20]. PHARE includes rules for recognizing relationships in sentences; it extracts pharmacogenomic relations with 80% precision. Coulet *et al* used PHARE to extract and normalize over 40,000 relationships among drugs, genes and phenotypes [19]. Later work used PHARE-normalized gene-drug relations to predict drug-drug interactions [107] (see Chapter 2).

Here we compare the structure of PHARE to the structure predicted using a popular method for unsupervised word similarity assessment called random indexing (Chapter 3, Section 3.4). We show that the word pair similarities predicted by random

Figure 4.1. An example of relation normalization using the PHARE ontology. Here two sentences that look very different on the surface are mapped to the same normalized "fact".

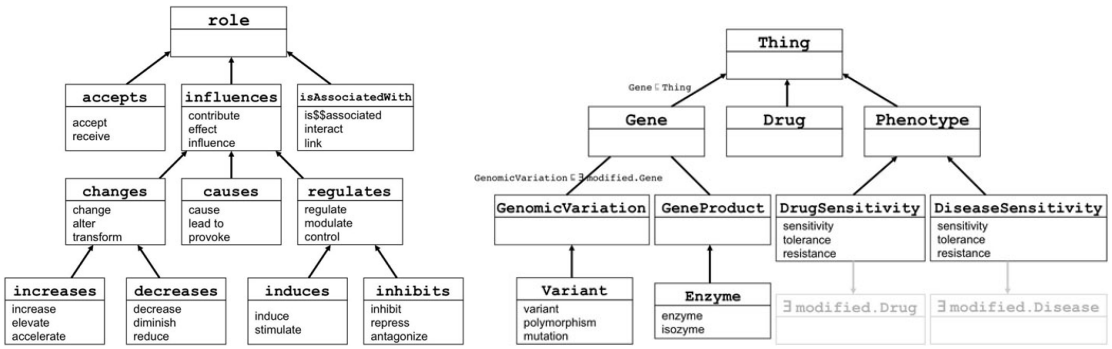


indexing correlate significantly with the words' relative positions within PHARE. We further examine the degree to which random indexing could be expected to *reproduce* PHARE; that is, to assign PHARE's word labels to the appropriate concepts and roles within the ontology. Although random indexing, at least as it was applied here, is not sufficient to fully reproduce the PHARE ontology, we conclude that it shows promise for identifying candidate terms for inclusion in future versions of the ontology.

4.2.1 Background: The PHARE Ontology

The (PHARE) ontology was created in 2010 by Adrien Coulet and colleagues at Stanford University. The researchers extracted approximately 40,000 raw relationships (verbs and nominalized verbs) among 3007 drugs, 41 genes and 4202 phenotypes from biomedical sentences and identified the 200 most frequent relationship types from within this set. They then manually merged similar relationship types into conceptual *roles* and organized these roles in a hierarchy [20]. They repeated this process for the nouns most often modified by drug and gene entities, such as *expression* and *polymorphism*, creating a hierarchy of modifier *concepts*. Finally, they defined a set of rules for application of the roles and concepts to drug, gene, and phenotype terms found in real English sentences. In particular, they limited the application of certain roles and concepts to certain classes of entities. (*Polymorphism*, for example, was only permitted to modify gene names, not drug or phenotype names.) The English words

Figure 4.2. Portions of the role (left) and concept (right) hierarchies of the PHARE ontology. These figures are taken from [19].



that map to each concept and role are called *labels*. The final version of PHARE consists of (a) a hierarchy of roles, (b) a hierarchy of concepts, and (c) a set of labels associated with each role or concept (Figure 4.2).

Recently, we investigated the degree to which pharmacogenomic relationships of interest described in PubMed sentences conformed to the grammatical structures PHARE is able to recognize. We found that although PHARE is excellent at extracting relationships of that form (nearly 100% sensitivity), its recall on interesting pharmacogenomic relationships as a whole is quite low. Of 72 sentences describing an inhibitory relationship between itraconazole and CYP3A4, for example, PHARE was able to extract only 2 relations. We concluded that to extract all useful pharmacogenomic relationships from Medline sentences, we would need to account for greater variability in sentence structure and phrasing than PHARE currently supports. As a first step in expanding PHARE's coverage, we decided to experiment with automated techniques for identifying other potential labels and their likely locations within PHARE.

4.2.2 Methods: Extracting Drug-Gene Sentences

We extracted all sentences from Medline 2012 that mentioned a drug and a gene and were between 4 and 50 words in length (approximately 95% of all sentences in Medline fell within this range). Drug and gene mentions were established using simple string matching and lexicons of drug and gene terms from PharmGKB [46, 62]. We included

only single-word drug and gene names for simplicity. We manually removed several common words that were accidentally included in the lexicons and were not actually drugs or genes (such as *enzymes*, *glycine*, and *vaccines* for drugs; *dehydrogenase*, *protease*, and *murine* for genes). The final lexicons included 1470 unique drug strings and 37,922 unique gene strings. Our final corpus consisted of 494,804 sentences.

4.2.3 Methods: The Semantic Vectors Package

We used the Java-based Semantic Vectors package [151] to construct vector representations of all words occurring at least three times in our corpus. Semantic Vectors is a convenient implementation of random indexing (see Section 3.4) based on Apache Lucene [86]. We varied the window size, vector dimension and seed length to evaluate how much these parameters affected our representations, and to find the combination that created the optimal vectors for our task. We also evaluated the means by which word order was encoded: *basic* vectors did not encode word order, *drxn* vectors encoded only the direction associated with a context word (before or after the target word), and *perm* vectors used permutations to encode the relative position of each context word relative to the target word. The degree of semantic similarity between two [unit-normalized] vectors was calculated using cosine similarity.

4.2.4 Methods: Calculating Concordance with PHARE

We wanted to see how well similarity scores for word pairs calculated using random indexing corresponded to those words' semantic relatedness within PHARE. Because we could not calculate the semantic relatedness of PHARE's concepts and roles directly using random indexing (since they are not English words), we instead calculated pairwise similarity scores between all concept labels, and independently, all role labels, in PHARE. We also wanted to determine whether a particular formulation of the semantic vectors we generated (such as a particular window width, dimension, or seed length) optimized the vectors' concordance with the structure of the PHARE ontology. We tested all combinations of the following: window widths 1, 3, and 5, vector dimensions 50, 100, 150, 300, 500, and 1000, word order encodings *basic*, *drxn*,

and *perm*, and seed lengths 4, 10, and 20.

We hypothesized that high similarity scores would correspond to close ontological relationships, meaning larger numbers of common ontological parents. For each label pair, we measured (a) the cosine similarity of its two labels' context vectors and (b) the number of common ontological parents for the labels in that pair (traversing the ontology upward until we reached the root node). We then repeated these measurements for all concept label pairs and, separately, all role label pairs in the ontology. We used the Kendall-Tau nonparametric correlation coefficient, specifically the implementation in R's *stats* package, to test the correlation between cosine similarity and number of common ontological parents separately for both concepts and roles. Unfortunately, the algorithm for calculating the Kendall-Tau coefficient is $\mathcal{O}(n^2)$; because the number of data points in our experiments was so large and the number of ties so high, and because we performed many different trials with different parameter values for our semantic vectors, calculating the full Kendall-Tau coefficient for each trial took too long. We therefore used 1000-point samples of our data and repeated the calculation of the Kendall-Tau coefficient 100 times for each sample; here we report the medians of those results. For all subsequent analyses, we used the best performing vectors, the specific formulation of which differed for roles and concepts.

4.2.5 Methods: Reassigning Labels Within the Ontology

Next, we evaluated how well random indexing could localize labels within the ontology. We removed each concept or role label from the ontology, one at a time. (Call the removed label L, and call its corresponding context vector V_L.) We then evaluated V_L's (a) mean and (b) maximum cosine similarity with the vectors for the remaining labels from each ontological group (a concept, if L was a label for a concept, or a role, if L was a label for a role). We ranked the groups according to their label vectors' similarity with V_L to ascertain which concepts or roles L was most likely to belong to. The result was a ranked list of candidate concepts or roles for each L. Ideally, the correct concept/role assignment for each label would rise to the top of its ranked list of candidates.

There are 228 concepts and 77 roles in the PHARE ontology. However, if a role was the passive-voice version of another role (*isInducedBy*, rather than *induced*) it was excluded from our analysis and its labels added to the active form version of the role. We therefore evaluated our performance on 54 of the 77 original roles.

4.2.6 Methods: Identifying New Terms for PHARE

Finally, we wanted to see if our semantic vectors could be used to efficiently augment the PHARE ontology. PHARE only includes a few hundred of the most common role and concept labels found in Medline; since its precision is only 80%, there are likely other reasonable labels that it missed. We wanted to see which other words might logically be added as labels to each concept and role. As a preliminary investigation of this possibility, we compared the vectors for each non-ontology term to all known label vectors from the ontology. For each concept or role label within the ontology, we found the top non-ontology term whose semantic vector best matched its own. This led to a ranked list of possible ontology candidates, ordered by their similarity to a current label in the ontology. For role labels, we restricted our list to verbs or nominalized verbs. For concept labels, we restricted our list to nouns (nominalized verbs, like *identification*, were also acceptable here). We then manually reviewed the lists for the most likely "ontology augmentation candidates".

4.2.7 Results: Optimizing Vector Construction

Figure 4.3 shows the results of our initial experiments to ascertain which type of semantic vector, generated by random indexing, best captured the structure of the PHARE ontology. As described in the Methods, we evaluated a variety of different vector types (window widths, dimensions, word order encodings and seed lengths) to see which led to the highest Kendall-Tau correlation between X = cosine similarity of label vectors and Y = number of common parents for those labels within the ontology. No matter what type of semantic vector we constructed, the correlation between X and Y was significant at the 95% confidence level; the best performing vectors had median correlations of 0.108 ($p = 0.00121$; concepts) and 0.165 ($p < 0.0001$; roles).

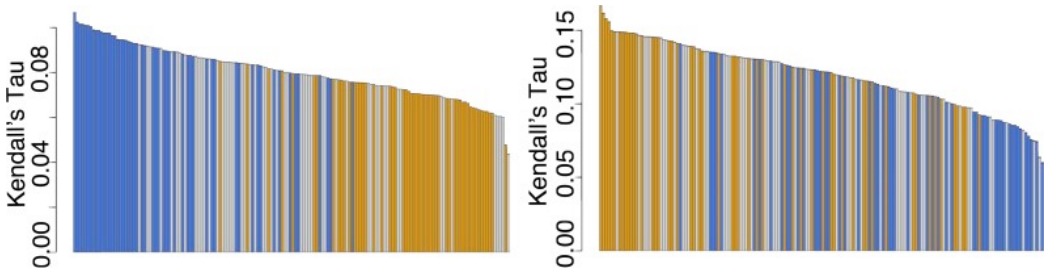


Figure 4.3. Bar plots of correlations between number of common parents in ontology and distributional similarity scores for (left) concepts and (right) roles. Each bar represents a different type of semantic vector. Orange bars represent vectors with width 1, gray width 3, and blue width 5.

Interestingly, the window widths associated with the best-performing vectors differed between concept and role labels. Concept labels correlated most highly with ontology position when a window width of 5 was used, while role labels were just the opposite; the correlation was highest with a window width of 1. Intuitively, this makes sense; concepts are nouns and roles are verbs, so one might speculate that most of the information about verbs is contained within the words immediately preceding and following them, while nouns' meaning depends on the more general "theme" of the sentence.

4.2.8 Results: Similar Concept and Role Labels

Some examples of highly similar concept and role labels, where similarity was assessed using the cosine similarity of the respective words' vectors, are shown in Table 4.1. The semantic relatedness of most of these word pairs is obvious. However, we do notice one peculiarity of the random indexing approach, which is that antonyms are not separated; in fact, antonyms have a high similarity score. This makes sense when one considers the nature of random indexing's context vector assembly process; there is no context where *downregulation* occurs in which *upregulation* could not also occur. However, it does raise a red flag in terms of random indexing's ability to reproduce the structure of the PHARE ontology; in the role portion of the ontology, for example, *induces* and *inhibits* live on separate branches. Random indexing could potentially

Table 4.1. Some close matches between known concept labels (top) and role labels (bottom) within the PhARE ontology.

Table 1. Examples of high-ranking pairs of *concept* labels from drug-gene sentences, ordered by cosine similarity.

Concept Label 1	Concept Label 2	Cosine Similarity Score
inhibition	suppression	0.983
downregulation	upregulation	0.982
incidence	prevalence	0.981
assessment	evaluation	0.977
pharmacokinetics	disposition	0.974
association	interaction	0.973
inactivation	inhibition	0.973
tolerability	safety	0.972

Table 2. Examples of high-ranking pairs of *role* labels from drug-gene sentences, ordered by cosine similarity.

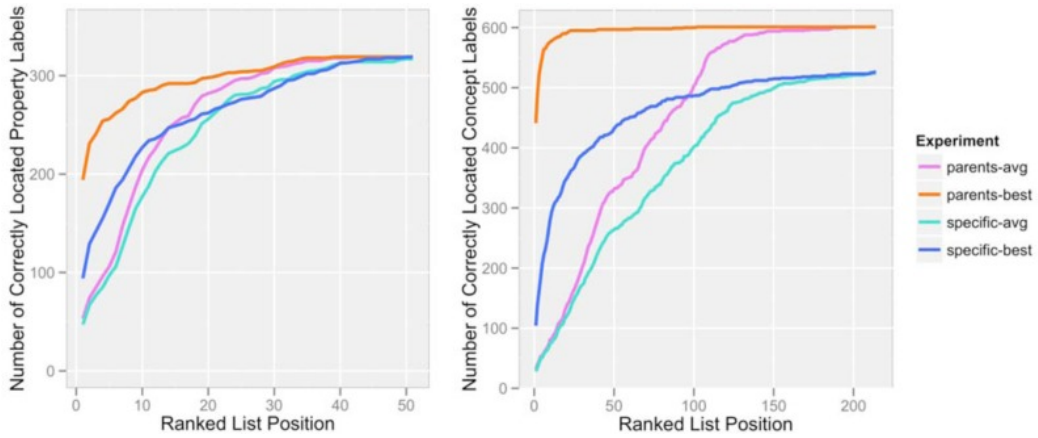
Role Label 1	Role Label 2	Cosine Similarity Score
investigate	examine	0.999
assess	evaluate	0.999
suggest	indicate	0.997
alter	affect	0.996
modulation	inhibition	0.992
suppress	stimulate	0.990
inhibit	prevent	0.988
catalyzed	catalysed	0.986

localize them only to within the same parent branch, *regulates*.

4.2.9 Results: Reassigning Labels to the Ontology

Our results for the "label reassignment" portion of our assessment are shown in Figure 4.4. The graphs display four lines: "specific-avg" and "specific-best" contain the number of correct concept/role assignments for labels that occurred within the top k items on their ranked lists (where k is the "Ranked List Position" value on the horizontal axis). The avg/best designation refers to the way in which the concept/role assignments were ranked; in the "avg" case, we calculated the test label's similarity to all labels within a concept/role and took the mean of those values as our match score for that concept/role. In the "best" case, we took the maximum of those values. Practically speaking, this means that if a test label was highly similar to only one member label of a concept/role, that concept/role would be ranked highly in the "best"

Figure 4.4. Correct concepts/roles found, by position in the ranked list. Separate graphs are shown for (left) roles and (right) concepts. The total number of concepts included here was 228 and the total number of roles was 54.



case but not in the "avg" case.

The "specific" vs. "parents" designation in Figure 4.4 refers to what we counted as a "hit". In the "specific" case, a concept/role label was considered correctly classified by position k only if its most specific matching concept/role appeared on the ranked list by that point. In the "parents" case, the most specific concept/role or one of its parent concepts/roles in the ontology could appear. We simply wanted to see whether some of our missed assignments were the result of the test label's being assigned to a more general super-class of the correct concept/role, which would be less of a problem than if it were assigned to an entirely incorrect part of the hierarchy.

Of the 602 concept labels we examined, 104 (17.3%) were correctly classified (i.e. the correct role was first on the ranked list) when the "best" method was used to assign the matches, and 17 (2.8%) were correctly classified when the "avg" method was used. This seems to indicate that often a label will be distributionally similar to some, but not all, other labels within its concept/role. Of the 319 role labels we examined, 94 (29.5%) were correctly classified when the "best" method was used and 25 (7.8%) were correctly classified when the "avg" method was used. If we relax our restriction on the concept/role assignment such that a parent of a given concept/role is also acceptable, 443 (73.6%) of concept labels are assigned correctly for "best" and 20 (3.3%) for "avg", and 194 (60.8%) of role labels are assigned correctly for "best"

and 31 (9.7%) for "avg".

In addition, performance increases if we consider assignments beyond rank position #1. Considering only the "best" assignment methods, since those seem to outperform "avg" at every turn, 234 (73.4%) of correct role labels and 420 (69.8%) of correct concept labels occur in the top 20% of the labels' ranked lists.

4.2.10 Results: Identifying New Ontology Terms

The final part of our analysis sought to identify those terms, not currently part of the ontology, that would make good candidates for inclusion as labels, and to localize those new labels within the ontology. Table 4.2 shows the best candidates, evaluated in terms of the criteria described in the Methods. Some of these terms, such as "tumors" and "combinations", were minor variants of other words that were already present in the ontology. In the case of both "tumors" and "combinations", their respective singular forms ("tumor", "combination") were already present as labels within the concepts assigned to them using random indexing. Findings like this boosted our confidence in random indexing considerably. Many of our findings from Table 4.2 are already under review for possible inclusion in future versions of PHARE.

However, so as not to over-sell this method to the reader, we have also included some errors in Table 4.2. "Capillary" was the highest-similarity word to "gel", for example, probably due to their common proximity to the relatively uncommon word "electrophoresis", but "gel"'s corresponding concept in the ontology is *TopicalFormulation*. Similarly, because "treated" is often used to describe chemical treatment of cell cultures in our corpus, it matched closely with "pretreated", while in PHARE "treated" is only permitted to describe a drug's treatment of a disease. A similar problem occurs for "incubation" and "treatment".

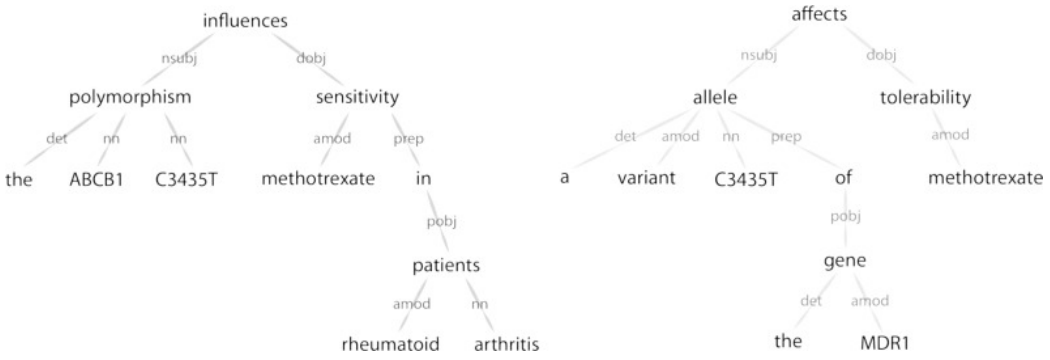
It is interesting to note that training semantic vectors on domain-specific corpora like our 500,000 drug-gene sentences seems to yield an increase in the specificity with which word senses are represented. For example, a context vector for the word "given" trained on text from the Wall Street Journal probably would not share much similarity with one for the verb "administered". However, because of the specific

Table 4.2. Some new concepts and roles discovered for the PHARE ontology. Top 15 ontology augmentation candidates for roles. Errors are denoted by a gray background. Top 15 ontology augmentation candidates for concepts. We include one example of a concept associated with the given concept label; there could have been more than one in the ontology, since labels are not unique for concepts. Errors are denoted by a gray background.

Candidate Label	Matching Concept Label	Cosine Similarity	Concept	Candidate Label's Occurrences in Corpus
participation	involvement	0.984	GeneProductFunction	246
enhancement	augmentation	0.97	Overexpression	1349
tumors	neoplasms	0.96	Cancer	3430
utility	usefulness	0.958	DrugEfficacy	371
combinations	coadministration	0.952	DrugTreatment	962
estimation	measurement	0.952	GeneAnalysis	221
superfusion	perfusion	0.949	DrugTreatment	150
identification	detection	0.943	PhenotypeAnalysis	575
comparable	similar	0.935	DrugAnalog	1472
assembly	formation	0.913	Synthesis	270
cascade	pathway	0.907	GenePathway	434
protocol	regimen	0.862	DrugTreatment	776
perturbation	modification	0.856	ChemicalModification	78
chronic	acute	0.822	DiseaseSeverity	8493
reactivation	recurrence	0.802	DiseaseRelapse	204
capillary	gel	0.621	TopicalFormulation	529
summary	conclusion	0.988	DrugEffect	494

Candidate Label	Matching Role Label	Cosine Similarity	Role (Active Form)	Candidate Label's Occurrences in Corpus
suppression	inhibition	0.985	inhibits	2809
ascertain	determine	0.972	demonstrates	145
abrogated	abolished	0.960	suppresses	517
impact	influence	0.940	influences	1292
infused	injected	0.935	administers	911
given	administered	0.928	administers	5345
uptake	transport	0.926	transports	4813
formed	generated	0.881	produces	1128
utilizing	using	0.871	uses	325
display	exhibit	0.866	has	378
underwent	received	0.839	accepts	897
verified	confirmed	0.835	demonstrates	148
documented	established	0.794	demonstrates	567
devised	developed	0.755	produces	48
maintain	sustain	0.749	demonstrates	417
pretreated	treated	0.971	treats	1621
incubation	treatment	0.923	treats	2527

Figure 4.5. Dependency parses for the two example sentences shown in Figure 4.1. Because the structure of these sentences is so similar, one could conceive of using distributional semantics methods to establish an alignment between them, thus performing a task akin to normalization without the use of an ontology.



contextual cues found in drug-gene sentences, "given"'s closest vector neighbor is indeed "administered". This is because, in drug-gene sentences, to "give" something (a rat, a human) a drug is to administer that drug. There are not many other contexts within these sentences in which "given" is used. The same argument is probably also true for "cascade" and "pathway" and "uptake" and "transport" (Table 4.2).

4.2.11 Discussion: Interpreting Correlation Strength

The relatively weak correlation between the proximity of word labels within the PHARE ontology and their vector space similarities is a strong indication that there is more information in the ontology than can be captured purely by looking at how words are used in context. For example, several of the ontological concepts and roles contained labels that were common terms, like "find", that gained additional specificity by the rules PHARE provides on how they are to be applied to real biomedical sentences. Our investigations here take none of these *word sense* factors into account, aside from our selection of a training corpus in which the word senses in question are limited. To our semantic vectors, "established" (as in "established methods") is the same as "established" (as in "established a new technique for"). This is a major limitation of the distributional approach used here; ambiguities like this were one reason PHARE was created.

However, it is interesting to consider the degree to which these imperfections matter for real biomedical applications. For example, consider the two dependency parses shown in Figure 4.5. (A dependency parse is one technique for representing the deep grammatical structure of a sentence.) These parses are for the two example sentences shown in the normalization example in Figure 4.1. Noted biomedical relation extraction algorithms like RelEx [36] already use dependency parses in their analysis, but they apply manually-generated rules to them to extract relations of interest. (PHARE was also inspired by Coulet et al's observation of common structural "motifs" in dependency-parsed biomedical sentences.) We immediately notice that the sentences in this figure are structurally similar, and that we might conceive of aligning the two dependency graphs and using vector space representations of word meanings to compare the quality of these alignments. This assessment of the sentences' similarity would perform a task akin to normalization. In this case, even if "arthritis" and "tolerability" somehow ended up with similar distributional representations, it wouldn't matter for the purposes of assigning the alignment score because they exist in different grammatical "places" within the two graphs. Alignment-based approaches like this are already common in the computer science literature; for example, in automatic essay grading [91] and entailment recognition [45]. So, practically speaking, even a weak distributional "signal" might be enough for some interesting applications.

4.2.12 Discussion: Limitations of this Approach

Our approach suffers from a few additional limitations that are worth mentioning. First, our corpus consisted of individual Medline sentences containing drug and gene names; we did not consider additional contextual cues from the rest of the abstracts. We did this in the interest of building semantic vectors that were as domain specific as possible; however, the lack of additional domain cues probably hurt us, especially with respect to concept assignments (which, as we observed, preferred wider bin widths). Second, as briefly alluded to earlier and as lamented frequently in the distributional semantics literature, our techniques did not capture the opposing nature of antonyms. As far as we know, there is no way to reliably distinguish antonyms using distributional

means.

In addition, our evaluation of word similarities, and our assignment of word labels to concepts and roles within the PHARE ontology, ignored much of the ontology's deeper structure. For example, some concepts are only permitted to modify phenotypes, while others are only permitted to modify genes. We ignored this structure and compared the labels from these different concepts directly. This was done in the interest of quick exploration and simplicity, but restricting our comparisons to labels from specific concepts/roles could very well have improved our performance reassigning labels to PHARE. However, since the point of our study was to see how much the structure of PHARE could be captured without human intervention, we did not choose to restrict our analysis in this way.

And finally, label assignments within PHARE are unique for roles but not concepts. This meant that a given label could have more than one concept associated with it, and it probably explains the huge increase in performance we experienced when we included parent concepts in our analysis in Figure 4.4.

4.2.13 Summary: Ontology Building with Random Indexing

Random indexing produces vector representations of words that correlate significantly with these words' positions within a biomedical ontology. In this section, we showed that word pairs' semantic vectors became increasingly similar as the words shared more common parents within the PHARE ontology. We also discovered that words could be assigned to reasonable concepts and roles within the ontology if we scored them based on their maximum similarity with other word labels within a given concept or role. We expanded this approach to assign some new word candidates that are not currently in the ontology to their most likely ontological locations.

Although these representations do not capture all of the information contained in the ontology, they have several advantages. First, they are quick and easy to produce, and can easily be adapted to different corpora (Medline, other biomedical text, or specialized subsets of Medline such as the drug-gene sentences we examined here). Second, they seem to capture much of the semantic meaning of individual

words, at least as those words are represented within the PHARE ontology, and they can be used to quickly and easily "bootstrap" connections to other words in the corpus that could be suitable for inclusion in the ontology. They can also provide a rough sense of where those words should be located within the ontology. And finally, and most importantly, construction and evaluation of these vectors requires no manual rule-making or annotation; the vectors are learned in an unsupervised manner from unlabeled text corpora. Although our explorations here are preliminary and much work remains to be done to fully establish the role of distributional semantics methods within biomedical text mining, increasing interest in this field within the biomedical community could lead to exciting new applications in the areas of named entity recognition, concept normalization, and specialized ontology building within bioinformatics.

4.3 Normalizing Biomedical Entity Names

One might hypothesize that distributional similarity (how words and phrases are used in context) approximates the kind of similarity we as humans consider when building structured lexicons and ontologies. If this is true, these distributional methods could provide scalable alternatives to ontologies for the normalization of biomedical entity names, such as drug, gene, and disease names, in biomedical text.

Here we apply the distributional semantics software package `word2vec` to normalize drug/chemical, gene/protein, and disease names based on their usage patterns in Medline text, evaluating its performance against gold standard synonym sets from PharmGKB, UMLS, NCBI and DrugBank. We investigate three different versions of the original `word2vec` implementation, as well as a newer variant of `word2vec` called dependency `word2vec`. We develop a set of heuristics for efficiently using this technology to find synonyms.

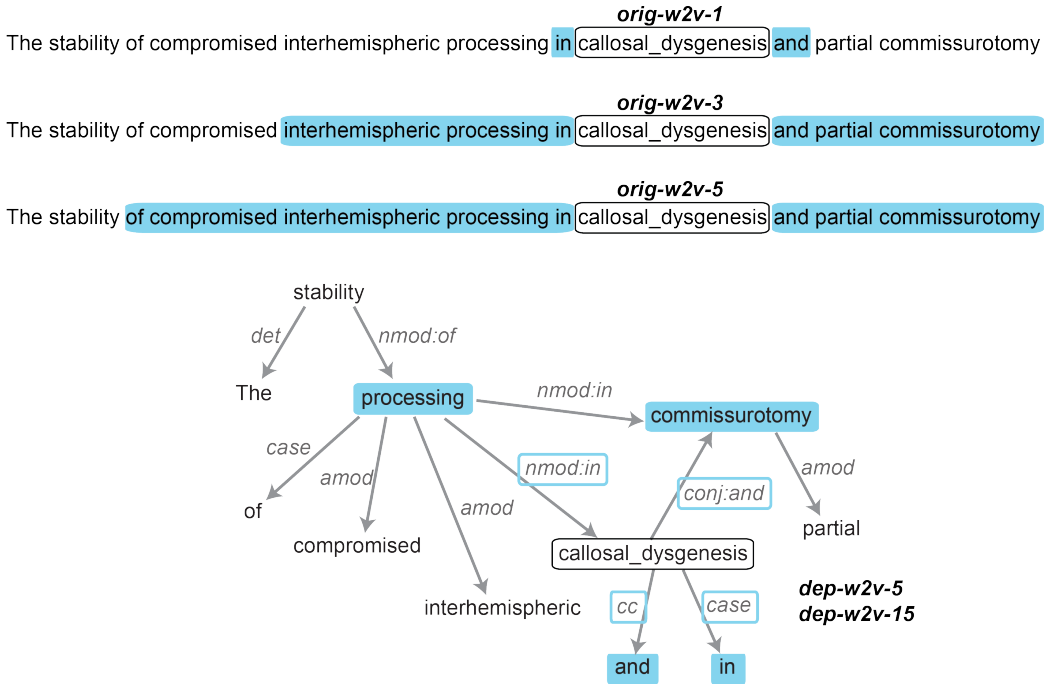


Figure 4.6. Context used for different variants of word2vec.

4.3.1 Methods: Word and Phrase Vectors with `word2vec`

For this project, we used the `word2vec` package created by Mikolov *et al* [88] to build vector representations of all terms in our two corpora (defined below). The original implementation of `word2vec` uses a linear context window, meaning that all terms that occur within a window width w of a target term in a corpus will contribute to the context for that term (see Figure 4.6). The `word2vec` package has a number of settable parameters including vector length, the size of the context window, and the choice of model (CBOW vs. skip-gram). In our experiments, we used the skip-gram model with vector length 100 and a context window width of 1, 3, or 5. We call these three models *orig-w2v-1, orig-w2v-3,* and *orig-w2v-5.*

Recently, other researchers have pointed out that terms can be similar in many different ways, and that a linear context window only creates one type of embedding. We therefore implemented an alternative algorithm, dependency `word2vec` [71], that produces markedly different embeddings from original `word2vec`. Instead of a linear

context window, dependency `word2vec` uses terms connected to a target term in a dependency graph, along with their grammatical relationships to the target term, as its notion of context (Figure 4.6). We used two different choices for the "negative sampling" parameter of dependency word2vec, since the original `word2vec` software uses a value of 5 for this parameter but the code for dependency `word2vec` uses 15 by default. We therefore call these two models *dep-w2v-5* and *dep-w2v-15*.

Because `word2vec` constructs a separate vector for each token in the text, there is no intuitive way to compose the vectors for "diabetes" and "mellitus" into a combined vector for the term "diabetes mellitus". However, many biomedical synonyms have different numbers of tokens; "CYP3A4" (one token), for example, is a synonym for "cytochrome P450 3A4" (three tokens). To handle this, we need some way of concatenating multi-word entities into single tokens before running `word2vec`. Below, we describe how we addressed this problem for each of our two corpora.

4.3.2 Methods: Construction of PubTator Corpus

PubTator [149] provides downloadable annotations for all of Medline using a suite of state-of-the-art named entity recognition (NER) tools. It also provides the full titles and abstracts for those Medline records in which at least one biomedically relevant entity, such as a chemical, gene, or disease, was annotated. There are approximately 16.5 million such abstracts in the corpus. Annotations are updated daily. Our version of the PubTator annotations was downloaded on November 29, 2015.

We used the PubTator annotations to concatenate phrases corresponding to annotated biomedical entities; for example, the phrase "typhoid fever", if identified as an entity by PubTator, was changed to "typhoid_fever" (using the underscore). This concatenation step was performed first, before any additional preprocessing was done.

We then constructed the final PubTator corpus by tokenizing the text of each title or abstract in every PubTator record using the Stanford CoreNLP toolkit[1] [85], concatenating the resultant tokens using single spaces, and lowercasing all of the text. Each processed record was then added to the corpus file on its own line.

[1] http://stanfordnlp.github.io/CoreNLP/

4.3.3 Methods: Constructing Gold Standard Synonym Sets

Because we are testing whether distributional similarity approximates the kind of similarity we as humans consider when deciding whether terms are synonyms, a natural source of gold standard evaluation data for our project is human-curated lexicons and ontologies. We consider three types of entity in our evaluation: drug/chemical names ("chemicals"), gene/protein names ("genes") and disease/phenotype names ("diseases"). All evaluations were performed using vectors built on the PubTator corpus. We produced gold standard synonym sets from two different sources for each entity type, keeping only those terms for which both original and dependency `word2vec` could construct vectors, meaning that they occurred at least 5x in the PubTator corpus. The details of these synonym sets are in Table 4.3.

4.3.4 Methods: Vector Construction and Comparison

Through these experiments, we wanted to discover both (a) what method of creating vectors best recapitulates human notions of biomedical synonymy, and (b) how vectors should be compared to each other to best distinguish synonyms from non-synonyms. Our technique for comparing different methods was to create distributions of similarity scores separately for synonym and non-synonym term pairs and look for the method for which the Kolmogorov-Smirnov (KS) statistic, a measure of dissimilarity between two distributions, was highest. We used all five methods of creating distributional term vectors shown in Figure 4.6 (*orig-w2v-1, orig-w2v-3, orig-w2v-5, dep-w2v-5, dep-w2v-15*) and compared the vectors using one of three techniques:

1. **Cosine similarity.** The cosine similarity of two vectors v and w is defined as

$$\sigma(v, w) = \frac{v \cdot w}{\|v\|\|w\|}$$

   where $\|\cdot\|$ denotes the Euclidean norm. This technique used the raw value of the cosine similarity as the similarity score between v and w.

2. **Rank + reverse rank.** Here the vector for term i in the vocabulary, v_i, was compared to all other vectors v_j in a background set and the list of all similarity

Table 4.3. Details of the construction of synonym sets for each entity type. Each synonym set consists of all terms corresponding to a single database identifier for which vectors could be built on the PubTator corpus.

Name	# synonym sets / # synonym pairs	Number of total terms (# annotated by PubTator)	Entity type	Description
pharmgkb-chemical	1451 / 14,830	6798 (5610; 82.5%)	drug/chemical	The PharmGKB drug lexicon [150] originated from DrugBank [152] and Micromedex, and also contains some chemical names from PubChem [8] and NDF-RT[2], as well as a few additional names added manually by the PharmGKB curators. All terms corresponding to the same PharmGKB ID were considered synonyms and constituted one synonym set.
pharmgkb-gene	10,137 / 36,080	28,623 (27,570; 96.3%)	gene/protein	The PharmGKB gene lexicon [150] is based on terms from EntrezGene [83] and HGNC [110]. All terms corresponding to the same PharmGKB ID constituted one synonym set.
pharmgkb-disease	1841 / 19,466	7803 (7491; 96.0%)	disease	The PharmGKB disease lexicon [150] is based primarily on MeSH[3] [6], but also contains some terms from SNOMED[4] and MedDRA[5]. All terms corresponding to the same PharmGKB ID constituted one synonym set.
drugbank-chemical	1697 / 9701	6146 (5334; 86.8%)	drug/chemical	Each drug record in DrugBank [152] contains a main name, DrugBank ID, set of synonyms for the main name, and set of product names for the drug. All synonyms and product names for a given DrugBank ID together constituted one synonym set.
ncbi-gene	42,388 / 109,821	62,128 (52,583; 84.6%)	gene/protein	The NCBI gene_info.txt file contains gene information from NCBI's Gene database[6]. All terms corresponding to a single database identifier constituted one synonym set.
icd9-disease	1564 / 10,167	5503 (5247; 95.3%)	disease	We parsed the 2015AB version of UMLS, specifically the MRCONSO.RRF file, and found all concept unique identifiers (CUIs) mapping to at least one ICD9 code. Since ICD9 is a disease ontology, these CUIs corresponded to known diseases or disease categories. We then traversed the file again and found all alternate strings mapping to these same CUIs. All strings mapping to the same CUI constituted one synonym set.

scores was ordered from largest to smallest. The rank of each term k was found relative to term i and the reverse rank of term i to term k (on term k's ranked list) was also found. The similarity score for i and k is then the sum of the rank and reverse rank.

3. **Min(rank, reverse rank).** The same as score 2, except that instead of summing the rank and reverse rank, we take the minimum. So if term i ranks term k near the top of its list (low rank) but k puts i near the bottom (high rank), the score for that pair is term i's rank for term k.

Because the synonym sets were of different sizes, we had to be careful about how we constructed the synonym and non-synonym score distributions. Let S_i be a synonym set of size N_i. The similarity scores contributing to the synonym score distribution for this set are $\sigma(S_{ij}, S_{ik})$, where $j = 1, \ldots, N_i$, $k = 1, \ldots, N_i$, and $j \neq k$, where S_{ij} is the jth member of the ith synonym set. Note that this double-counts each synonym pair, but does not change the score distribution.

To create the non-synonym score distribution, we created parallel sets R_i of randomly chosen terms from a background set. The background sets were those terms annotated [by PubTator] with the same entity class as S_i. We then added the scores $\sigma(S_{ij}, R_{ik})$, where $j = 1, \ldots, N_i$, $k = 1, \ldots, N_i$, and $j \neq k$, to the non-synonym score distribution.

This method creates a "worst case scenario", where the non-synonym terms are close in meaning to the synonym terms, and also reflects a potential common use case for these techniques, in which NER software is first used to create a filtered set of candidate terms and then a distributional semantics method like `word2vec` is applied to normalize them.

4.3.5 Methods: Performance Metrics for Synonym Finding

Using the best-performing vector construction and comparison methods, we calculated similarity scores for (a) correct synonym pairs and (b) incorrect synonym pairs – these were pairs where both terms were present in the synonym set file, but were from different synonym sets. We calculated precision, recall and $F_{0.5}$ numbers for synonym

retrieval for each of our six synonym set types (pharmgkb-chemical, icd9-disease, etc.), where

$$F_{0.5} = (1 + 0.5^2)\frac{\text{precision} \cdot \text{recall}}{(0.5^2 \cdot \text{precision}) + \text{recall}}$$

is a commonly used measure similar to the F_1 score except that it weights precision higher than recall. We evaluated 60 different score cutoffs and found a score threshold that optimized $F_{0.5}$ for each set.

4.3.6 Results: Best Practices for Constructing and Comparing Vectors

A summary of the Kolmogorov-Smirnov statistics comparing synonym and non-synonym similarity score distributions is shown in Table 4.4. Regardless of scoring method and synonym set type, the *orig-w2v-5 vectors* (linear context window of width 5) performed best at identifying biomedical synonyms[7]. These were followed in basically consistent order by *orig-w2v-3* and *orig-w2v-1*, with the dependency-based vectors performing worst on nearly all synonym sets.

As for scoring method, although the obvious choice would be to establish a threshold for synonymy based on the value of the cosine similarity score, this method turns out to perform rather poorly. The best choice for separating synonym and non-synonym distributions is to use the *min(rank, revrank)* scoring method. This is apparent from the plots in Figures 4.7 and 4.8, which show distributions of similarity scores for synonyms and non-synonyms using the *orig-w2v-5* vectors on all different synonym set types, and the cosine similarity (Figure 4.7) and *min(rank, revrank)* (Figure 4.8) scoring methods.

Also apparent from these plots is the fact that some entity types have more recognizable synonym pairs than others. In general, disease synonyms are the easiest to recognize using distributional similarity, followed by chemical synonyms and then gene synonyms.

[7]There were only two exceptions: *orig-w2v-3* edged out *orig-w2v-5* on the *pharmgkb-chemical* synonym sets using the *min(rank, revrank)* scoring method, and *orig-w2v-1* won on the *drugbank-chemical* sets using the *min(rank, revrank)* score. However, the differences here were so small that we concluded the *orig-w2v-5* vectors perform best in general.

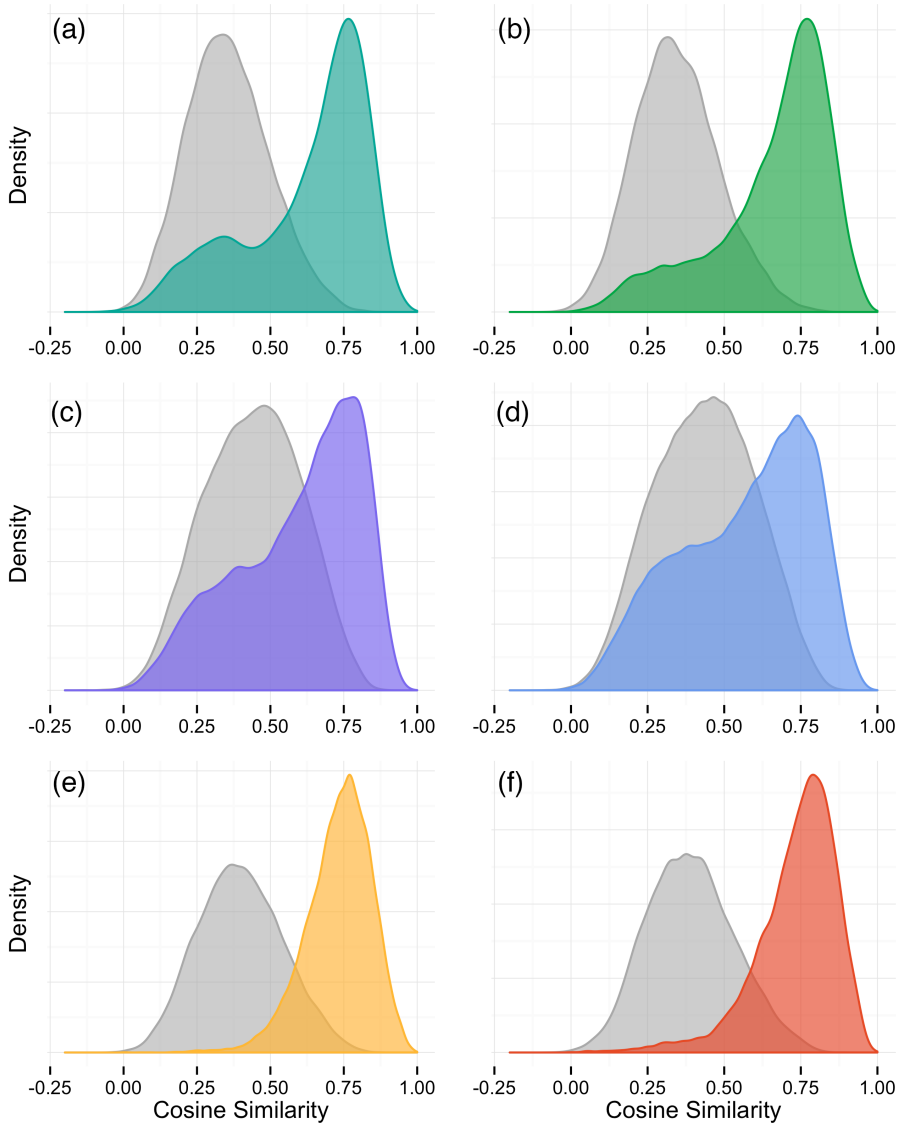


Figure 4.7. Distributions of cosine similarities of synonyms (colored densities) for different entity types, relative to those entities' cosine similarities with randomly-chosen words from the corpus (gray densities). (a) pharmgkb-chemical, (b) drugbank-chemical, (c) pharmgkb-gene, (d) ncbi-gene, (e) pharmgkb-disease, (f) icd9-disease.

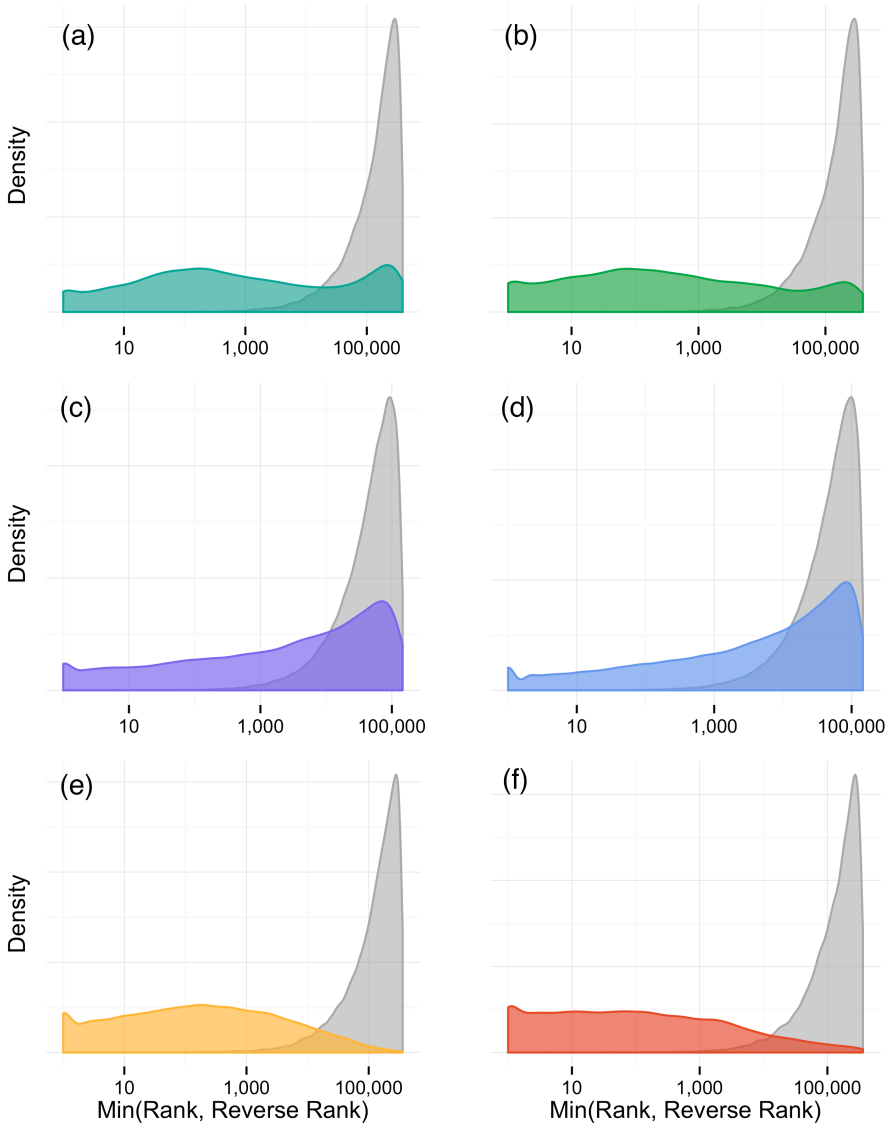


Figure 4.8. Distributions of min(rank, revrank) for synonym pairs (colored densities) for different entity types, and the same scores relative to randomly-chosen words from the corpus (gray densities). (a) pharmgkb-chemical, (b) drugbank-chemical, (c) pharmgkb-gene, (d) ncbi-gene, (e) pharmgkb-disease, (f) icd9-disease.

Table 4.4. Kolmogorov-Smirnov statistics for synonym vs. non-synonym distributions, for each type of synonym, for each vector type. We considered three types of scoring methods: (a) cosine similarity, (b) rank + reverse rank, (c) minimum of rank and reverse rank.

COSINE SIMILARITY	orig-w2v-5	orig-w2v-3	orig-w2v-1	dep-w2v-5	dep-w2v-15
pharmgkb-chemical	0.617	0.573	0.443	0.365	0.354
drugbank-chemical	0.691	0.646	0.543	0.446	0.441
pharmgkb-gene	0.392	0.347	0.250	0.181	0.172
ncbi-gene	0.334	0.290	0.206	0.147	0.139
pharmgkb-disease	0.797	0.730	0.587	0.544	0.543
icd9-disease	0.785	0.727	0.580	0.518	0.519
RANK + REV. RANK	orig-w2v-5	orig-w2v-3	orig-w2v-1	dep-w2v-5	dep-w2v-15
pharmgkb-chemical	0.665	0.639	0.545	0.487	0.490
drugbank-chemical	0.742	0.713	0.640	0.572	0.578
pharmgkb-gene	0.468	0.428	0.338	0.276	0.282
ncbi-gene	0.412	0.371	0.294	0.248	0.255
pharmgkb-disease	0.833	0.786	0.678	0.679	0.683
icd9-disease	0.833	0.782	0.675	0.654	0.666
MIN(RANK, REV. RANK)	orig-w2v-5	orig-w2v-3	orig-w2v-1	dep-w2v-5	dep-w2v-15
pharmgkb-chemical	0.784	0.789	0.783	0.763	0.775
drugbank-chemical	0.846	0.844	0.847	0.821	0.828
pharmgkb-gene	0.720	0.711	0.698	0.686	0.697
ncbi-gene	0.693	0.687	0.680	0.674	0.689
pharmgkb-disease	0.950	0.940	0.917	0.921	0.927
icd9-disease	0.941	0.932	0.908	0.894	0.905

Table 4.5. Performance metrics for synonym retrieval for the six synonym set types against a background of terms tagged with the same entity type by PubTator.

Synonym set type	Optimal threshold	Precision	Recall
pharmgkb-chemical	6	0.19	0.08
drugbank-chemical	4	0.20	0.08
pharmgkb-gene	3	0.08	0.02
ncbi-gene	3	0.04	0.01
pharmgkb-disease	7	0.44	0.15
icd9-disease	6	0.53	0.18

We therefore conclude that the optimal way to recognize biomedical synonyms, at least from among the options we tried, is to build vectors using a linear context window of width 5, and to establish a cutoff for synonymy of two terms based on the *min(rank, revrank)* score.

4.3.7 Results: Synonym Finding for Chemicals, Genes and Diseases

Figure 4.9 shows precision, recall and $F_{0.5}$ results for all six synonym set types, for 60 different *min(rank, revrank)* score thresholds. The optimal *min(rank, revrank)* thresholds, based on $F_{0.5}$, for the six types, along with precision and recall numbers at those thresholds, are shown in Table 4.5. In general, performance at recognizing synonyms against a background of similar entities is quite low. Precision and recall were highest for the *icd9-disease* dataset, at 0.53 and 0.18, respectively, and lowest for the *ncbi-gene* dataset, at 0.04 and 0.01 respectively. It appears that one can tell disease syonyms are similar (and different from other diseases) much more easily from context than one can for genes/proteins.

4.4 Learning a Radiology Lexicon

In the previous section, we investigated using `word2vec` to find synonyms for biomedical entities such as drugs, genes and phenotypes. We established a set of heuristics

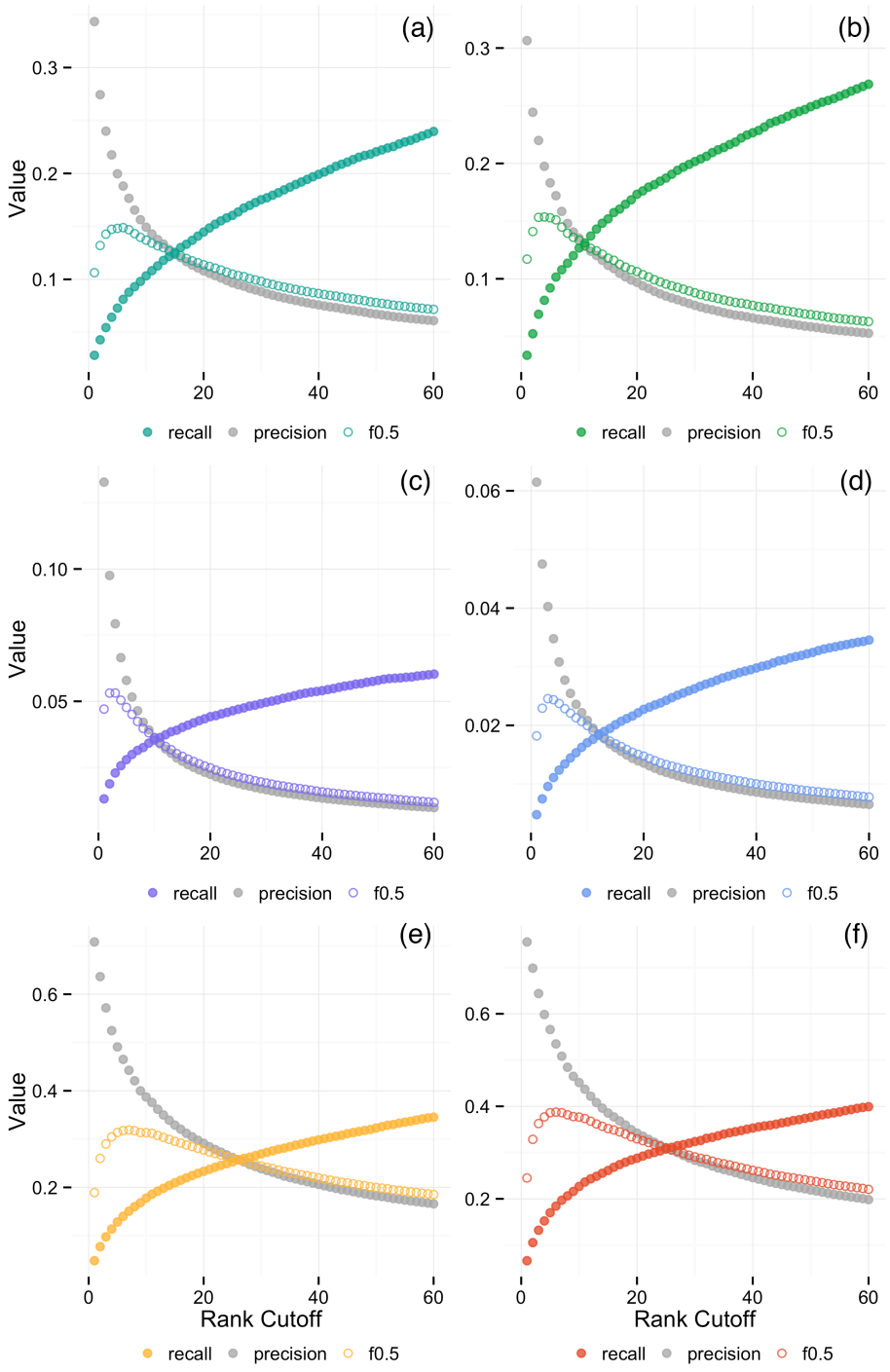


Figure 4.9. Performance measures and optimal threshold cutoffs for recognizing different entity types. (a) pharmgkb-chemical, (b) drugbank-chemical, (c) pharmgkb-gene, (d) ncbi-gene, (e) pharmgkb-disease, (f) icd9-disease.

around using distributional semantics to find synonyms. Here we apply the best-performing technique to a corpus of clinical radiology notes to recreate RadLex, a human-curated ontology of radiology terms, based solely on contextual usage patterns. Our approach identifies synonym pairs found in RadLex with 61% precision and 30% recall. We use it to identify a total of 19,770 predicted new terms for RadLex out of a possible 775,248 unique strings in our corpus, and to connect these new predictions to their closest synonyms within RadLex.

4.4.1 Methods: Radiology Note Corpus

Our radiology corpus consisted of the RadCore [42] database and a corpus of additional radiology reports from Stanford University. RadCore is a multi-institutional database of radiology reports aggregated in 2007 from three major healthcare organizations: Mayo Clinic (812 reports), MD Anderson Cancer Center (5000 reports), and Medical College of Wisconsin (1,893,819 reports). To these we added 4,056,227 radiology reports of 564,210 patients from Stanford Hospital and Clinics Epic electronic health record system since 1998. These were obtained from the STRIDE (Stanford Translational Research Integrated Database Environment) database at Stanford [80]. STRIDE is a research and development project to create a standards-based informatics platform supporting clinical and translational research.

The preprocessing of the raw radiology notes to create the radiology corpus was handled much in the same way as for the PubTator corpus, except that because we did not have entity annotations, we needed another way of concatenating phrases. We therefore used `word2phrase`, a tool provided by the creators of `word2vec`, to concatenate likely phrases in this corpus. We used the default parameters for `word2phrase`.

4.4.2 Methods: Recreating RadLex

RadLex is a lexicon of 46,340 radiology concepts and associated terms created over a decade by over 30 professional radiology organizations [67]. The lexicon is not explicitly designed for the purpose of normalizing synonyms, but it does contain 636

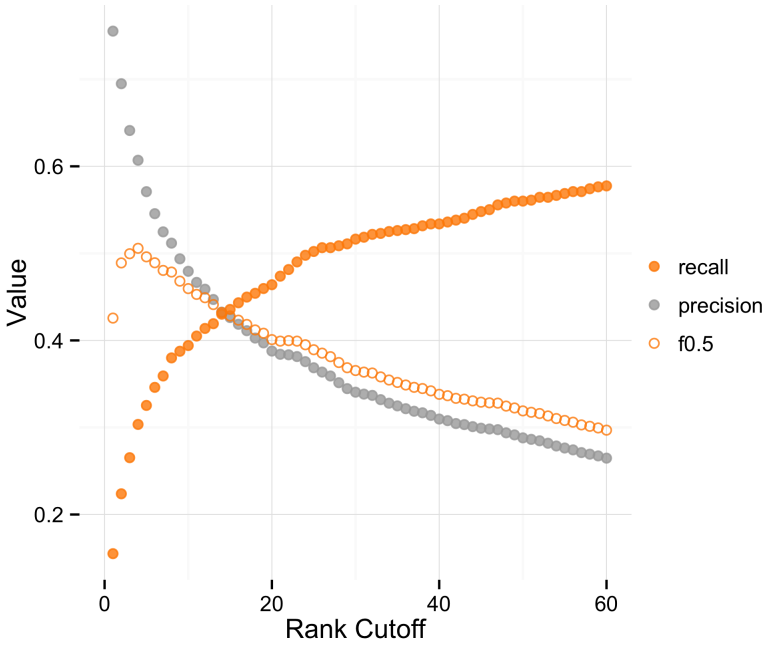


Figure 4.10. Performance measures and optimal threshold cutoff for RadLex synonyms.

concepts that map to multiple strings. We repeated the calculation of performance metrics for these synonyms to get an idea of what kind of performance to expect when finding synonyms for all of RadLex. After evaluating 60 different score cutoffs and finding the score threshold that optimized $F_{0.5}$, we built a network representation of all the predicted new synonym pairs and visualized the results in Cytoscape [133].

4.4.3 Results: RadLex Synonym Finding

Figure 4.10 shows a plot similar to those in Figure 4.9 for synonym finding in RadLex. The maximum value of $F_{0.5}$ is 0.51 at a threshold of 4. This choice of threshold leads to a precision of 0.61 and recall of 0.30. These numbers are much higher than for the PubTator dataset, which probably reflects the fact that we are comparing against a background of all terms vs. those tagged with the same entity type, as well as the fact that the terms in RadLex are intended, by design, to be used in very specific contexts.

Using 4 as a threshold for *min(rank, revrank)*, we connected all terms at that score or below. Figure 4.11 shows two small network representations of new predictions

and their closest RadLex terms. The first shows that *heterogeneous_enhancement* is connected to the spelling variant *heterogenous_enhancement*, as well as *inhomogeneous_enhancement* and *enhances_heterogeneously*, neither of which is currently in RadLex but both of which are indeed synonyms for *heterogeneous_enhancement*. Of course, Figure 4.11(a) also illustrates one of the classic errors associated with distributional similarity finding: it has trouble distinguishing antonyms, which are frequently used in similar contexts. *Homogenous_enhancement*, a known RadLex term, is predicted to be a synonym of *heterogeneous_enhancement*, when in fact they are opposites.

Figure 4.11(b) examines a set of anatomical terms surrounding the RadLex term *brainstem*. Interestingly, the term *brainstem* is connected within this network to all of its component parts: the pons, medulla, and midbrain (or mesencephalon, another synonym identified by word2vec), as well as to the concatenation variant *brain_stem*. There are many connections in this subnetwork that involve relationships other than total synonymy, such as the fact that *dorsal_pons* and *ventral_pons* are both component parts of their connected RadLex term *pons*.

A list of the top 40 new candidate terms for RadLex, as well as their closest matching terms in RadLex, is shown in Table 4.6. Of the top 40 hits, by our estimation, 29 are synonym pairs (or close enough that substituting one term for another would not change meaning, though it might lead to an ungrammatical sentence). The rest illustrate most of the potential failure modes associated with distributional techniques for synonym finding. For example, the top hit, *rt/lt*, actually reflects two errors. The terms *RT* and *LT* are frequently used in radiology notes to denote *right side* and *left side*, and both should probably be in RadLex. However, *rT* is also used to mean the *rostrotemporal auditory area*, which is why that term was present in RadLex and *LT* was not. Connecting *LT* to *RT* would normally have produced an antonym error, but it also reflects a situation where the same abbreviation is used to refer to multiple entities, and distributional similarity cannot catch that.

Rows 2 and 7 reflect cohypernym errors. Both indicate that the term *middle_finger* should be included in RadLex, which is probably true, but they assign it to the synonym terms *ring_finger* and *index_finger*, which are cohypernyms of the term *finger*

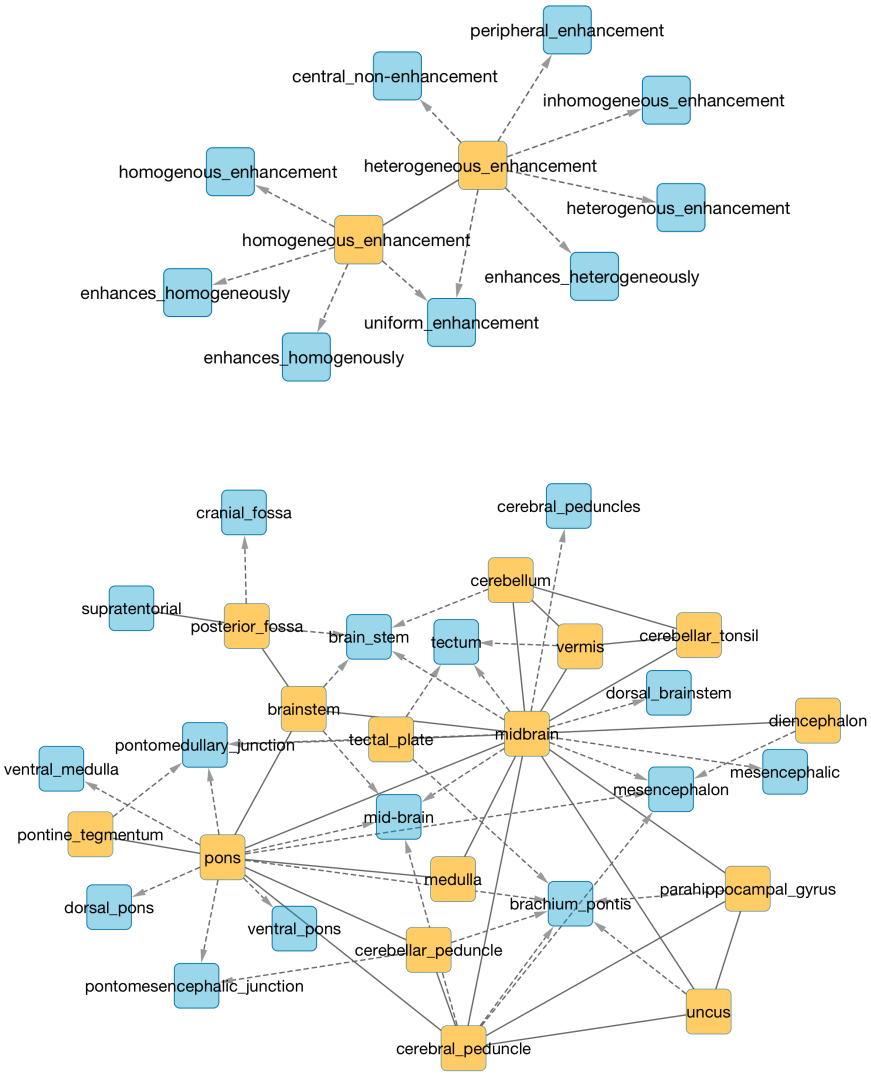


Figure 4.11. Two network representations of new connections to existing RadLex terms found through distributional similarity using 4 as a threshold for *min(rank, revrank)*. All of these examples come from the pairs in Table 4.6 and the other new and old synonymy connections that surround them. Nodes that are currently in RadLex are shown in orange, and new predictions are shown in blue. If an edge connects two nodes that are both in RadLex, it is shown as a solid line with no arrow. If it is connecting a current RadLex node to a new prediction, it is a dotted line with an arrow pointing toward the new term.

Table 4.6. A list of the "top hits" for inclusion in RadLex. These are terms where *min(rank, revrank)* to one or more terms in RadLex was at or below 4. The cosine similarities of these terms to their top matching RadLex terms are also shown.

	Existing RadLex term	New term	Cosine similarity	Synonyms or close?
1	rt	lt	0.988	N (ant.)
2	ring_finger	middle_finger	0.987	N (cohyp.)
3	int_rot	pel_max	0.984	N (error)
4	ureterovesical_junction	ureterovesicular_junction	0.984	Y
5	brainstem	brain_stem	0.982	Y
6	well-circumscribed	well_circumscribed	0.981	Y
7	index_finger	middle_finger	0.981	N (cohyp.)
8	corpus_luteum	corpus_luteal	0.980	Y
9	gastric_pull-through	gastric_pull-up	0.979	Y
10	optic_chiasm	chiasm	0.977	Y
11	fifth_metatarsal	5th_metatarsal	0.977	Y
12	fat_saturation	fat_sat	0.977	Y
13	occipital_lobe	cerebellar_hemisphere	0.977	N (rel.)
14	parapelvic_cyst	peripelvic_cyst	0.976	Y
15	fourth_ventricle	4th_ventricle	0.975	Y
16	transplanted_kidney	transplant_kidney	0.975	Y
17	iodine-131	iodine_131	0.974	Y
18	hill-sachs_deformity	hill-sachs_lesion	0.973	Y
19	mucus	mucous	0.973	Y
20	nephrostomy	nephrostomy_tube	0.972	Y
21	hypodense	hypoattenuating	0.972	Y
22	external_fixator	external_fixation	0.972	Y
23	hindfoot	hind_foot	0.971	Y
24	ground-glass_opacity	ground-glass_opacification	0.971	Y
25	spinal_cord	cord	0.971	N (gen.)
26	main_bronchus	mainstem_bronchus	0.970	Y
27	non-union	nonunion	0.970	Y
28	gradient_coil	by_adjusting	0.969	N (error)
29	gastroduodenal_artery	gda	0.969	Y
30	fifth_digit	5th_digit	0.969	Y
31	pre-contrast	precontrast	0.969	Y
32	aortopulmonary_window	aorticopulmonary_window	0.968	Y
33	ureteral_calculus	ureteral_stone	0.968	Y
34	t2_hyperintense	t2_bright	0.968	Y
35	lateral_ventricle	frontal_horn	0.968	N (rel.)
36	mucous_membrane	mucus_membrane	0.967	Y
37	corona_radiata	centrum_semiovale	0.967	N (rel.)
38	corpus_callosum	splenium	0.967	N (rel.)
39	heterogeneous_enhancement	heterogenous_enhancement	0.967	Y
40	fourth_digit	5th_digit	0.967	

(as is *middle_finger*) but not synonyms.

Rows 13, 35, 37, and 38 illustrate connections between closely related terms (in this case, brain structures) that are not synonyms. However, all of the newly predicted terms are correct brain structures that should be present in RadLex. The frontal horn refers to an anterior portion of the lateral ventricle, the centrum semiovale is a mass of white matter continuous with the corona radiata, and the splenium is a portion of the corpus callosum. All three terms are actually present in RadLex, but not in these forms; for example, "splenium of corpus callosum" is a term in Radlex, but "splenium" by itself is not.

In the case of *spinal_cord/cord* (row 25) one could make the case that in a radiology report, the term *cord* is most probably referring to the spinal cord, but it is certainly not a synonym. Finally, there are two relationships that are simply errors: *int rot* is a radiology view, while *pel max* refers to the maximum permissible exposure limit of radiation. The terms *gradient coil* and *by adjusting* have nothing semantically to do with each other.

4.4.4 Summary: Using Word Vectors for Lexicon Building from Clinical Documents

Ultimately, using distributional normalization allowed us to filter a set of around 770,000 terms down to a set of 19,770 potential candidate terms for RadLex, but these terms would still need to be reviewed by a human before they are entered into RadLex. At 10 seconds per term, reviewing 770,000 terms would take a human approximately 2140 man-hours, or around 267 days (working 8 hours a day with no breaks) while reviewing 19,770 terms would only take about a week. In addition, distributional methods provide "nearest neighbors" within a lexicon, so the question becomes not "Should this word exist somewhere in this lexicon and if so, where?" but "Is this word synonymous with this other word, yes or no?". The second question is much easier to answer.

Our conclusion is thus that distributional normalization can provide a valuable and efficient technique for expanding the coverage of existing lexicons and ontologies,

but it is not accurate enough to perform completely automated normalization. Its performance also depends on the nature of the entities involved. Someone wishing to apply it to his or her specific domain should consider starting with a wide context window (probably $w = 5$), the *min(rank,revrank)* score, and a score (rank) threshold of 4-6 (unless he/she has a sample set of synonyms available on which to optimize $F_{0.5}$).

Chapter 5

Ensemble Biclustering for Classification

In this chapter, we introduce a novel algorithm, Ensemble Biclustering for Classification (EBC), that is related to several of the techniques from Chapter 3 but uses a different notion of context than they typically employ. It was designed to serve as a more efficient alternative to LSA for relation extraction. EBC solves some of the technical challenges associated with relation extraction on pair-pattern matrices. In this chapter, we discuss the technical aspects of EBC. In Chapter 6, we show how EBC can be used to discover classes of drug-gene relationships from the unstructured text of the biomedical literature.

Much of the text of this chapter was borrowed from our published paper [106] and its supplementary material.

5.1 Background

The techniques discussed in Chapter 3 all build vector space models of targets (words, phrases and relations) based on cooccurrence counts with their contexts. Random indexing and word2vec, the two models we explored in Chapter 4, build vector representations of words and phrases based on the counts of surrounding context words (or some weighted version of these counts), though both could, in principle, be

extended to arbitrary contexts.

EBC is closely related to these techniques, but it has a few important differences. In particular, EBC:

- *Uses a binary matrix - either a feature can occur in a given context or it cannot.* Instead of counting cooccurrences of target and context, EBC operates on a matrix of 1s and 0s - did this context occur for this target *ever* in the corpus: yes or no?[1]
- *Is efficient for large, sparse datasets.* It takes $\mathcal{O}(\tau \cdot z \cdot N(k + \ell))$ operations to run EBC, where τ is the number of iterations each run of ITCC takes to converge (see Section 5.2), z is the number of nonzero matrix elements, N is the number of independent runs of ITCC, and k and ℓ are the row and column cluster numbers. The N independent runs of ITCC can easily be parallelized. In contrast, algorithms for computing the SVD depend on the row and column dimensions of the matrix itself, which may be huge compared to the number of nonzero elements.
- *Handles missing data naturally.* Empty cells in the matrix are simply ignored. In this sense, EBC is closer in spirit to random indexing than it is to word2vec, which requires sampling of contexts that did *not* occur[2].
- *Enables flexible feature spaces.* The context for EBC can be anything, from dependencies to neighbor words to documents. Although this is also theoretically true for all of the methods from Chapter 3, since all can be represented as matrices, in practice it is more difficult for some than others. EBC is entirely agnostic to the type of data in its input matrix.
- *Has no free parameters.* We have developed a heuristic for finding row and column cluster numbers for EBC that leads to high performance on drug-gene relation extraction and the other tasks for which we have tried EBC. Although one could optimize k and ℓ for a given task, for example by using a validation set,

[1] In principle, there is no reason why EBC could not be applied to other matrix types, though another type of biclustering algorithm may be more appropriate for dense, real-valued matrices. ITCC really shines when the matrices involved are sparse.

[2] This sampling results in an additional parameter, the negative sampling parameter, which must be set by the user.

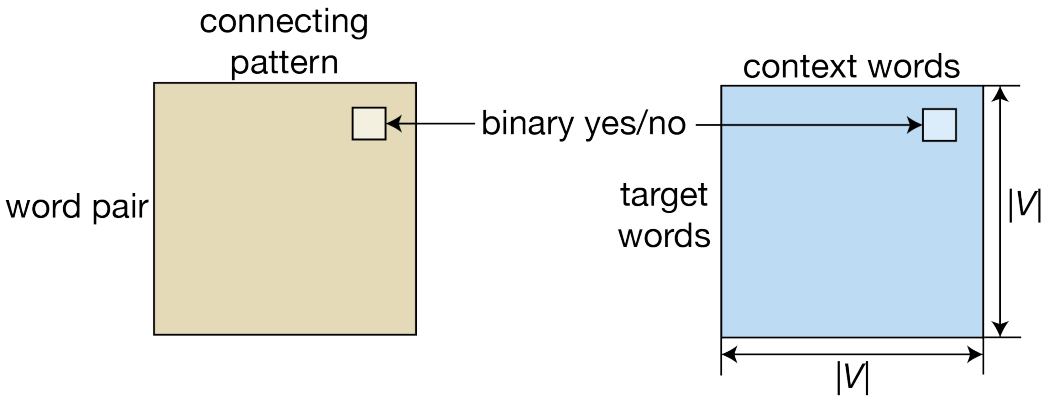


Figure 5.1. What the matrices for EBC look like if EBC is applied to (left) relation extraction and (right) word similarity.

we valued being able to choose k and ℓ in circumstances where we have access to little or no training data. The development of our heuristic was inspired by a problem that had plagued LSA: how to choose the number of singular values used by the truncated SVD produce reduced-dimensional vectors for words. LSA's performance is sensitive to that choice, as we show below in Section 6.7[3], whereas for EBC, it's possible to let the data tell us what choices to make.

- *Unifies the target representation and similarity assessment.* Random indexing, word2vec, LSA and all of the methods from Chapter 3 produce vectors that are typically compared using cosine similarity or some other type of vector similarity measure. In EBC, the similarity assessment of different target rows is baked into the representation - we bicluster the matrix multiple times, and the similarity score is how frequently any two rows cluster together. This could be considered either good or bad, but it is definitely different from how other distributional semantics methods calculate similarity.

EBC is an ensemble method in which we form a low-rank approximation of a data

[3]The authors of the original LSA paper noted that the optimal choice of dimension for their vectors was an important parameter that, depending on the dataset, strongly affected their results. Because they were psychologists and viewed LSA as a model of learning, they suggested that an organism might learn by optimizing this parameter for different tasks [66]: "Because, as we see later, the model predicts what words should occur in the same contexts, an organism using such a mechanism could, either by evolution or learning, adaptively adjust the number of dimensions on the basis of trial and error."

matrix using biclustering and stack thousands of slightly different approximations on top of each other to make similarity assessments of the matrix's rows (or columns). It is most similar in spirit to Latent Semantic Analysis (LSA) [26] (Section 3.2), which uses the singular value decomposition (SVD) [137] instead of biclustering to accomplish a similar goal, and has been applied in at least one case to corpus-level relationship extraction (a technique called Latent Relational Analysis, or LRA) [145]. The big difference, aside from the points above, is that EBC uses multiple low-rank approximations of the matrix, whereas the SVD finds a single, optimal low-rank approximation.

5.2 Information-Theoretic Co-Clustering (ITCC)

The backbone of EBC is a biclustering algorithm called Information-Theoretic Co-Clustering (ITCC) [27]. ITCC treats a matrix, M, as a joint probability distribution over its rows (Y) and columns (X). Given fixed numbers of row and column clusters, ITCC finds a set of cluster assignments for the rows and columns for which the mutual information between the clustered random variables, $\hat{X}$ and $\hat{Y}$, is as high as possible relative to the mutual information between X and Y. In other words, the algorithm finds maps C_X and C_Y, where

$$C_X : \{x_1, x_2, \ldots, x_m\} \rightarrow \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\}$$
$$C_Y : \{y_1, y_2, \ldots, y_m\} \rightarrow \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m\}$$

that ensure the difference

$$I(X, Y) - I(\hat{X}, \hat{Y}) \tag{5.1}$$

is minimized. Equation 5.1 thus becomes the objective function of ITCC.

We will go through some of the math from [27] here, both to provide some background for our discussion of EBC, and because it is interesting.

5.2.1 Deriving the Algorithm

The objective as KL divergence Lemma 2.1 in [27] states that we can write the loss in mutual information from Equation 5.1 as

$$\begin{aligned} I(X,Y) - I(\hat{X},\hat{Y}) &= D(p(X,Y) \parallel q(X,Y)) \\ &= \sum_{\hat{x}} \sum_{\hat{y}} \sum_{x \in \hat{x}} \sum_{y \in \hat{y}} p(x,y) \log \left(\frac{p(x,y)}{q(x,y)} \right) \end{aligned}$$

where $D(\cdot \parallel \cdot)$ denotes the Kullback-Liebler (KL) divergence, or relative entropy, and $q(X,Y)$ is a distribution of the form

$$q(x,y) = p(\hat{x},\hat{y})p(x|\hat{x})p(y|\hat{y}) \tag{5.2}$$

where $x \in \hat{x}$, and $y \in \hat{y}$. Importantly, as is shown in [27], $p(\hat{x},\hat{y}) = q(\hat{x},\hat{y})$, $p(x|\hat{x}) = q(x|\hat{x})$, and $p(y|\hat{y}) = q(y|\hat{y})$, so we could just as easily write q instead of p in Equation 5.2. Also note that the KL divergence only takes this form because we are considering hard clusterings of the rows and columns, so

$$p(\hat{x},\hat{y}) = \sum_{x \in \hat{x}} \sum_{y \in \hat{y}} p(x,y) \tag{5.3}$$

$$p(\hat{x}) = \sum_{x \in \hat{x}} p(x) \qquad p(\hat{y}) = \sum_{y \in \hat{y}} p(y)$$

and the conditional distributions $p(x|\hat{x})$ and $p(y|\hat{y})$ have the form $p(x|\hat{x}) = p(x)/p(\hat{x})$ for $\hat{x} = C_X(x)$ and 0 otherwise (and same for y). Having expressed the objective in this form, we need a way to figure out how to assign the maps C_X and C_Y to create clusters so that this KL divergence is minimized.

Additional identities If $\hat{y} = C_Y(y)$ and $\hat{x} = C_X(x)$, then two more identities are also true:

$$q(y|\hat{x}) = q(y|\hat{y})q(\hat{y}|\hat{x}) \qquad q(x|\hat{y}) = q(x|\hat{x})q(\hat{x}|\hat{y}) \tag{5.4}$$

$$q(x,y,\hat{x},\hat{y}) = p(x)q(y|\hat{x}) \qquad q(x,y,\hat{x},\hat{y}) = p(y)q(x|\hat{y}) \tag{5.5}$$

The proofs of both Equations 5.4 and 5.5 are straightforward and can be found in [27].

Minimizing the objective We already showed that the ITCC objective can be expressed as the KL divergence between two distributions, $p(x, y)$ and $q(x, y)$. Lemma 4.1 from [27] shows that this objective can be expressed in two different ways, which helps the authors formulate an algorithm for minimizing it. First,

$$\begin{aligned} D(p(X,Y,\hat{X},\hat{Y}) \parallel q(X,Y,\hat{X},\hat{Y})) &= \sum_{\hat{x},\hat{y}} \sum_{\substack{x:C_X(x)=\hat{x} \\ y:C_Y(y)=\hat{y}}} p(x,y,\hat{x},\hat{y}) \log \frac{p(x,y,\hat{x},\hat{y})}{q(x,y,\hat{x},\hat{y})} \\ &= \sum_{\hat{x},\hat{y}} \sum_{\substack{x:C_X(x)=\hat{x} \\ y:C_Y(y)=\hat{y}}} p(x)p(y|x) \log \frac{p(x)p(y|x)}{p(x)q(y|\hat{x})} \\ &= \sum_{\hat{x}} \sum_{x:C_X(x)=\hat{x}} p(x) \sum_{y} p(y|x) \log \frac{p(y|x)}{q(y|\hat{x})} \end{aligned}$$

where the first equality follows from Equation 5.5 and also uses the fact that as long as $\hat{y} = C_Y(y)$ and $\hat{x} = C_X(x)$, $p(x, y, \hat{x}, \hat{y}) = p(x, y) = p(x)p(y|x)$. This expresses the objective solely in terms of the $\hat{X}$ clusters. We can perform a similar derivation to express the objective solely in terms of the $\hat{Y}$ clusters, which results in

$$D(p(X,Y,\hat{X},\hat{Y}) \parallel q(X,Y,\hat{X},\hat{Y})) = \sum_{\hat{y}} \sum_{y:C_Y(y)=\hat{y}} p(y) \sum_{x} p(x|y) \log \frac{p(x|y)}{q(x|\hat{y})}$$

Aside from allowing us to express the objective in terms of either the row or column clustering, this derivation also allows us to define the distribution $q(Y|\hat{x})$ as a "row cluster prototype" and the distribution $q(X|\hat{y})$ as a "column cluster prototype". Intuitively, we can see that assigning each row and column to its closest "prototype" cluster (where "closest" is defined in terms of KL divergence) will minimize the objective.

The ITCC algorithm follows directly from the derivation above. In Figure 5.2, we reproduce it from Figure 1 of [27]. The only difference is that we treat y as the index for rows and x as the index for columns (because in looking at a matrix, the columns are along the x-axis and the rows along the y-axis), and Dhillon *et al* do the opposite.

Input The joint probability distribution $p(X, Y)$, the number of row clusters, k, and the number of column clusters, ℓ.

Output The partition functions $C_X^\dagger$ and $C_Y^\dagger$.

1. Initialization: Set $t = 0$. Start with some initial partition functions $C_X^{(0)}$ and $C_Y^{(0)}$. Compute
$$q^{(0)}(\hat{X}, \hat{Y}), q^{(0)}(X|\hat{X}), q^{(0)}(Y|\hat{Y})$$
and the distributions $q^{(0)}(Y|\hat{x})$, $1 \leq \hat{x} \leq \ell$ using Equation 5.4.
2. Compute column clusters: For each column x, find its new cluster index as
$$C_X^{(t+1)}(x) = \text{argmin}_{\hat{x}} D\left(p(Y|x) \| q^{(t)}(Y|\hat{x})\right),$$
resolving ties arbitrarily. Let $C_Y^{(t+1)} = C_Y^{(t)}$.
3. Compute distributions
$$q^{(t+1)}(\hat{X}, \hat{Y}), q^{(t+1)}(X|\hat{X}), q^{(t+1)}(Y|\hat{Y})$$
and the distributions $q^{(0)}(X|\hat{y})$, $1 \leq \hat{y} \leq k$ using Equation 5.4.
4. Compute row clusters: For each row y, find its new cluster index as
$$C_Y^{(t+2)}(y) = \text{argmin}_{\hat{y}} D\left(p(X|y) \| q^{(t+1)}(X|\hat{y})\right),$$
resolving ties arbitrarily. Let $C_X^{(t+2)} = C_X^{(t+1)}$.
5. Compute distributions
$$q^{(t+2)}(\hat{X}, \hat{Y}), q^{(t+2)}(X|\hat{X}), q^{(t+2)}(Y|\hat{Y})$$
and the distributions $q^{(t+2)}(Y|\hat{x})$, $1 \leq \hat{x} \leq \ell$ using Equation 5.4.
6. Stop and return $C_X^\dagger = C_X^{(t+2)}$ and $C_Y^\dagger = C_Y^{(t+2)}$ if the change in objective function value, that is,
$$D\left(p(X, Y) \| q^{(t)}(X, Y)\right) - D\left(p(X, Y) \| q^{(t+2)}(X, Y)\right)$$
is "small" (say 10^{-3}). Else set $t = t + 2$ and go to step 2.

Figure 5.2. Information theoretic co-clustering algorithm from Figure 1 of [27].

5.2.2 Implementation Considerations

We have implemented ITCC in both Java and Python[4]. Implementing the algorithm has alerted us to several technical considerations that others might wish to consider if they decide to use it. We have listed these below.

1. *Initialization strategy.* To initially assign rows and columns to clusters, we choose $k-1$ row and $\ell-1$ column cluster centers from among the rows and columns of the matrix and match the rest of the rows and columns to these clusters based on Jaccard[5] similarity with the cluster centers.
2. *Dealing with rows and columns that cannot be initialized.* If a row or column has zero Jaccard similarity with all of the centers (zero overlap with all) it is initialized to a separate "other" cluster, so all of the rows that cannot initially be assigned start out in the same cluster, and the same is true for columns. Thus the total number of row and column clusters is always k and ℓ, as desired.
3. *Sensitivity to initialization strategy.* Results, in terms of both the final objective and the appearance of the final clusters, are sensitive to the initialization strategy. For example, assigning rows and columns uniformly to clusters (for example, by choosing a random $\hat{x} \in [0, \ell)$ for every column x and a random $\hat{y} \in [0, k)$ for every row y) is a terrible idea because it leads to clusters whose conditional distributions over rows and columns very closely resemble each other.
4. *Ensuring k and ℓ stay constant.* In Dhillon and Modha's original algorithm, it is possible to "lose" clusters over the course of one run of ITCC. This can happen, for example, if two rows with exactly the same distributions of nonzero elements over the columns are chosen as cluster centers. One cluster center row can then be assigned to the other cluster, reducing the total number of row clusters by 1. We therefore introduced a check in our code: after each iteration of ITCC, the code checks to see if the desired numbers of rows and columns are present; if not, it randomly selects the number of missing rows/columns uniformly from the matrix rows/columns and relabels them as the cluster centers of the missing

[4] https://github.com/blpercha/ebc

[5] In the original EBC paper [106], we used cosine similarity, but later concluded that for binary matrices, Jaccard similarity makes more sense. The results are similar in both cases.

clusters. It then checks again to ensure that the total numbers of row and column clusters are correct, and continues the random assignment process until they are. This was especially important when implementing the heuristic that determines optimal cluster numbers (Section 5.3).

5. *Using only binary matrices.* If one only wants to work with binary matrices, it is possible to make ITCC faster and less memory intensive by simply storing the locations of the nonzero elements in the matrix and storing the uniform value of these elements as a separate variable.
6. *Rows first or columns first?* Although the ITCC algorithm in Figure 5.2 is shown as clustering the columns first, it does not matter whether one chooses to cluster the rows or columns first. The final objective, in general, appears to be similar in both cases. One thing this does affect, however, is the optimal k and ℓ found using our heuristic (Section 5.3), so it is important to use a consistent axis ordering (rows-columns or columns-rows) when optimizing cluster numbers and when running the algorithm.
7. *Don't store the cross-conditional distributions.* Despite what it says in Figure 5.2, it is not necessary to store the $q(Y|\hat{x})$ and $q(X|\hat{y})$ distributions, as these values can be calculated "on the fly" from the joint distribution over clusters, $q(\hat{X}, \hat{Y})$ and the conditional distributions $q(Y|\hat{y})$ and $q(X|\hat{x})$. The $q(Y|\hat{x})$ and $q(X|\hat{y})$ distributions (we call these the *cross-conditional* distributions) can be large because they are $m \times \ell$ and $n \times k$ in size, so it's best to avoid storing them.
8. *Iterate over nonzero elements only.* Figure 5.2, for expositional clarity, often makes it seem as though one should iterate over all the rows and columns of the matrix. However, the optimal runtime of the algorithm depends on iterating over only the nonzero elements. There are multiple ways to store the matrix internally to accomplish this, so we leave that choice to the reader. We use two different options in our Java and Python packages for EBC.

5.3 Finding Optimal Cluster Numbers k and ℓ

5.3.1 A Heuristic for Finding Cluster Numbers

There are two unknown parameters that ITCC requires as input: the numbers of row (k) and column (ℓ) clusters. The objective function of the ITCC algorithm is the difference in the mutual information between X and Y in the original dataset, $I(X, Y)$, and between the clusters $\hat{X}$ and $\hat{Y}$ in the clustered dataset, $I(\hat{X}, \hat{Y})$ as shown in Equation 5.1. This objective generally decreases as k and ℓ increase, since by introducing more clusters, the approximate distribution $q(x, y)$ more closely approximates the original distribution $p(x, y)$. Therefore, we need a separate heuristic to decide on the optimal k and ℓ[6].

The key fact that led us to our solution is that the objective function decreases with increasing k and ℓ no matter what the original matrix (M) looks like. In particular, this should be true if we choose any random matrix (M_r) whose $n \times m$ individual elements are the same as those of M, but where the locations of those elements within the matrix are randomized with respect to both rows and columns. We can think of the "optimal clustering" as one that captures the structure inherent in M using as few clusters as possible. We therefore reasoned that we should search for k and ℓ such that the value of the objective function for the clustering over M was as low as possible relative to its value for M_r, meaning that the clustering captured more of the original mutual information $I(X, Y)$ than would be expected due to chance. In other words, we sought to minimize

$$\text{objective}(M) - \text{objective}(M_r) \tag{5.6}$$

over many different randomized M_r matrices[7]. In spirit, our heuristic most resembles

[6] Although ITCC is a general-purpose biclustering algorithm and has not been used much in NLP, the authors of the original ITCC paper actually evaluated it on a word-document cooccurrence matrix. They stated, "Different data sets achieve their maximum at different numbers of word clusters. In general selecting the number of clusters to start with is a non-trivial model selection task and is beyond the scope of this paper." They discussed using an information-theoretic regularization procedure like minimum description length to select the optimal numbers of clusters, but to my knowledge did not pursue this.

[7] *Randomization scheme.* We make lists of the row and column indices for all nonzero elements

the popular gap statistic for unidimensional clustering [144].

To identify optimal k and ℓ, therefore, we perform grid searches over ranges of k and ℓ, typically in the range $[5, 400]$ ($50 - 100$ separate clusterings at each (k, ℓ); grid size = 5 or 10) and identify the k and ℓ that minimize the empirical mean of Equation 5.6. If necessary, once the neighborhood of the minimum is identified, we perform an additional, finer-grained grid search over that area. Note that although this step requires a large number of ITCC runs, the output from all of these runs is independent and they can be parallelized.

5.3.2 Examples of Heuristic on Small Matrices

We present here three examples of small, binary matrices, and show how our heuristic can be used to find k and ℓ for each matrix. Beside the matrix are contour plots of the score from Equation 5.6. In these plots, dark denotes a low value of Equation 5.6 (good) and light denotes a high value (bad). The left plot is what happens in the case of row priority clustering (rows clustered first) and the right plot is for column priority clustering. The two plots are virtually identical except in the case of the first matrix, which is asymmetric[8].

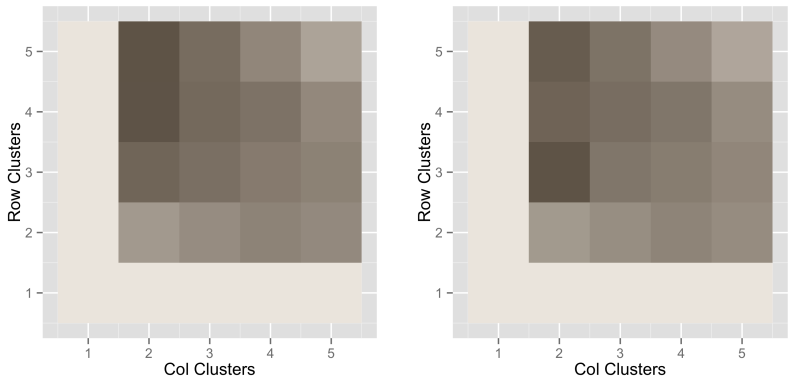
Example 1 This is a "binarified" version of the original example matrix used in Dhillon *et al*'s 2003 paper [27]. In the paper, the authors choose $k = 3$ as the number of row clusters and $\ell = 2$ as the number of column clusters, but our heuristic finds

in the matrix and shuffle the column indices relative to the rows to create new pairs. If necessary, we randomly reshuffle some column indices until there are no duplicate row-column pairs in the lists (this step is necessary as duplicates do sometimes occur when the lists are randomized). This procedure ensures that there is a nonzero element in every row and column and also has the nice property that it preserves the row and column marginal distributions.

[8]These plots are similar to what was shown in the supplementary material for [106]. In that paper, the randomization scheme we used was slightly different than our current approach. It did not preserve marginals and in addition, two nonzero matrix elements could be randomized to the same location (the values of those cells were simply added together). As a result, the minimum for the first matrix shown here is different than it was in the paper. We believe our current randomization scheme leads to cluster numbers that are more accurate. For instance, it finds $\ell = 2$ column clusters for the first matrix here whereas the method we used in the paper led to $\ell = 3$. In general the optimal cluster numbers found using both methods are very close.

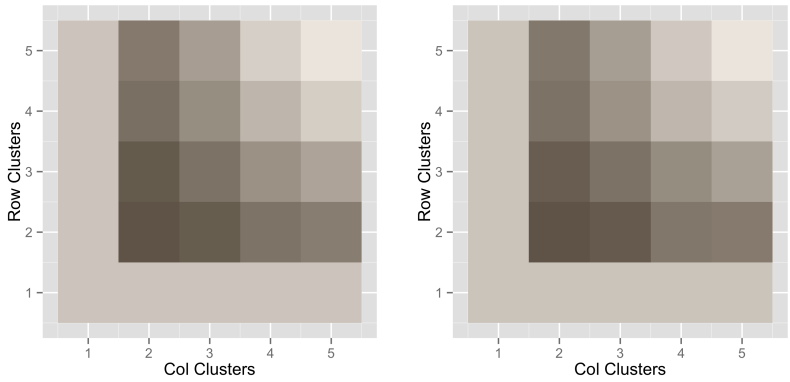
$k = 4$, $\ell = 2$ (when rows are clustered first) and $k = 3$, $\ell = 2$ (when columns are clustered first).

1	**1**	**1**	0	0	0
1	**1**	**1**	0	0	0
0	0	0	**1**	**1**	**1**
0	0	0	**1**	**1**	**1**
1	**1**	0	**1**	**1**	**1**
1	**1**	**1**	0	**1**	**1**



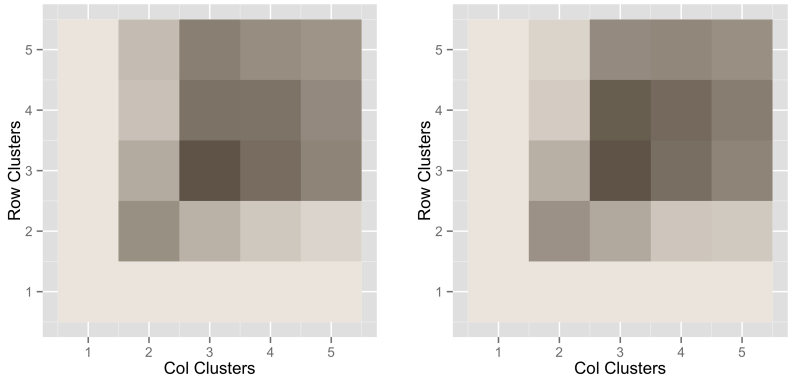
Example 2 Here is an example where the matrix obviously contains two row and two column clusters:

1	**1**	**1**	0	0	0
1	**1**	**1**	0	0	0
1	**1**	**1**	0	0	0
0	0	0	**1**	**1**	**1**
0	0	0	**1**	**1**	**1**
0	0	0	**1**	**1**	**1**



Example 3 Finally, here is an example where the matrix obviously contains three row and three column clusters:

1	**1**	0	0	0	0
1	**1**	0	0	0	0
0	0	**1**	**1**	0	0
0	0	**1**	**1**	0	0
0	0	0	0	**1**	**1**
0	0	0	0	**1**	**1**



5.4 The EBC Algorithm

Due to random initialization of the row and column cluster centers, ITCC will converge to a different locally optimal biclustering on each run. The EBC algorithm is an ensemble version of ITCC in which the heuristic from Section 5.3 is employed to select k and ℓ and then information from thousands of independent ITCC runs is combined. Here we assume the input data matrix is binary and of dimension $m \times n$. How we use EBC is task-dependent.

5.4.1 Semisupervised EBC

One potential use of EBC is to find new targets (for relation extraction, these are entity pairs, such as drug-gene pairs) that share a relationship with some targets we already have. For example, we may have access to a few drug-gene pairs that share some relationship and want to find more like them. We call this limited input data the *seed set*, and we call this version of EBC *semisupervised EBC* because it incorporates both unsupervised clustering and supervised classification steps. In Chapter 6, we show how semisupervised EBC can be used to extract drug-gene relationships of two types from the biomedical literature. The two steps are:

1. *Unsupervised step.* Use ITCC to bicluster the matrix N times, recording the number of runs in which each row appears in a row cluster with each other row. The result is an $m \times m$ array, C, of co-occurrence values[9]. Note that no information about the seed set is incorporated at this stage, so the unsupervised step need be run only once per data matrix.
2. *Supervised step.* Identify a seed set, S, of rows that share some property of interest. (In our experiments in Chapter 6, these were drug-gene pairs with known pharmacogenomic or drug-target relationships.) Also identify a test set, T, which may be all of the other rows in the matrix or a subset of them. Rank

[9]When m and n get very large, recording the $m \times m$ matrix of cooccurrence values becomes inefficient. In that case, we simply record the final cluster IDs for each row on every run. This means recording only N numbers (usually $N \approx 2000$) for each of m rows, instead of m numbers (where m could be as high as a few million). The score calculations and similarity assessments can then be done on the fly using these stored, length N vectors of cluster assignments.

them based on a scoring function related to how often they co-cluster with members of S (details below). Repeat this step as desired with different seed sets.

Similarity Assessments Once EBC's unsupervised step is performed and appropriate seed (S) and test (T) sets identified, test set items can be ranked as follows:

1. *EBC raw similarity scores.* The simplest way to rank the test set items is by their total cooccurrence frequency with members of the seed set. Let $C(T_i, S_j)$ be the cooccurrence frequency of the ith member of the test set and the jth member of the seed set. Then the score for test set member i would be:

$$\text{score}(T_i) = \sum_{j=1}^{|S|} C(T_i, S_j).$$

2. *EBC rank sum scores.* For each test set member, T_i, rank all m rows of the data matrix based on how frequently they cocluster with T_i. This produces a ranking R_i of length m in which pairs that frequently cocluster with T_i are assigned high ranks and those that seldom cocluster get low ranks. Ties are broken randomly. The score for T_i is then the rank sum of the members of the seed set, S, within this list, or:

$$\text{score}(T_i) = \sum_{j=1}^{m} j \cdot I\{R_{ij} \in S\}$$

where

$$I\{R_{ij} \in S\} = \begin{cases} 1 & \text{if } R_{ij} \in S \\ 0 & \text{otherwise} \end{cases}$$

Using ranks instead of absolute co-clustering frequencies produces a score that does not depend on how often, on average, a given row co-clusters with other rows. For some applications, those differences might not matter (or they might be informative) but we normalized to ranks so promiscuous pairs (which are often well-known or frequently mentioned entity pairs) would not consistently receive higher scores than less promiscuous pairs. The rank sum scoring function will assign a high score to a test set member as long as the seed set rows tend to cluster with it more frequently

than other rows do.

To see how much EBC improves our ability to prioritize relevant information from the literature over simple context matching (comparing matrix rows directly), we typically compare it to two other similarity assessment methods that compare the rows of the matrix directly:

1. *Average cosine distance or average Jaccard distance.* Let v_{T_i} be the row vector in the data matrix associated with test set member i. This vector is of length n. Let v_{S_j} be the row vector associated with seed set member j. Here we score each test pair T_i based on the average cosine similarity of v_{T_i} with all of the row vectors from the seed set, or:

$$\text{score}(T_i) = \frac{1}{|S|} \sum_{j=1}^{|S|} \frac{v_{T_i} \cdot v_{S_j}}{\|v_{T_i}\| \|v_{S_j}\|}$$

   where $\|\cdot\|$ denotes the Euclidean norm. Alternatively, since we mainly deal with binary matrices, we can use the Jaccard similarity score, where $Z(T_i)$ is the set of nonzero column indices for the row in the data matrix associated with test set member i, and $Z(S_j)$ is the set of nonzero column indices for the row in the data matrix associated with seed set member j

$$\text{score}(T_i) = \frac{|Z(T_i) \cap Z(S_j)|}{|Z(T_i) \cup Z(S_j)|}.$$

2. *Rank sum scores.* In keeping with the spirit of EBC's rank sum scoring function, for each T_i we rank all m rows of the data matrix based on cosine (or Jaccard) similarity to v_{T_i}. This produces a ranking R_i of length m in which rows with high similarity to v_{T_i} are assigned high ranks and those with low similarity get low ranks. The score for T_i is the rank sum of the members of S within this list, and looks identical to that for EBC's rank sum scoring function; the only difference is that the rankings R_i are produced using cosine or Jaccard similarity and not EBC.

5.4.2 Unsupervised EBC

It is also possible to use the unsupervised step of EBC separately to discover patterns of similarity among matrix rows; for example, through hierarchical clustering. One could use the raw co-clustering frequencies as input to a hierarchical clustering algorithm, but this also tends to prioritize promiscuous matrix rows and make them appear more similar to each other than they are to less promiscuous rows. In unsupervised EBC, we recommend using a method that has some of the advantages of the rank sum scoring functions described for semisupervised EBC.

To implement EBC's scoring function in an unsupervised manner to construct a dendrogram, we start with our $m \times m$ matrix of co-occurrence values, C, in which C_{ij} is the number of runs (out of N total) in which row i co-clusters with row j. We then convert C into a correlation matrix, ρ, also $m \times m$, where ρ_{ij} contains the Spearman correlation of $C_{i\cdot}$ and $C_{j\cdot}$, the ith and jth rows of C (note that C is symmetric, so we could just as easily have used columns). These correlations are, as in EBC's rank scoring function, measures of how similarly row i and row j rank all other rows in the matrix, and are not biased in favor of promiscuous pairs. We then use $1 - \rho$ as the distance measure for hierarchical clustering.

In Chapter 6, we use this approach with the minimax linkage function [5] to produce a dendrogram showing the relationships among 3514 drug-gene pairs. Using a different linkage function or distance metric, obviously, would produce a different-looking dendrogram. In other words, there is no one correct way to use unsupervised EBC.

Chapter 6

The Structure of Drug-Gene Relationships

The creation of comprehensive, structured resources that catalog the relationships between drugs and genes would accelerate the translation of basic molecular knowledge into discoveries of genomic biomarkers for drug response and prediction of unexpected drug-drug interactions [105]. The published biomedical research literature encompasses most of our understanding of how drugs interact with gene products to produce physiological responses (phenotypes). Unfortunately, this information is distributed throughout the unstructured text of over 24 million articles.

Although a great deal of research effort has been directed at the problem of relationship extraction from biomedical text in pharmacogenomics [11, 20, 154], and in the biomedical domain in general [36, 74, 87, 111, 113, 131], high-quality biomedical knowledge bases like OMIM [40], DrugBank [152] and PharmGKB [150] still rely almost entirely on human curators, who comb the literature manually in search of new relationships. The authors of BioGraph, a new biomedical knowledge base incorporating data from 21 different sources, recently decided to exclude databases that were not manually curated, citing data quality issues [75].

Here we focus on the problem of drug-gene relationship extraction and characterization from unstructured biomedical text, using statistical dependency parsing to extract descriptions of drug-gene relationships from Medline sentences and applying

EBC (Chapter 5) to recognize when two drug-gene pairs share a similar relationship, even when they are described differently in the text. We show that EBC significantly improves our ability to extract both pharmacogenomic (PGx) and drug-target relationships, and use it to discover new drug-gene relationships for PharmGKB and DrugBank.

Finally, we combine EBC and hierarchical clustering to map the global landscape of drug-gene interactions, revealing much unforeseen complexity in how these relationships are described in text.

As in Chapter 5, much of the text of this chapter was borrowed from our published paper [106] and its supplementary material.

6.1 Extracting Biomedical Relations by Analogy

Relationships can be extracted from the biomedical literature in two ways:

1. *Matching to known relationships.* If we have a small set of known information, such as a few drug-gene pairs that share a relationship, we can look for other drug-gene pairs that share that relationship.
2. *Unsupervised clustering.* If we have a way to assess how similar the relationship between one drug-gene pair is to the relationship between any other drug-gene pair, we can employ unsupervised clustering to discover relationship classes without specifying what they are.

If we look closely, neither of these approaches requires that we explicitly map any particular *pattern* (way of describing a relationship in text) to any well-defined relationship class. In both cases, our concern is with the drug-gene pairs. We simply want to know how likely it is that two drug-gene pairs share a relationship, either so we can match new pairs to our template pairs (1) or so the similarity scores we input to our unsupervised clustering algorithm will be accurate (2).

Unfortunately, we cannot directly assess how similar two drug-gene pairs' relationships are if they are described differently in the text. However, it turns out that EBC (Chapter 5) provides a key to this dilemma.

The backbone of EBC is a biclustering algorithm called Information-Theoretic

Co-Clustering (ITCC; [27], Section 5.2). Figure 6.1 shows the result of one ITCC run on a small sample dataset consisting of patterns[1] that connect different drugs to the gene CYP3A4[2] at least five times in Medline 2013. This dataset contains 62 drug-gene pairs (where the gene is always CYP3A4) and 14 unique patterns. These are arranged in a pair-pattern matrix, M, where an element M_{ij} is "1" if drug-gene pair i is connected by pattern j somewhere in Medline, and "0" otherwise. We used ITCC to bicluster this matrix into four row clusters and six column clusters. Besides biclustering the matrix, the $q(x, y)$ approximate distribution from ITCC (Section 5.2) creates a smoothed version of the matrix where certain elements that were not observed in the original dataset are filled in.

Figure 6.1 illustrates that the rows fragment into four clusters that reflect distinct ways that drugs can interact with CYP3A4. Row cluster 1 contains CYP3A4 inhibitors, a few of which are also substrates. Row cluster 2 contains CYP3A4 inducers. Row clusters 3 and 4 contain substrates of CYP3A4 that are not known inhibitors.

However, in looking at Figure 6.1, we notice that some drug-gene pairs, such as *ciprofloxacin/CYP3A4* and *quinidine/CYP3A4*, end up in the same row cluster even though they share no dependency paths in common. This happens because at the same time the rows are being clustered, the columns are also being clustered, so the rows' similarity is assessed with respect to the column *clusters*, not the columns themselves[3].

What about the column clusters? Figure 6.1 shows that they naturally fragment into clusters reflecting known biomedical properties. All of the paths referring to inhibition, for example, appear together in column cluster 2. The sole path referring to induction appears by itself in column cluster 6. The other four clusters include paths describing situations where the drug is a substrate of CYP3A4, or is metabolized by

[1] Dependency paths; we explain how these are extracted below.

[2] CYP3A4 is a liver cytochrome involved in the pharmacokinetic pathways of many drugs.

[3] Another quote from Dhillon and Modha's original paper is relevant here. Speaking of biclustering a word-document matrix, they state: "Word-document matrices that arise in information retrieval are known to be highly sparse. For such sparse, high-dimensional data, even if one is only interested in document clustering, our results show that co-clustering is more effective than a plain clustering of just documents. The reason is that when co-clustering is employed, we effectively use word clusters as underlying features and not individual words. *This amounts to implicit and adaptive dimensionality reduction and noise removal leading to better clusters*" [27] (p. 9, emphasis mine).

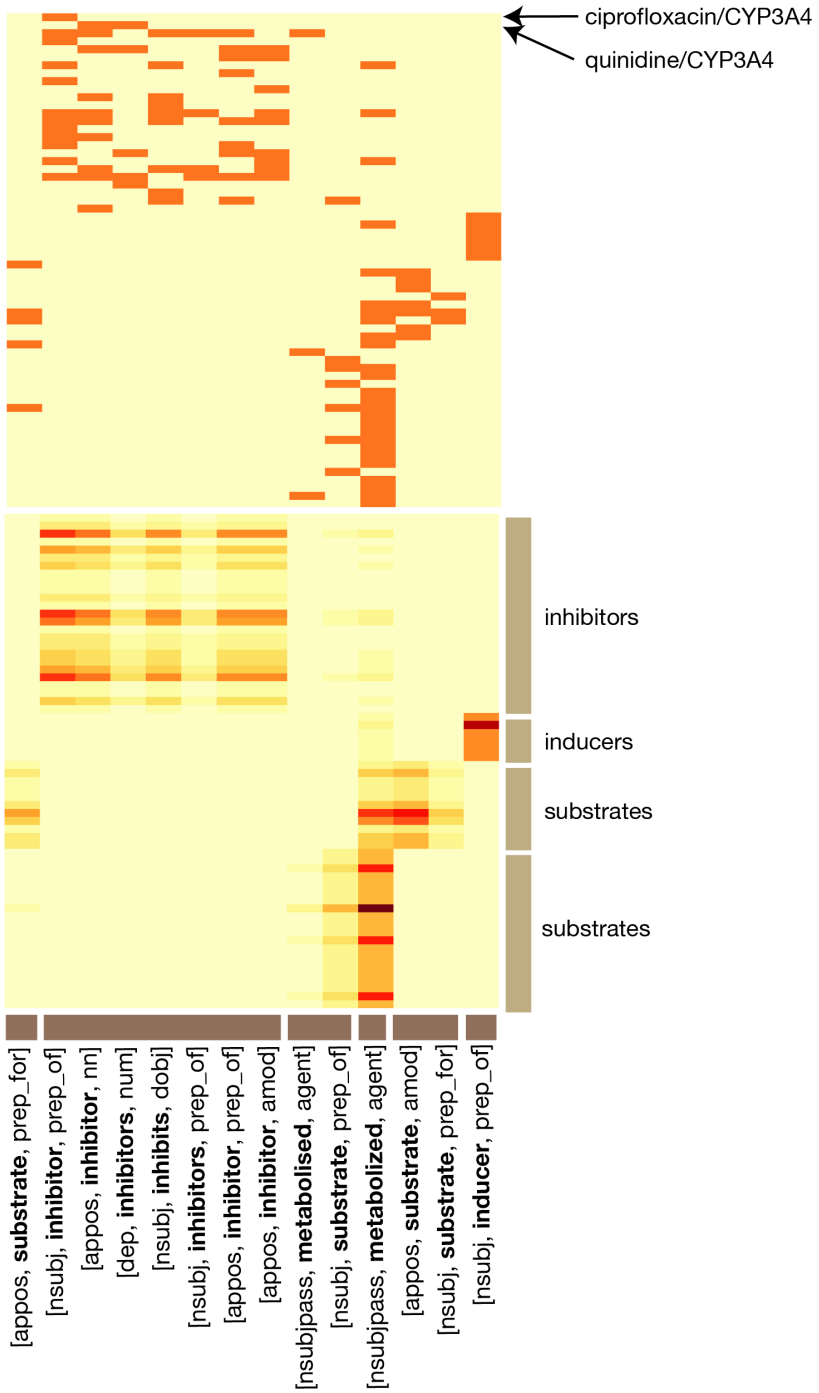


Figure 6.1. Example of ITCC output for a small matrix consisting of drug-CYP3A4 pairs and their associated dependency paths. The top heatmap shows the original data after the clustering was performed. An orange square represents an observed path (column) between a given drug-gene pair (row). The bottom heatmap shows the approximate distribution arising from a single ITCC run.

it. In other words, paths that cluster together appear to be semantically related.

EBC combines information from thousands of different biclusterings like this one to assess the relationship similarity of any two drug-gene pairs (rows) in the matrix, by looking at how frequently they cluster together. As we have seen in this example, EBC gives us a way to implicitly reason about drug-gene relationships "by analogy". If two drug-gene pairs are connected by similar sets of patterns, EBC infers that they are related. Similarly, if two dependency paths connect similar sets of drug-gene pairs, EBC infers that the meaning of those paths is similar. Over several iterations, EBC is able to quantitatively estimate the relatedness of two drug-gene pairs, even when they share no dependency paths in common. A similar idea forms the basis of the technique *distant supervision*, commonly applied to relationship extraction problems [90], as well as the algorithms LRA [146] and DIRT [77] (Section 3.2.2).

6.2 Building a Pair-Pattern Matrix

The matrix in Figure 6.1 provides a nice illustration of the method, but in reality we want to bicluster much larger matrices. We therefore created matrices of drug-gene pairs and their connecting patterns for all of Medline. To create these matrices, we did the following:

1. *Identify all drug-gene pairs co-occurring in sentences in Medline.* Call the number of drug-gene pairs m. See Section 6.2.1 for details.
2. *Extract all dependency paths connecting these drug-gene pairs in the corpus.* Call the total number of observed paths n. See Section 6.2.2 for details.
3. Arrange the data in an $m \times n$ matrix where the rows represent drug-gene pairs and the columns dependency paths. A cell with coordinates (i, j) in this matrix contains "1" if drug-gene pair i has been connected by path j somewhere in the corpus, and "0" otherwise.

6.2.1 Named Entity Recognition of Drugs and Genes

We identified drug and gene entity names in the text using simple string matching to lexicons, though any type of named entity recognition software could be incorporated at this stage [69, 70]. We obtained drug and gene lexicons from PharmGKB [150] and filtered them against a dictionary of common English words to remove promiscuous terms (such as "CAT", which is both a gene name and an animal). We included only drug and gene entities with one-word names, as these names mapped to single nodes in the dependency graphs. The final drug lexicon contained 4008 unique terms, and the final gene lexicon contained 109,597 terms (many genes/proteins had multiple names).

6.2.2 Extraction of Dependency Paths from Medline Abstracts

We used the Stanford Parser [25] to generate dependency graphs for all sentences in Medline 2013 between 4 and 50 words in length (roughly 95% of all sentences in Medline). The input to the parser is a raw Medline sentence, and the output is a dependency graph. A dependency graph (see Figure 6.2) is one way to represent the grammatical architecture of a sentence; the nodes are words, and the edges are grammatical dependencies (grammatical relationships between pairs of words, described in detail in [24]).

A dependency path is a path through a dependency graph that connects two entities of interest. Considering a dependency path, instead of an entire sentence, can help prune out irrelevant terms and phrases and focus our attention on the part of the sentence directly relevant to the relationship between the two entities.

It is possible for a single sentence to generate more than one dependency path if multiple drug or gene names are present in the sentence. We oriented our paths so that they always started at the drug and ended at the gene, and we eliminated edge directions[4]. We eliminated paths containing dependencies of type *conj* [24], because

[4]We never observed a single situation where we accidentally collapsed paths with different meanings in doing so, since most pairs of words can only be connected by a particular dependency type, like

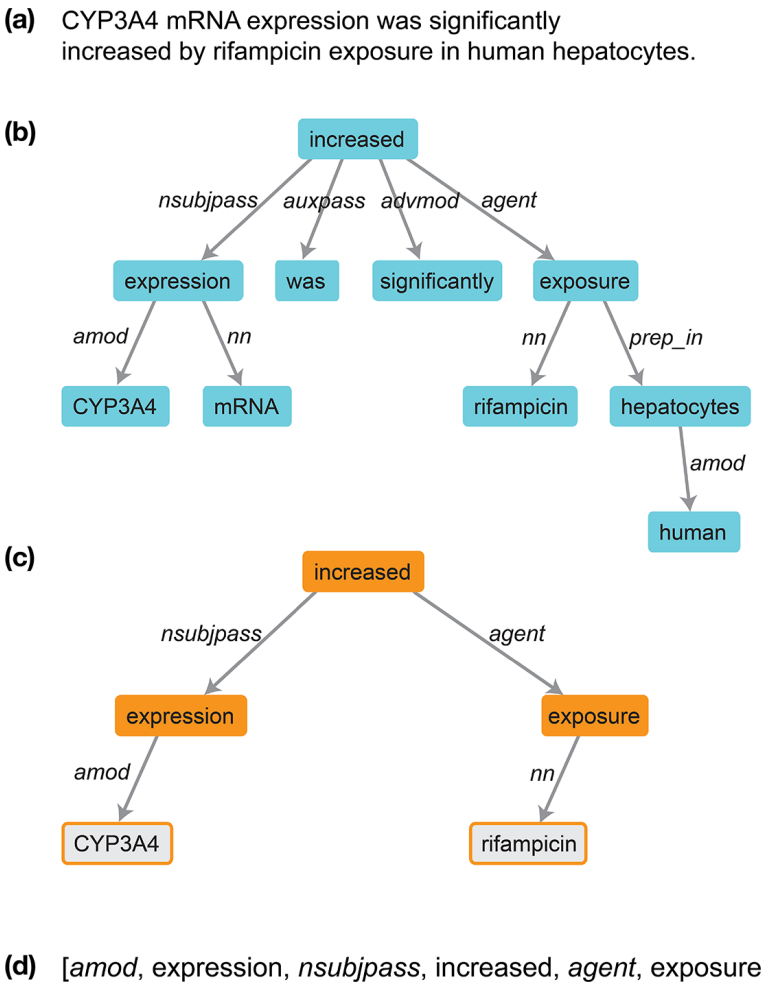


Figure 6.2. Example of a dependency graph for a Medline 2013 sentence. (a) The raw sentence. (b) The complete dependency graph for the sentence. (c) The dependency path connecting the gene CYP3A4 with the drug rifampicin. (d) A more compact representation of the dependency path.

these were usually errors arising from inadequacies in how the dependency parser represents lists. Note that because the dependency graphs are trees, there is one unique dependency path for each drug-gene pair in a sentence.

6.2.3 Results

The full set of abstracts from the 2013 edition of Medline contains approximately 184,000 sentences in which at least one drug name and at least one gene name are present. Many of these sentences contain multiple drug and gene names; the total number of unique drug-gene-sentence combinations is approximately 236,000.

As described in Section 6.2.2, we use dependency parsing to extract dependency paths between drugs and genes. Figure 6.2 illustrates how dependency paths are constructed from raw sentences. Table 6.1 provides some common drug-gene dependency paths and associated example sentences. Details about the meanings of the individual grammatical dependencies, with examples, can be found in [24].

We can quantitatively estimate the diversity of drug-gene descriptions in Medline by considering the space of all unique drug-gene dependency paths. The vast majority of dependency paths are rare, indicating high variability in how drug-gene relationships are described. The total number of unique drug-gene dependency paths in Medline is approximately 197,000, of which 7,272 (4%) connect at least two different drug-gene pairs. The total number of unique drug-gene pairs co-occurring in Medline sentences is 49,564, of which 14,052 (28.4%) share a dependency path with at least one other drug-gene pair.

Table 6.2 describes the two final datasets used in this project, which consist of matrices, M, created as described at the beginning of Section 6.2. Both of these datasets are over 99% sparse.

---

amod or *nn*, in one direction.

Table 6.1. Selected dependency paths and representative sentences. The drug and gene names flanking each path are bolded. Some key abbreviations are listed here: *appos*: appositional modifier, *amod*: adjectival modifier, *prep*: prepositional modifier (if *prep_of*, the specific preposition used is "of"), *nsubjpass*: passive nominal subject, *agent*: complement of passive verb, *dobj*: direct object of active verb, *nsubj*: noun subject of active verb.

	Dependency path	Example sentence (PubMed ID)	Frequency
[1]	[*appos*, inhibitor, *amod*]	**Geldanamycin** (GA), an **HSP90** inhibitor, is able to suppress 1,25-induced differentiation of HL60 cells. (20138989)	1181
[2]	[*appos*, inhibitor, *prep_of*]	The mNQO activity was insensitive to **dicoumarol**, a potent inhibitor of cytosolic **NQO1**. (10683249)	452
[3]	[*appos*, antagonist, *amod*]	The recommended therapy for stage III disease, based on clinical trials and by the Israeli Ministry of Health for 2006, includes **bosentan** (Tracleer), an **endothelin-1** antagonist. (18686806)	338
[4]	[*nsubjpass*, metabolized, *agent*]	**Amodiaquine** is mainly metabolized hepatically towards its major active metabolite desethylamodiaquine, by the polymorphic P450 isoform **CYP2C8**. (18855526)	204
[5]	[*nsubj*, inhibits, *dobj*]	**Salbutamol** inhibits **IFN-gamma** and enhances IL-13 production by PBMCs from asthmatics. (20523061)	118
[6]	[*nsubj*, inhibited, *dobj*, activity, *amod*]	**Clonidine** noncompetitively inhibited acetylcholinesterase activity in vitro and after in vivo administration at protective doses. (3761196)	73
[7]	[*appos*, antibody, *prep_against*]	**Trastuzumab**, a monoclonal antibody against **HER2**, has shown survival benefits when given with chemotherapy in all setting of HER2-positive breast cancer patients. (21129604)	71
[8]	[*nsubj*, increased, *dobj*, expression, *amod*]	**Carbachol** significantly increased **VEGF** expression in TMps, and this effect was totally reversed by methoctramine and pirenzepine. (15987429)	64
[9]	[*nsubj*, substrate, *prep_of*]	**Cyclosporin**, an immunosuppressant with a narrow therapeutic window, is a substrate for both **CYP3A4** and P-glycoprotein (Pgp). (12427482)	57
[10]	[*agent*, activated, *nsubj*pass]	These results suggest that **TRPV2** is specifically activated by **probenecid** and that this chemical might be useful for investigation of pain-related TRPV2 function. (17850966)	53
[11]	[*nsubj*, binds, *prep_to*]	**Pertuzumab** binds to **ErbB2** near the center of domain II, sterically blocking a binding pocket necessary for receptor dimerization and signaling. (15093539)	51
[12]	[*nsubj*, induces, *dobj*]	Tadalafil is mainly metabolized by cytochrome P450 (CYP) 3A4, and as **bosentan** induces **CYP2C9** and CYP3A4, a pharmacokinetic interaction is possible between these agents. (18305126)	30
[13]	[*nsubj*, increased, *dobj*, levels, *amod*]	When cells were cultured in a medium containing estrogen, **resveratrol** increased the **ErbB2** protein levels in a dose-dependent manner. (16488535)	29
[14]	[*prep_of*, metabolism, *prep_in*, involved, *nsubjpass*]	The results of preclinical studies demonstrated that **CYP3A4** is involved in the metabolism of **gefitinib** and that gefitinib is a weak inhibitor of CYP2D6 activity. (16176119)	21
[15]	[*nsubj*, inhibits, *dobj*, activation, *prep_of*]	**Imatinib** also inhibits the activation of **c-Abl**, which is a key downstream molecule of transforming growth factor-beta signaling, and PDGF receptors. (17603257)	17

Table 6.2. Summary of datasets for the PGx and drug-target relation extraction tasks. In the dense dataset, the drug-gene pairs and dependency paths represented must have occurred at least five times in Medline. In the sparse dataset, the dependency paths must have occurred at least twice, and all drug-gene pairs connected by these paths were included, even if they only occurred once.

Dataset	Task	Drug-gene pair	Dependency paths	Nonzero matrix elements (sparsity)	Known relationships in dataset	Optimal row and column cluster numbers
Dense	PGx	3514	1232	10,007 (99.8%)	290	$k = 30$, $\ell = 125$
	Drug-target				410	
Sparse	PGx	14,052	7272	29,456 (99.97%)	545	$k = 7$, $\ell = 25$
	Drug-target				779	

Table 6.3. Some dependency paths that cluster together with relatively high frequency.

First pattern	Second pattern	Frequency of co-clustering
[*nsubj*, antibody, *partmod*, directed, *prep_against*]	[*nsubj*, antibody, *partmod*, targeting, *dobj*]	0.59
D is an antibody directed against *G*.	*D* is an antibody targeting *G*.	
[*prep_such_as*, inhibitor, *amod*]	[*prep_including*, inhibitors, *amod*]	0.31
G inhibitor such as *D*	*G* inhibitors, including *D*	
[*prep_such_as*, agonists, *nn*]	[*amod*, activators, *nn*]	0.12
G agonists, such as *D*, ...	*G* activators, *D* and ...	
[*nsubjpass*, metabolized, *agent*]	[*dep*, substrates, *nn*]	0.11
D is metabolized by *G*	*G* substrates (*D*, ...) ...	
[*nsubj*, blocked, *dobj*, activation, *amod*]	[*nsubj*, inhibited, *dobj*]	0.07
D blocked *G* activation	*D* inhibited *G*	
[*nsubj*, increased, *dobj*, expression, *prep_of*, mRNA, *nn*]	[*nsubj*, induces, *dobj*, activity, *amod*]	0.03
D increased the expression of *G* mRNA	*D* induces *G* activity	

6.3 Examining Similar Dependency Paths

As in Section 6.1, we can also examine which columns of the pair-pattern matrix cluster together, as this provides insight into how the method is working. In Section 6.1, we observed that dependency paths with similar semantics clustered together. We see a similar pattern emerge when we examine co-clustering frequencies of the columns on a larger dataset: the dense dataset from Table 6.2. Table 6.3 shows some dependency paths from this dataset that frequently cluster together over 2000 separate runs of ITCC. Paths that frequently cluster together, again, appear to be semantically related.

6.4 Mapping the Drug-Gene Relation Landscape

Unsupervised EBC provides a measure of relationship similarity between every drug-gene pair and every other pair (the frequency with which each pair of rows in the data matrix cluster together). By combining these assessments with hierarchical clustering as described in Section 6.1, we created the dendrogram shown in Figure 6.3, the details of which are described in the figure caption. Section A.1 summarizes the general "themes" of the clusters from Figure 6.3 and includes the size of each cluster and the density of known PGx and drug-target relationships within that cluster.

6.4.1 Clustering drug-gene pairs based on EBC

EBC provides a natural measure of similarity for each drug-gene pair and every other pair: the number of times the rows corresponding to those two pairs clustered together over the N ITCC runs. However, as we have seen, these raw values are not fair measures of distance for all pairs, since some drug-gene pairs tend to cluster frequently with many other pairs, and others cluster less frequently. EBC's rank-based scoring function accounts for this by normalizing to ranks: each drug-gene pair ranks all other pairs by co-clustering frequency, and these ranks are used in place of the raw co-clustering values in the scoring function.

To implement EBC's scoring function in an unsupervised manner to construct our dendrogram, we started with our $m \times m$ matrix of co-occurrence values, C, in which C_{ij} was the number of runs (out of N total) in which drug-gene pair i co-clustered with drug-gene pair j. We then converted C into a correlation matrix, ρ, also $m \times m$, where ρ_{ij} contained the Spearman correlation of $C_{i\cdot}$ and $C_{j\cdot}$, the ith and jth rows of C (note that C is symmetric, so we could just as easily have used columns). These correlations are, as in EBC's scoring function, measures of how similarly drug-gene pair i and pair j rank all other pairs in the matrix, and are not biased in favor of promiscuous pairs. We then used $1 - \rho$ as the distance measure for hierarchical clustering using minimax linkage [5] to produce the dendrogram shown in Figure 6.3. Using a different linkage function or distance metric, obviously, would produce a different-looking dendrogram.

We used several R packages to produce the dendrogram figures, including *ape*

(a library for making phylogenetic trees), and *protoclust* (a library for hierarchical clustering using minimax linkage). To achieve the radially-spaced tip markers, we used a separate package[5].

6.4.2 Results

Figure 6.3 shows the dendrogram that we produced using the coclustering frequencies from unsupervised EBC along with hierarchical clustering using minimax linkage. Cluster 8, the largest cluster, contains drug-gene pairs whose descriptions mainly refer to inhibition. This cluster is highly enriched for both PGx and drug-target relationships. When cluster 8 is subdivided by cutting the dendrogram at a lower height, a subcluster (8a) of antagonists and their protein targets splits off from the main cluster. EBC has learned that antagonism is a subclass of inhibition. Cluster 10, which is a close relative of cluster 8 in the dendrogram, contains drug-gene pairs where the drug is both an inhibitor and a substrate of the protein, such as verapamil/P-glycoprotein.

Cluster 3, another large cluster, is almost exclusively devoted to metabolism and substrate relationships, and is highly enriched for PGx relationships, though not drug-target relationships. Cluster 3 contains three subclusters with slightly different properties. Cluster 3a involves mainly substrate relationships where the concept of "metabolism" is not mentioned. These include, for example, transport relationships like aminopterin/hOAT1. Cluster 3b contains most of the metabolic relationships, many of which involve liver cytochromes like CYP3A4 and CYP2D6. Cluster 3c includes substrate relationships where the drug is often also described as having an effect on the activity of the protein.

Other clusters enriched for drug-target relationships include cluster 12, where the protein is described as the receptor for the drug, cluster 14a, where the drug is described as an agonist of the protein, and cluster 15, which refers to protein binding. Notably, cluster 14a (agonists) is part of a larger cluster, cluster 14, that encompasses activation and stimulation relationships. Here, EBC has learned that agonism is a subclass of activation. Interestingly, cluster 14b, the part of cluster 14 that refers to

[5]https://github.com/willpearse/willeerd/blob/master/R/phylo.plots.R

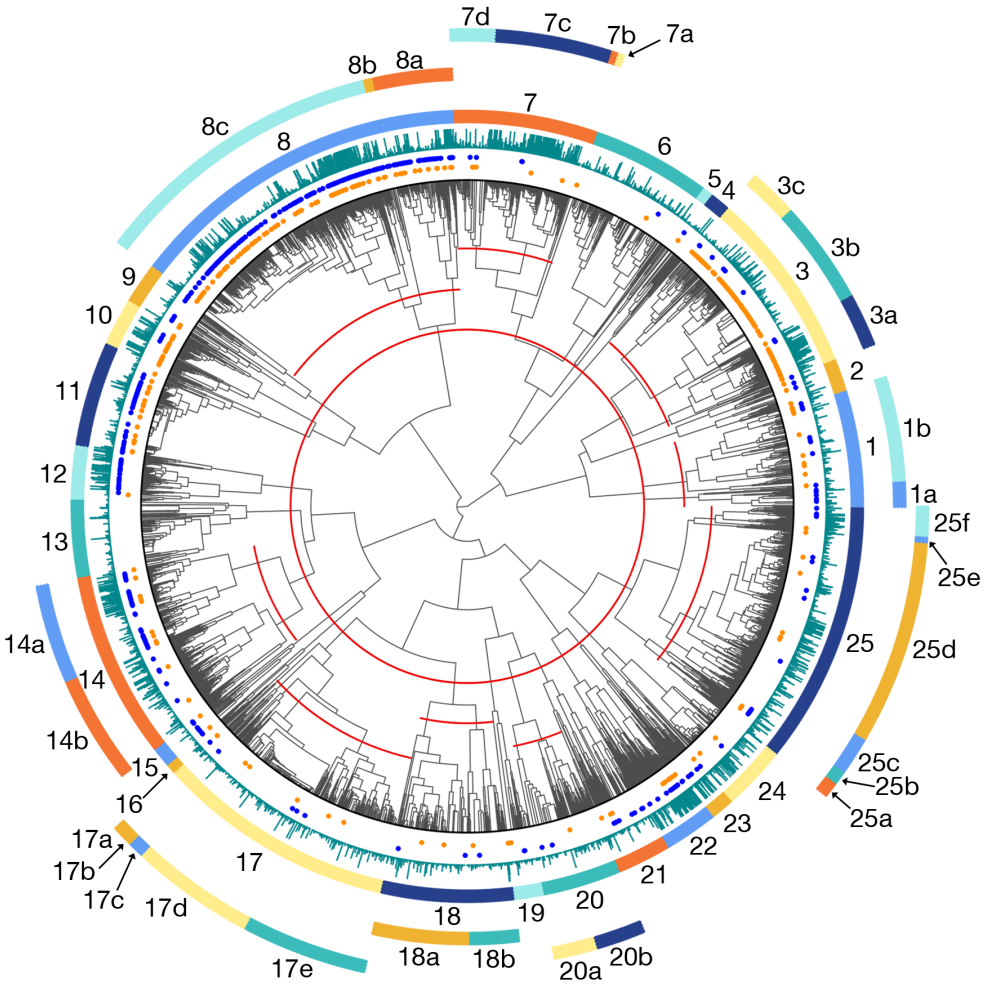


Figure 6.3. Dendrogram illustrating the semantic relationships among 3514 drug-gene pairs. In this dendrogram, the leaves represent 3514 drug-gene pairs that co-occur in Medline sentences at least 5 times, and we have cut the dendrogram at various levels (illustrated by the red lines in the interior of the dendrogram) to produce the colored clusters shown around the edges. Drug-gene pairs that are known drug-target relationships from DrugBank are denoted by blue dots, and those that are known PGx relationships from PharmGKB are denoted by orange dots. The heights of the turquoise bars are proportional to how often the corresponding drug-gene pairs co-occur in Medline sentences (a proxy for how well-documented they are).

activation more broadly and does not specifically refer to agonism, is not enriched for drug-target relationships.

Clusters 1-16, which comprise 3 of the 4 main high-level groups within the dendrogram, are relatively easy to interpret: in general, each displayed a consistent theme. Clusters 17-25, however, involve descriptions of experimental methods or results about drug effects on gene expression or protein activity. Here, the dendrogram reveals a distinction between past and present knowledge. Drug-gene pairs that are already well-studied are often reported in a static context - *D is an inhibitor of G*, or *D is a G agonist* whereas other pairs are reported primarily in an experimental context: *we investigated the effect of D on G expression*, *G was activated by D*, or *exposure to D significantly increased G activity*. Depending on the relative frequency of different types of descriptions, a drug-gene pair exemplifying an inhibitory relationship might end up in cluster 8 (mostly static descriptions) or cluster 21 (mostly experimental descriptions). Interestingly, drug-gene pairs from cluster 21 appear together in the literature significantly fewer times than drug-gene pairs from cluster 8 (median 9 times for cluster 21 vs. 16 times for cluster 8; maximum 66 times for cluster 21 vs. 2722 times for cluster 8; $p < 0.0001$, Mann-Whitney test), which seems to corroborate our assertion that the drug-gene pairs from cluster 21 represent more tentative experimental findings as opposed to well-established static knowledge.

Finally, the dendrogram reveals that PGx and drug-target relationships do not constitute distinct classes of relationships, but are chimeras. PGx relationships are composed of relatively distinct subgroups corresponding to (a) situations where the drug inhibits the gene/protein (and therefore, mutations in the gene could be expected to impact response to the drug), and (b) situations where the protein is involved in the metabolism or transport of the drug. Drug-target relationships overlap with (a) but not (b), and include other non-PGx subclasses, such as receptor binding and agonism.

6.5 Recognizing Drug-Gene Relations in Text

We evaluated EBC's ability to mine the literature for drug-gene pairs exemplifying two specific types of drug-gene relationships. The algorithm was given only the full,

unlabeled text of Medline and a small number of drug-gene pairs that exemplified each type of relationship. We refer to the small sets of labeled drug-gene pairs (sizes 1, 2, 3, 4, 5, 10, 25, 50, and 100) as *seed sets*. No text was annotated and no specific sentences were marked as "evidence" for any particular type of relationship. The two relationship types we examined were:

1. *Pharmacogenomic (PGx) relationships.* PharmGKB's relationships database [150] contains 6283 manually-curated drug-gene associations in which polymorphisms in the gene are known to impact drug response.
2. *Drug-target relationships.* DrugBank [152] maintains a list of known drug-gene relationships in which the protein product of the gene is a known target of the drug. This list contains 14,594 known relationships.

6.5.1 Evaluating PGx and Drug-Target Relation Rankings

For both the PGx and drug-target tasks, and for seed set sizes $|S| = 1, 2, 3, 4, 5, 10, 25, 50,$ and 100, we generated 1000 random seed sets and 1000 corresponding test sets, ensuring that the seed sets and test sets did not overlap. The test sets were all composed of 100 drug-gene pairs, 50 of which had known PGx or drug-target relationships and 50 of which did not. All of the ranking methods from Section 5.4.1 were used to rank the members of each test set, using its associated seed set for scoring.

We also explored the impact of data sparsity by performing these evaluations on two separate datasets. In the "dense" dataset, we included only drug-gene pairs and dependency paths that occurred at least five times in Medline. In the "sparse" dataset, we included dependency paths occurring at least twice, and any drug-gene pairs they connected (even if they only co-occurred in a single sentence). More information about the two datasets can be found in Table 6.2.

We evaluated the quality of each ranking by calculating the area under the receiver operating characteristic curve (AUC) [9], a measure of how likely it is that a positive element of the test set will be ranked higher than a negative element. We elected to use AUC instead of precision or recall because we wanted a threshold-independent measure of the overall quality of the ranking. We used R's *ROCR* package to calculate

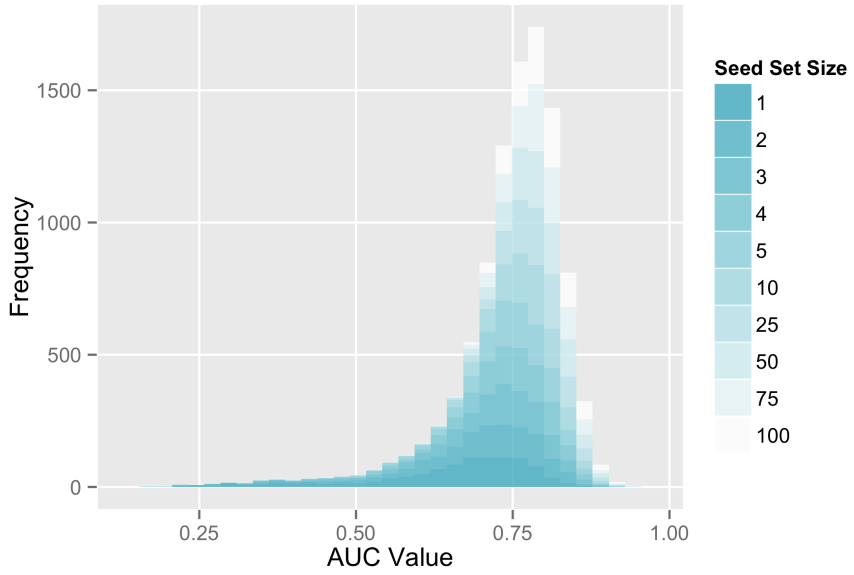


Figure 6.4. On the dense matrix, a histogram of the values of the AUC for the PGx task as a function of seed set size. The distribution is skewed (bimodal, actually, in the case of small seed set sizes), which is why we do not report mean values in our evaluation.

the AUCs.

From a practical standpoint, we were concerned mainly with the following scenario: Given that I have a seed set about whose quality I know nothing, what is the chance I can accurately prioritize the knowledge I am looking for within my [unlabeled] corpus? Our evaluation metric was, therefore, the fraction of the 1000 seed sets that ranked their corresponding test sets with AUC > 0.7. We did not use the median or mean AUC because the AUC distributions were highly skewed. In the case of small seed sets, they were actually bimodal: some seed sets steered the classifier in precisely the wrong direction; see Figure 6.4.

6.5.2 Results

Figure 6.5 shows EBC's performance extracting PGx and drug-target drug-gene pairs on the two datasets described in Table 6.2, and compares EBC to two alternative classifiers that do not account for the semantic relatedness of different dependency

paths.

On both datasets, and on both tasks, EBC outperforms the other classifiers by a significant margin. On the dense dataset, using seed sets of only 10 labeled drug-gene pairs as input, EBC accurately (AUC > 0.7) ranks 89.6% of test sets for the PGx task and 96.5% of test sets for the drug-target task. In comparison, using the same seed and test sets, the best-performing non-EBC classifier accurately ranks only 31.3% of test sets for the PGx task and 49.6% for the drug-target task. On the sparse dataset, EBC's increased performance is even more pronounced. Again using only 10 labeled pairs, EBC accurately ranks 54.4% of test sets on the PGx task and 90.4% on the drug-target task, compared to 1.1% and 6.3% for the best-performing non-EBC classifier.

6.6 New Relations for PharmGKB and DrugBank

EBC reliably detects new drug-gene pairs reflecting relationships of interest to PharmGKB and DrugBank, so we attempted to discover new examples from our corpus. We built seed sets containing all known relationships from PharmGKB and DrugBank and incorporated these into EBC to rank the remaining drug-gene pairs according to EBC's certainty that they represented PGx or drug-target relationships. There was 13.6% overlap between the two seed sets, with 84 drug-gene pairs in both, 206 in PharmGKB only, and 326 in DrugBank only, and 2898 pairs that were unknown to both.

The dendrogram shown in Figure 6.6 is identical to that in Figure 6.3, except that the clusters are replaced by vertical bars, the heights of which correspond to EBC's relative certainty that the pairs in question represent PGx relationships (shown in orange) or drug-target relationships (shown in blue). Known PGx or drug-target pairs are excluded from the bar graphs, but are denoted beneath the bars with orange or blue dots. As expected, we see high prediction certainty for drug-target and PGx relationships among the inhibitors in cluster 8, and high certainty for PGx relationships among the metabolic/substrate relationships in cluster 3. We also observe an interesting area of high enrichment for both types of relationships among clusters

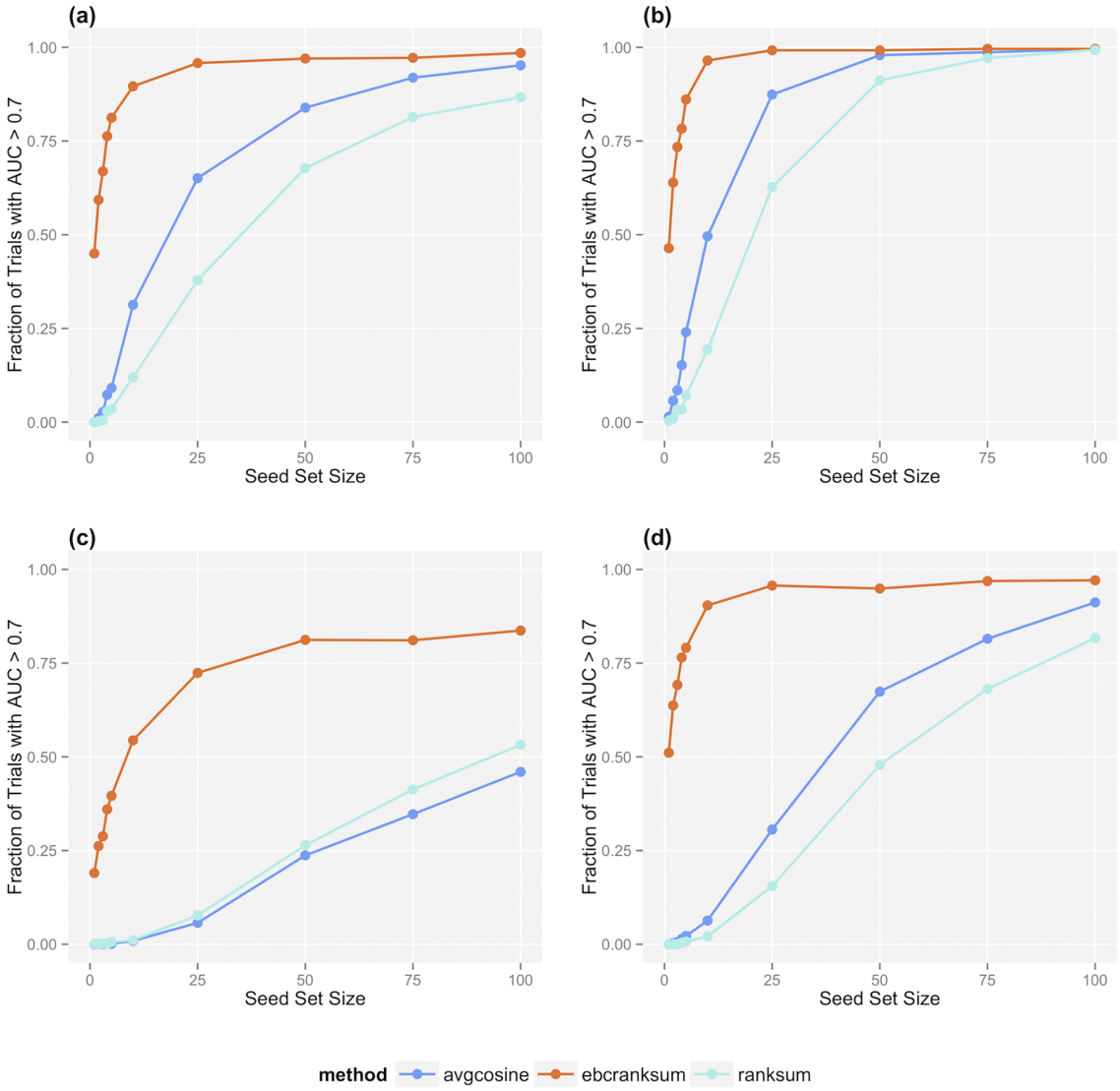


Figure 6.5. Classifier performance at the task of recognizing (a) PGx relationships (dense matrix), (b) drug-target relationships (dense matrix), (c) PGx relationships (sparse matrix) and (d) drug-target relationships (sparse matrix).

21-23, where inhibition is mostly reported in an experimental context, but the density of known PGx and drug-target relationships is quite low. These could represent new experimental findings that will be discussed as static knowledge in a few years.

Table 6.4 shows the top 20 predictions of new PGx candidate pairs for PharmGKB, and Table 6.5 shows the top 20 candidate drug-target pairs for DrugBank. Among the top 20 PGx predictions, five are already known to PharmGKB and have been demonstrated experimentally (one or more variants of the gene have been shown to impact response to the drug), but were coded in the PharmGKB relationships file in such a way that they were not included in the seed set. One is brand new: polymorphisms in ABCB1 (P-glycoprotein) do impact clinical response to fentanyl, but this relationship is currently unknown to PharmGKB. An additional eight pairs represent likely PGx relationships, such as known inhibitory or metabolic relationships, but no experiments have yet been conducted that might relate polymorphisms in the gene to drug response. And finally, in five cases, the potential for a PGx association was considered likely enough that it was investigated experimentally, but no significant clinical association between genotype and drug response was found.

Among the top 20 predictions for new drug-target relationships for DrugBank, four are already known but were listed in DrugBank under alternate gene names. An additional seven are new, proven drug-target relationships. Of these, five involve drugs that are themselves unknown to DrugBank (there are not yet entries for ketanserin, cangrelor, nutlin-3, or tropisetron in DrugBank). There are also several interesting, yet erroneous findings arising from parser and lexicon errors in which a molecule, such as IL-1, is mistaken for its receptor, and that receptor is the true target of the drug. These are explored further in the Discussion.

6.7 Comparing EBC to LSA

Several authors before us (Section 3.2.2) have pointed out that patterns co-occurring with similar entity pairs have similar meanings [77], and that entity pairs connected by similar patterns have similar semantic relations [145]. These ideas form the basis for *distant supervision*, a technique commonly employed in traditional relationship

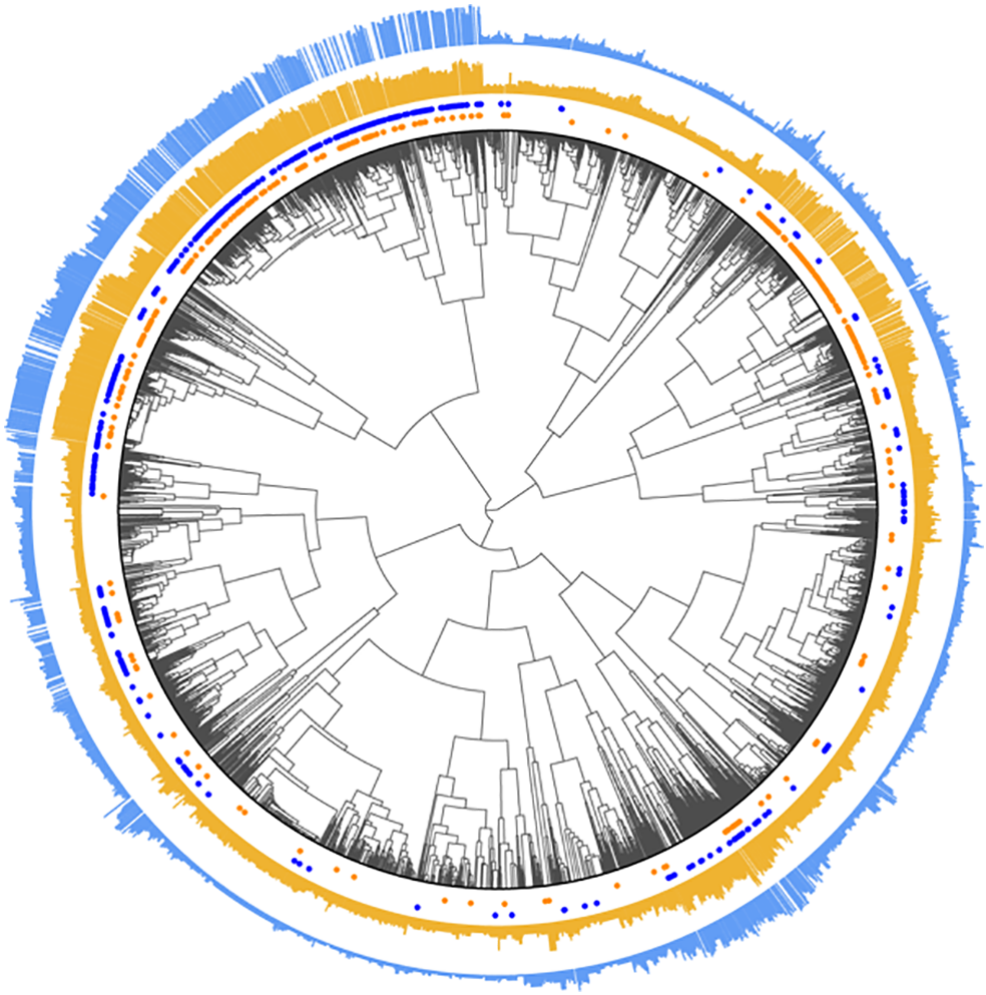


Figure 6.6. Dendrogram illustrating predictions of novel PGx and drug-target relationships among 3514 drug-gene pairs. The height of the bars corresponds to EBC's certainty that the pair in question represents a relationship of the corresponding type (orange: PGx relationships, blue: drug-target relationships). The dots represent known PGx and drug-target relationships, as in Figure 6.3.

Table 6.4. Top 20 predictions of new drug-gene relationships for PharmGKB, and whether a PGx relationship has been documented in the literature. $***$ indicates that an association has been demonstrated experimentally between changes in the expression/activity of the gene/protein and the efficacy of the drug, $**$ indicates that such an association is likely, but has not yet been studied, and $*$ indicates that the association has been studied experimentally, and the experiment has refuted the association. Here we include only associations between pharmaceutical compounds and single genes; predicted associations involving endogenous compounds and/or groups of genes are included in the supplementary material for the paper, however.

	Candidate drug-gene pair	Relative certainty	Literature reference (PMID)		Comment
[1]	omeprazole, CYP2C19	1.000	11069321	***	Individual polymorphisms of CYP2C19 already associated with omeprazole in PharmGKB.
[2]	mexiletine, CYP1A2	0.995	9690950	**	
[3]	fentanyl, P-gp	0.994	17192767	***	
[4]	voriconazole, CYP3A4	0.986	17433262	**	
[5]	cyclosporine, CYP3A4	0.983	18978522	***	Association listed in PharmGKB as "ambiguous".
[6]	duloxetine, CYP1A2	0.983	18307373	**	
[7]	fluconazole, UGT2B7	0.982	16542204	**	
[8]	montelukast, CYP2C8	0.973	21838784	**	
[9]	dydrogesterone, AKR1C1	0.968	20727920	**	
[10]	voriconazole, CYP2C9	0.966	16940139	*	
[11]	imipramine, FMO1	0.962	19262426	***	Experiment conducted in mice.
[12]	ticlopidine, CYP2C19	0.961	21178986	*	
[13]	moclobemide, MAO-B	0.960	7586937		In this article, MAO-B activity was studied in relation to moclobemide response, but specific polymorphisms were not investigated.
[14]	ritonavir, P-gp	0.958	16184031	***	Association listed in PharmGKB as "ambiguous".
[15]	cyclosporin, MDR1	0.955	15116055	*	
[16]	cyclosporin, P-gp	0.952	15116055	*	Same gene as [15].
[17]	vinblastine, P-gp	0.951	16917872	***	Association listed in PharmGKB as "ambiguous".
[18]	amprenavir, CYP3A4	0.950	9649346	**	
[19]	perazine, CYP1A2	0.945	11026737	**	
[20]	lopinavir, ABCB1	0.939	21743379	*	

Table 6.5. Top 20 predictions of new drug-target relationships for DrugBank. $***$ indicates that the drug has been shown experimentally to have modified the activity of the gene/protein, $**$ means that the interaction is known to DrugBank but is listed under an alternate drug or gene name, $*$ means the interaction has been studied and is unlikely. P refers to a particular type of parser error in which the ligand of a receptor is mistaken for that receptor; L refers to a lexicon error.

	Candidate drug-gene pair	Relative certainty	Literature reference (PMID)		Comment
[1]	ketanserin, 5-HT2A	1.000	16615363	***	Ketanserin not in DrugBank.
[2]	losartan, A-II	0.998	24807206	**	"A-II" refers to the angiotensin type II receptor. In DrugBank this is listed as "Type-1 angiotensin II receptor".
[3]	cangrelor, P2Y12	0.993	20048234	***	Cangrelor not in DrugBank.
[4]	phencyclidine, nAChR	0.992	9862757	***	Phencyclidine is a noncompetitive inhibitor of nAChR.
[5]	anakinra, IL-1	0.991		P	
[6]	bosentan, endothelin-1	0.987		P	
[7]	imatinib, EGFR	0.985	15887238	*	Imatinib's effect on EGFR is ambiguous. It is not likely to be a direct target.
[8]	propanolol, Beta2	0.984		P	
[9]	carvedilol, Alpha1	0.984		P	
[10]	MK-571, leukotriene	0.983		L	MK-571 is unknown to DrugBank.
[11]	zafirlukast, leukotriene	0.981		L	
[12]	degarelix, GnRH	0.980		**	GnRH receptor listed in DrugBank as "Gonadotropin-releasing hormone receptor". Complicated because degarelix often referred to as "GnRH antagonist" but the target is actually the GnRH receptor.
[13]	nutlin-3, Mdm2	0.980	18646312	***	Nutlin-3 disrupts the p53-Mdm2 complex. Nutlin-3 is unknown to DrugBank.
[14]	genistein, EGFR	0.979	21603581	***	Interestingly, authors found that genistein promotes cancer progression and increases EGFR signaling.
[15]	montelukast, leukotriene	0.977		L	
[16]	aprepitant, NK-1	0.977		**	NK-1 listed in DrugBank as "Substance-P receptor".
[17]	staurosporine, calmodulin	0.975	1846174	*	Staurosporine inhibits calmodulin-dependent protein kinase, not calmodulin.
[18]	nutlin-3, Hdm2	0.975	19696166	***	Nutlin-3 is unknown to DrugBank. Hdm2 refers to the human version of the Mdm2 protein ([13], above).
[19]	tropisetron, 5-HT4	0.974	11243577	***	Tropisetron is unknown to DrugBank.
[20]	basiliximab, CD25	0.972	12591363	**	CD25 is listed in DrugBank as "Interleukin-2 receptor subunit alpha".

extraction tasks, and Latent Relational Analysis [146] (which is in turn based off of the famous distributional semantics method Latent Semantic Analysis (LSA) [26]).

To compare EBC's performance to a more established method designed to solve a similar problem, we used the singular value decomposition (SVD) [137] to decompose the sparse and dense data matrices, creating "compressed" feature vectors of reduced dimensionality (of various lengths) for each drug-gene pair and incorporating these, rather than the raw row vectors, into the two non-EBC ranking methods described in the Methods (AvgCosine and RankSum). This approach is virtually identical to (LRA; [146]), except that our matrices are binary while the original matrices in the LSA and LRA papers used weighted counts.

The results of the PGx relationship extraction task on the dense and sparse matrices are shown in Figure 6.7. The RankSum ranking method appeared to work the best for LSA. We include the results for the RankSum method used on the original feature vectors (denoted by "RankSum") and on various lengths of compressed vectors (denoted by their vector lengths). For comparison, we also include the results for EBC.

We see that the performance of classifiers that rely on the SVD for dimensionality reduction strongly depends on the length of the compressed feature vectors, and the best-performing vector length varies with the size and structure of the data matrix. For the dense matrix, the optimal vector length was somewhere between 5 and 10 (depending on the size of the seed set), while for the sparse matrix, vectors of length 5 or 10 performed horribly, and the optimal length was somewhere between 1000 and 7272 (the uncompressed vector length). For the dense matrix, if the correct vector length was chosen, the results came close to those of EBC, but for the sparse matrix (unless there was a specific vector length between 1000 and 7272 that led to a rapid increase in performance) the results never approached EBC's.

More importantly, in a real curation scenario, the vector length would need to be chosen using a development set (necessitating additional training data beyond the seed set) or via some other heuristic, and the strong relationship between vector length and classifier performance means we would be unlikely to choose optimal values randomly. The authors of the original LSA paper specifically mentioned the choice of vector length as one of the major challenges facing their algorithm, so the heuristic in

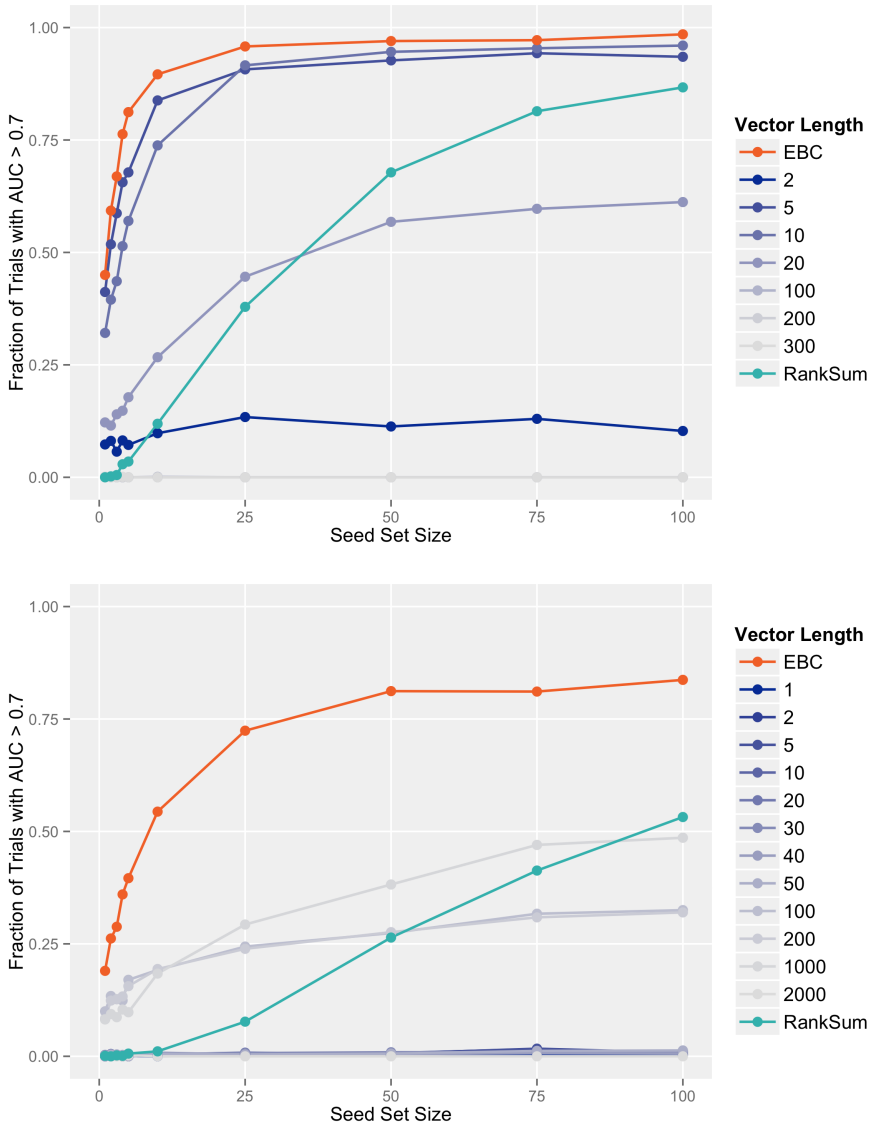


Figure 6.7. Comparison of EBC's performance to Latent Semantic Analysis (LSA).

Section 5.3 turns out to be a key advantage for EBC.

6.8 Rethinking the Relation Extraction Problem

Why is biomedical relationship extraction so challenging? Why don't more databases like OMIM, DrugBank, and PharmGKB incorporate NLP-curated relationships? We believe that one key stumbling block lies in how the problem has historically been defined. Biomedical relationship extraction is usually thought of as a sentence-level problem: does a particular sentence describe a specific type of relationship or not? However, as we have seen, sentence-level descriptions are highly erratic. Faced with a bewildering array of possibilities for how similar relationships can be described, sentence-level relationship extraction algorithms often rely on manually-constructed rules or ontologies that map diverse surface forms onto common semantics[6] [20,35,114]. These systems require a non-trivial amount of human maintenance and must be rebuilt for each new domain. Machine learning algorithms for sentence-level relationship extraction avoid rules but face another serious problem: the need for annotated training sentences. Recently, researchers have begun to produce annotated training sets for the biomedical domain [47,59] but manual annotation is almost as expensive as manual curation, both in time and human effort. As a result, little to no annotated training data exist for many classes of biomedically interesting relationships.

These are important problems for NLP, but they only exist because we think of biomedical relationships at the level of individual sentences. From a biomedical research standpoint, there is no need to do so. We are most interested in the true relationship between a drug and a gene, not in the meaning of any particular sentence. As a result, we have taken a corpus-level approach where all of the information about a drug-gene pair from all of its available sentence-level descriptions is combined. Latent connections among different-looking descriptions are then discovered in an unsupervised fashion from structure inherent in the raw text, requiring no human effort and boosting our ability to extract relationships of interest.

[6] http://www.technologyreview.com/news/523411/facing-doubters-ibm-expands-plans-for-watson/

6.8.1 Support for Corpus-Level Inference

We contend that biomedical relationships should be considered properties of biomedical entities like drug-gene pairs, not individual sentences. A description like *D decreased G levels* does not constitute an inhibitory relationship; it is simply an experimental finding that increases the likelihood of such a relationship. This allows the same sentence to provide evidence for or against multiple types of relationship, the exact definitions of which are application dependent. It also allows drug-gene pairs to exhibit multiple relationship types at once.

We see evidence for such an approach when we contrast EBC's performance at extracting PGx relationships with its performance extracting drug-target relationships. EBC was uniformly worse at extracting PGx relationships, even though these two sets of experiments used the same data matrices. We see why in Figure 6.3: it turns out that what we originally considered to be well-defined relationship classes (PGx and drug-target relationships) are actually composites of several finer-grained sub-classes. A high percentage of PGx relationships reside in cluster 3, the metabolism/substrate cluster, which inhabits a region of the dendrogram far from the inhibition clusters. In cases where the seed set consists mostly of metabolic relationships and the test set mostly of inhibition relationships, we would not expect EBC to perform well, even though both groups are still technically PGx relationships.

We initially believed that PGx relationships would be expressed in sentences relating specific polymorphisms to changes in drug efficacy, such as, *The CYP3A4 C3435T polymorphism influences rifampicin exposure in human hepatocytes.* In reality, however, relatively few such sentences exist. Most evidence for PGx relationships comes instead from descriptions of other types of relationships, such as inhibition and metabolism. So we see that although a PGx relationship can be considered a property of a drug-gene pair, it is not generally a property of any particular sentence describing that pair.

6.9 Study Limitations

In our analysis of drug-gene relationships, we made several choices about (a) how to identify drugs and genes in text, (b) the type of text to use as our corpus, and (c) what constitutes a pattern (a single column in the data matrix). In all cases, we made the simplest choices possible, both to enable others to reproduce our results, and to distinguish EBC's own limitations from errors/omissions in the preprocessing steps and text itself.

We identify drugs and genes in the text based on simple string matching to single-word drug and gene names from PharmGKB [62, 150]. Named entity recognition (NER) is its own area of NLP, and identifying biomedical entity names in text is itself a nontrivial proposition. We can see one obvious disadvantage of this approach in cluster 24 of Figure 6.3 and Section A.1, which include "gene names" like COPD (a.k.a. chronic obstructive pulmonary disease) and NIDDM (non-insulin-dependent diabetes mellitus). Table 6.5 also reflects a lexicon error where the term "leukotriene" is listed as a synonym for the leukotriene B4 receptor. Some such errors might be avoided if we used a more elaborate NER system [69, 70], though such systems themselves are not perfect and can introduce new sources of error. Our stipulation that the entity names be single words also led to errors in cases (see Table 6.5) where a molecule, such as IL-1, is mistaken for its receptor, the IL-1 receptor, because "IL-1 receptor" is a multi-word phrase not allowed in the lexicon, while "IL-1" is allowed.

We also made no attempt to normalize gene names, so in our results, ABCB1, MDR-1, and P-gp are all different. Again, this was done to avoid introducing normalization errors, and because genes and their corresponding proteins are often described in different contexts.

To construct dependency paths from raw Medline sentences, we used the Stanford Parser [25], a free and open-source statistical parser. The Stanford Parser was trained using labeled text from newswire corpora, so it sometimes fares poorly on biomedical text. For example, the parser often mistakes gene names for adjectives ("CYP3A4" in the phrase "CYP3A4 polymorphism" is frequently mislabeled as an adjective). We used the out-of-box implementation of the Stanford Parser and did not perform

any manual review or correction of parses to improve its performance (again in the interest of simplicity). Because EBC operates at the level of drug-gene pairs and not individual sentences, its performance is generally robust to parsing errors as long as the parser makes the same errors consistently.

There are some errors that do lead to incorrect conclusions, however. For example, we observe some situations where dependency paths bypass important details about relationships, such as a sentence where a drug is described as *transcriptionally up-regulating G expression* and the dependency path only captures the effect on expression, not its directionality. These are usually generalizations rather than errors, but they do result in some loss of information from the sentence.

Finally, our corpus consisted of all abstracts from the 2013 edition of Medline. Including information from the full text of the research articles could help discover relationships not mentioned in the abstracts, but many journals do not provide access to the full text, and we did not wish to bias our results in favor of relationships reported in a subset of journals. Our approach would remain the same regardless of the corpus.

6.10 Extensions and Future Applications

The combination of EBC and dependency path features described here allows us to reliably extract biomedical relationships of interest from Medline sentences, smoothing over differences in how these relationships are described. This finding opens the door to many interesting possible future applications. For example, EBC could be used to extract relationships spanning multiple sentences or entire abstracts by using features such as individual dependencies, words, or phrases in place of dependency paths. As new gold-standard sets of biomedical relationships become available (such as all drug-gene pairs reflecting inhibitory relationships or specific collections of drug-gene pairs relevant to particular laboratories' research efforts) these can seamlessly be incorporated into EBC to extract these relationships at scale. EBC could also potentially be used for lexicon or ontology expansion in a manner similar to LSA or random indexing [104, 120].

At its core, EBC is not relationship extraction-centric. The algorithm itself is agnostic to the type of data contained in its input matrix. EBC simply allows us to use latent structure in large, unlabeled datasets to boost our ability to extract new information from those datasets, even when our access to labeled training examples is limited. Datasets like these occur throughout biomedical research, even beyond NLP. We look forward to seeing how EBC fares on some other classes of related problems, in NLP and elsewhere.

Chapter 7

A Global Network of Biomedical Relationships

In Section 6.4, we saw how unsupervised clustering of drug-gene pairs, based on the similarities of their connecting dependency paths as assessed by EBC, could be used to discover characteristic "modes" by which drugs and genes interact. In this chapter, we extend the ideas from Chapter 6 to four different classes of biomedical relationships: chemical-gene (which encompasses both chemical-gene and chemical-protein relationships, and is similar to Chapter 6), chemical-disease (which encompasses relationships both with diseases and with non-disease phenotypes such as drug side effects), gene-disease (which includes all combinations of gene/protein and disease/non-disease-phenotype), and gene-gene (also protein-protein). Our finished product is a network of labeled biomedical relationships of these four different classes, produced entirely from the structure of relationship descriptions in the literature.

7.1 Dependency Paths and Datasets

7.1.1 Named Entity Recognition using PubTator

The PubTator annotations (see Section 4.3.2), which were released after the initial development of EBC, provide high-quality named entity annotations of chemicals,

genes, and diseases for all of Medline. Whereas in Chapter 6 we were forced to resort to simple string matching of single-word lexicon terms for drugs and genes, the PubTator annotations extend our coverage to multi-word terms, which has proved especially valuable for relationships involving diseases and side effects.

PubTator annotations for a single abstract consist of the full text of the abstract, its title, and a series of annotated concepts for which it provides: the location and string in the raw text that matched the concept, its entity type (chemical, gene, disease, etc.) and its closest database identifier[1]. There are approximately 16.5 million Medline abstracts annotated by PubTator as of this writing. Annotations are updated monthly. Our version of the PubTator annotations was downloaded on April 30, 2016.

7.1.2 Extraction of Dependency Paths

As in Chapter 6, we extracted all dependency paths connecting two recognized biomedical entities within a single sentence. This was a two-step process:

1. We used the PubTator annotations to concatenate phrases corresponding to annotated biomedical entities; for example, the phrase *cytochrome p450 3A4*, if identified as an entity by PubTator, was changed to *cytochrome_p450_3A4* (using the underscore). This concatenation step was performed first, before any parsing was done.
2. We divided the annotated and concatenated abstracts into sentences and parsed each sentence using the Stanford Dependency Parser [25]. From there, we found the dependency paths connecting (a) chemicals and genes, (b) chemicals and diseases, (c) genes and diseases, (d) genes and genes, using the method from Section 6.2.2.

The extraction of the gene-gene paths introduced an additional layer of complexity, since there is no natural way to order the paths (since both the start and end entities are proteins). We therefore extracted two paths for a sentence connecting G_1 and G_2: the path from G_1 to G_2, and the path from G_2 to G_1.

[1]PubTator matches strings to a variety of databases, including NCBI Gene (Gene), MEDIC (Disease), the NCBI Taxonomy (Species), MeSH (Chemical), and NCBI dbSNP (Mutation). For more information about its performance, please see Table 1 of [149]

7.1.3 Creating the Data Matrices

To prepare the data for unsupervised clustering using EBC, we selected the most frequent ≈ 700 dependency paths connecting (a) chemicals and genes, (b) chemicals and diseases, (c) genes and diseases, (d) genes and genes, and sampled 2000 entity pairs from the total set connected by one or more of those paths. Descriptions of the full datasets and these downsampled datasets can be found in Table 7.1.

Table 7.1. Descriptions of datasets for all four interaction types. The top four datasets were used with EBC to obtain similarity scores for the different dependency paths, which were then combined with hierarchical clustering to uncover likely relationship classes.

Matrix Type	Dependency Paths	Entity Pairs	Minimum Path Occurrences	Nonzero Elements	Row Clusters (K)	Column Clusters (L)
Chemical-Gene	697	2000	5	6276	100	100
Gene-Gene	636	2000	5	6022	150	170
Gene-Disease	739	2000	12	6450	190	150
Chemical-Disease	693	2000	100	7903	90	70
Chemical-Gene (full)	215,732	167,018	1	275,200		
Gene-Gene (full)	278,616	263,336	1	376,292		
Gene-Disease (full)	465,103	406,707	1	682,048		
Chemical-Disease (full)	4,594,012	1,900,501	1	6,078,346		

7.2 Creation of Relationship Classes

The analysis in this chapter differs from that in Chapter 6 in one important regard: we perform hierarchical clustering on the *dependency paths*, rather than the *entity pairs*, to create labeled relationship classes. This idea was inspired by our finding in Section 6.3 that semantically similar dependency paths tend to cluster together during EBC. It was also inspired by practicality: labeling the dendrogram in Figure 6.3 relied upon a human annotator (read: me) who manually examined drug-gene pairs in the different clusters, along with their connecting dependency paths, to assign labels to the clusters. It is much easier for a human without extensive prior knowledge of pharmacology to assign labels to clusters of dependency paths, which represent real textual patterns (e.g. "D is an inhibitor of G"), than it is to assign labels to clusters of entity pairs.

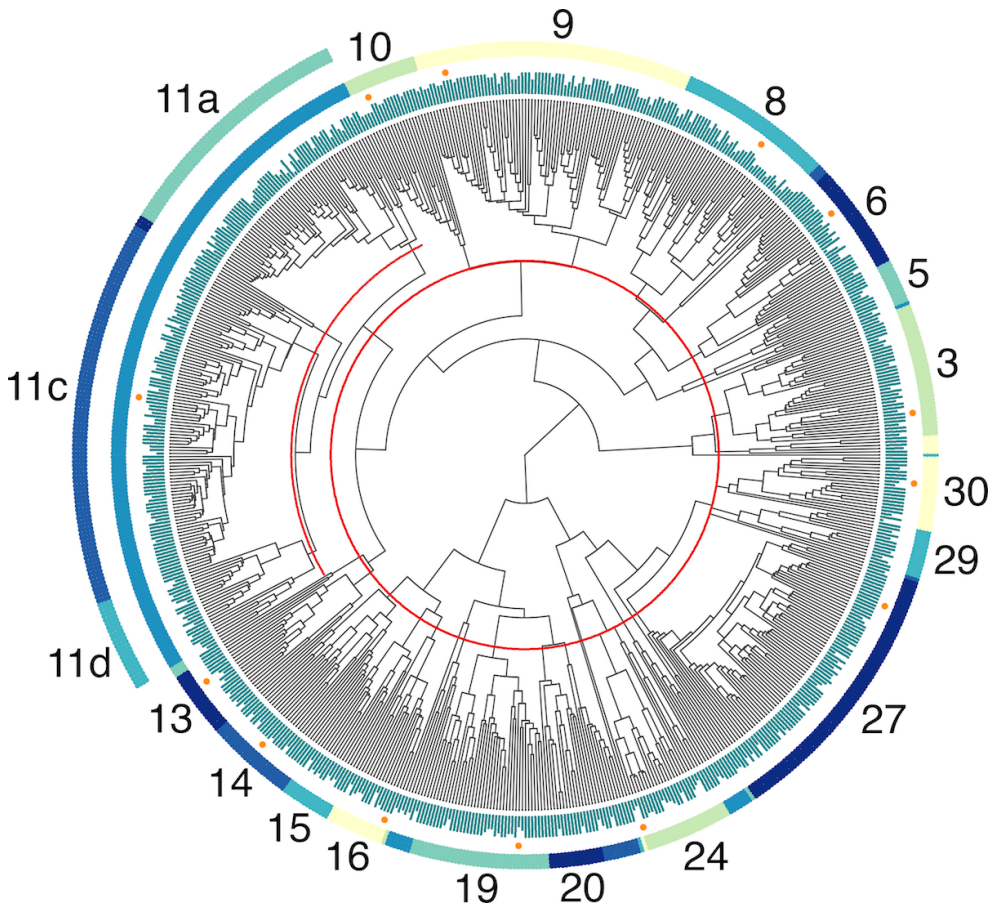


Figure 7.1. Dendrogram of dependency path classes among chemical-gene pairs, using PubTator annotations to identify chemical and gene names in the text. Each leaf node represents one dependency path.

We applied the same basic method as described in Section 6.4.1 to the downsampled data matrices from Table 7.1 to produce the dendrograms shown in Figures 7.1, 7.2, 7.3, and 7.4. In these dendrograms, each leaf node represents one dependency path, rather than one entity pair.

7.2.1 Cluster Labeling

We cut the dendrograms in Figures 7.1, 7.2, 7.3, and 7.4 at a level that produced 30 clusters. Any clusters of 10 or fewer dependency paths that emerged were not examined further, and upon visual inspection, very large clusters with obvious internal

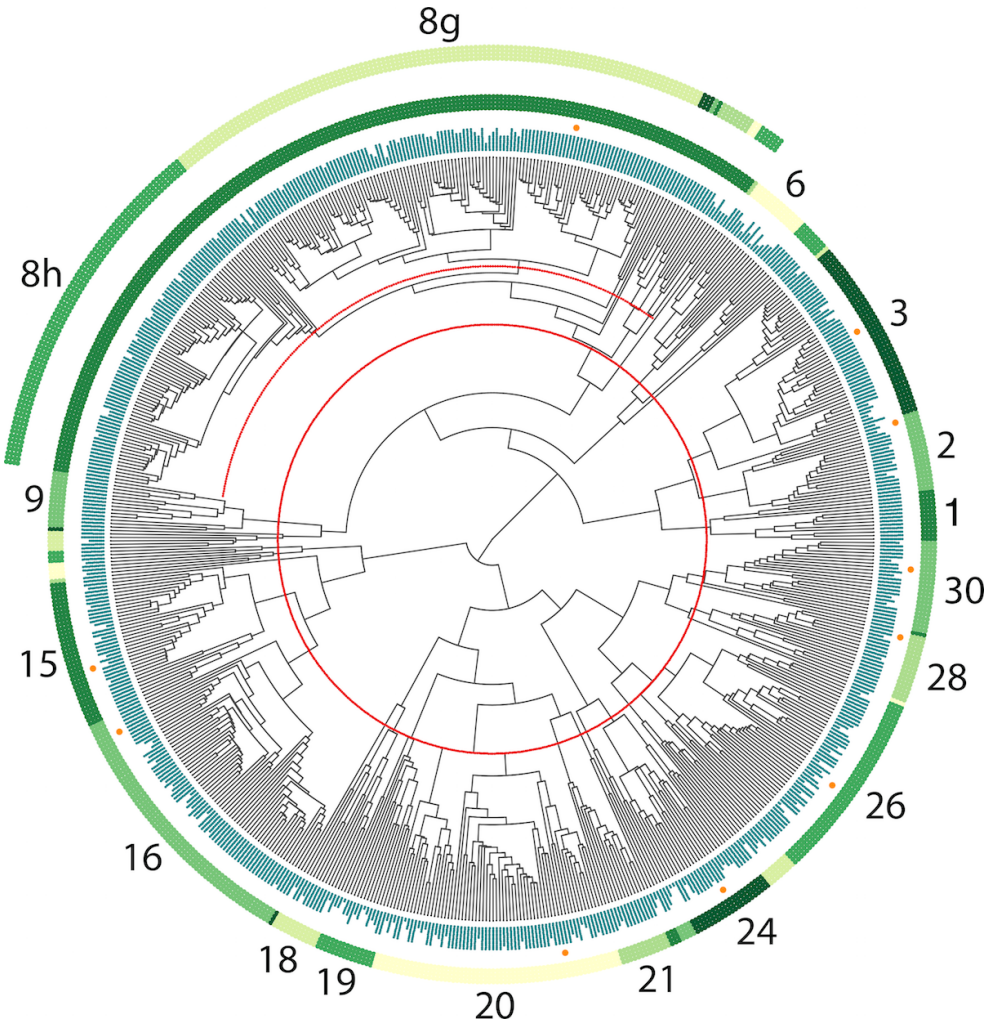


Figure 7.2. Dendrogram of dependency path classes among chemical-disease pairs, using PubTator annotations to identify chemical and disease names in the text. Each leaf node represents one dependency path.

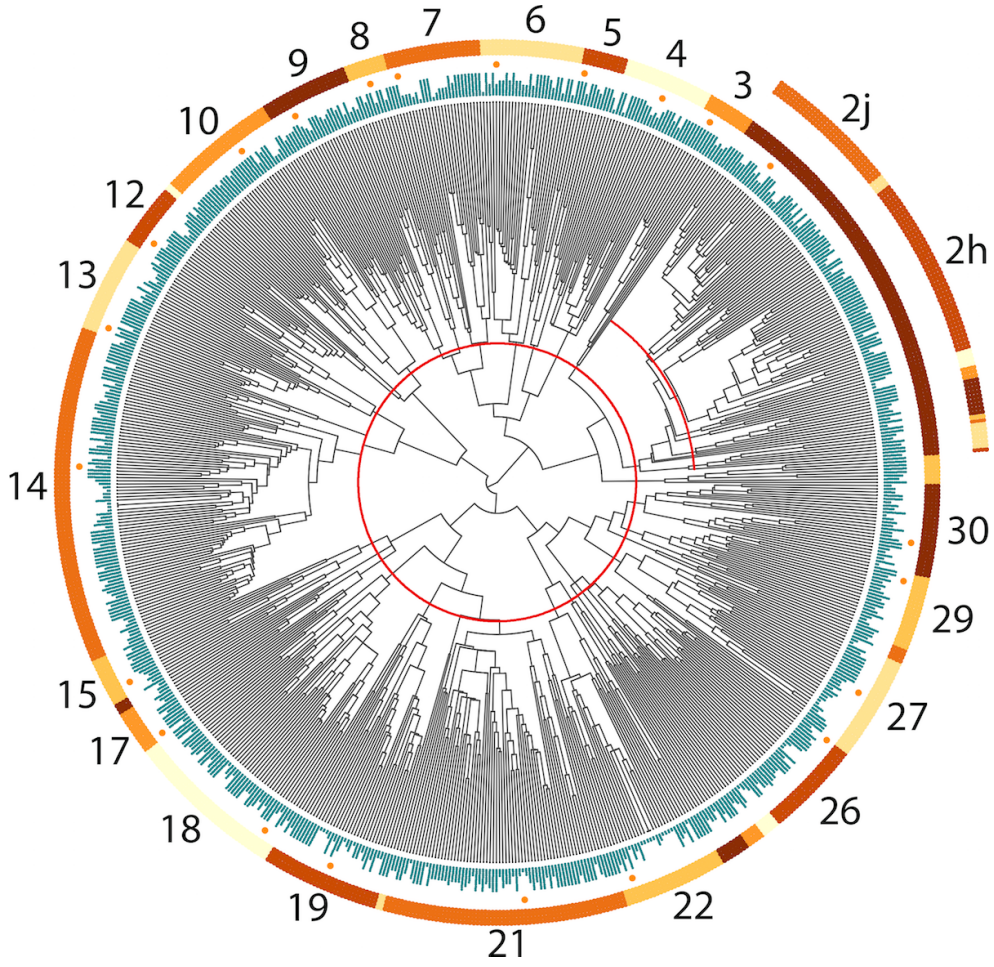


Figure 7.3. Dendrogram of dependency path classes among gene-disease pairs, using PubTator annotations to identify gene and disease names in the text. Each leaf node represents one dependency path.

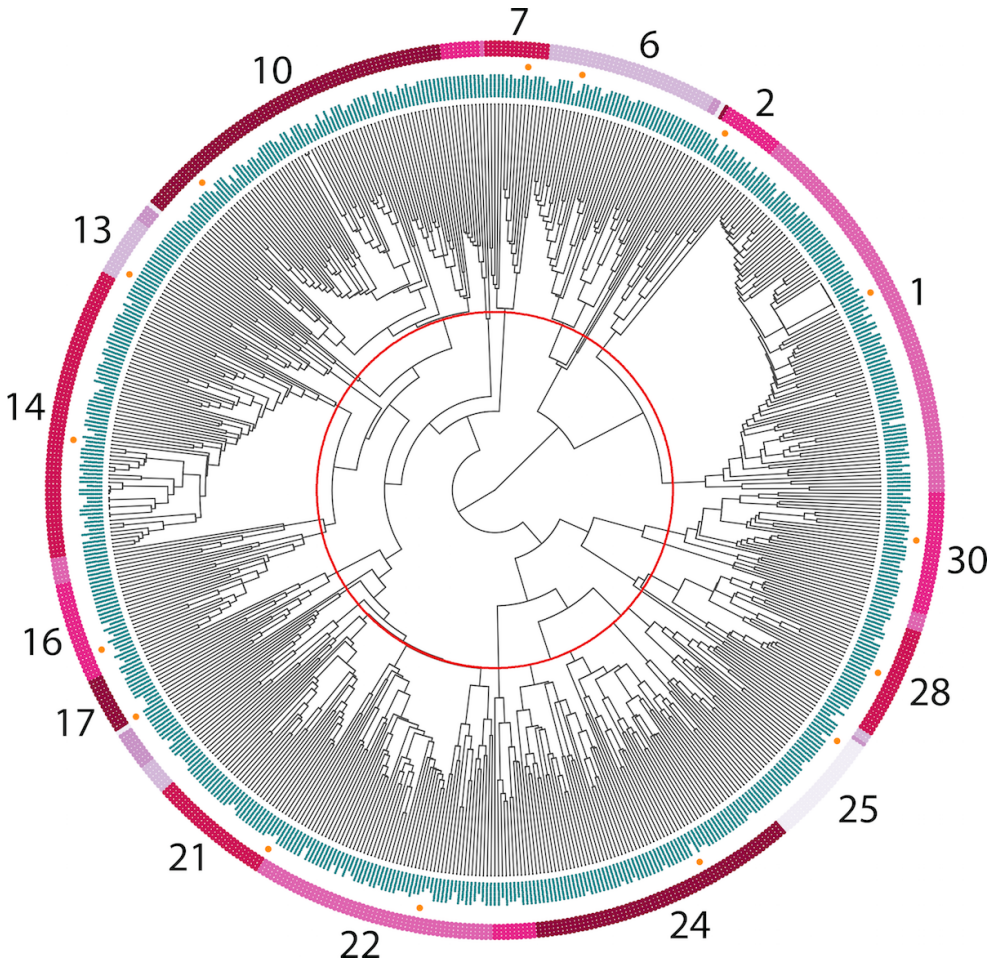


Figure 7.4. Dendrogram of dependency path classes among gene-gene pairs, using PubTator annotations to identify gene names in the text. Each leaf node represents one dependency path.

structure were cut further down to produce smaller subclusters. For each cluster, a set of 10 dependency paths was selected at random, and a human annotator (again: me) examined the paths and several associated example sentences from the literature to deduce a label. The full results of this labeling process occupy 16 pages in Appendix A.

7.2.2 Simplified Relationship Themes

As expected, nearby clusters sometimes shared similar themes. Occasionally, clusters that were not close together in the dendrograms also shared similar themes. This most often occurred when the same relationship type was described in slightly different ways within distinct groups of entity pairs. For example, clusters 6, 15, and 16 in Figure 7.2 all referred to descriptions of side effects or adverse events related to the administration of a chemical, yet cluster 6 was also closely related to clusters 8 and 9, which described investigations of experimental agents.

We simplified the clusters from the tables in Section A.2 into thematically-related groups and assigned each group a symbol. The complete list of groups can be found in Table 7.2. Two of the groups in the chemical-gene dendrogram contained relationships where we perceived the directionality to be important for future applications: activation (agonism vs. antagonism, cluster 6) and changes in expression (up, down, or neutral, clusters 8-10). By cutting the clusters further down, we could potentially have separated some of these directional changes, but the clusters were small enough that we decided to simply label the positive and negative directional dependency paths manually to ensure perfect separation. This is what the "+" and "-" signs refer to in Table 7.2.

7.3 Properties of the Relationship Clusters

7.3.1 Chemical-Gene Relationships

Figure 7.1 shows the hierarchical clustering of the chemical-gene dependency paths. Full descriptions of the labeled clusters can be found in Table A.2 and simplified themes are in Table 7.2.

Table 7.2. Simplified relationship themes derived from the clusters shown in Figures 7.1, 7.2, 7.3, and 7.4. A symbol is bolded if it refers to a theme that appears in multiple dendrograms. Complete descriptions of the individual clusters can be found in Appendix A.

Type	Symbol	Theme	Relevant Dendrogram	Supporting Cluster(s)
Chemical-Gene	A+	agonism, activation	Figure 7.1	6+
	A-	antagonism, blocking		6-
	B	binding, ligand (esp. receptors)		14-16
	E+	increases expression/production		8+,9+
	E-	decreases expression/production		8-,9-,10
	E	affects expression/production (neutral)		8,9,11a
	N	inhibits		3
Gene-Chemical	O	transport, channels	Figure 7.1	19,21
	K	metabolism, pharmacokinetics		11c
	Z	enzyme activity		20
Chemical-Disease	T	treatment/therapy (incl. investigatory)	Figure 7.2	8g,8h,9
	C	inhibits cell growth (esp. cancers)		2,3
	Sa	side effect/adverse event		6,15,16
	Pr	prevents, suppresses		1,9,21,24,28
	Pa	alleviates, reduces		26,30
	J	role in pathogenesis		20
Disease-Chemical	Mp	biomarkers (progression)	Figure 7.2	18,19
Gene-Disease	U	causal mutations	Figure 7.3	14
	Ud	mutations affect disease course		13
	D	drug targets		10,12
	J	role in pathogenesis		2h,4,6,8,9
	Te	possible therapeutic effect		2j,3
	Y	polymorphisms increase risk		22,26,27
	G	promotes progression		29
Disease-Gene	Md	biomarkers (diagnostic)	Figure 7.3	5,7
	X	overexpression in disease		15,17,30
	L	improper regulation linked to disease		18,19,21
Gene-Gene	**B**	binding, ligand (esp. receptors)	Figure 7.4	10
	W	enhances response		13
	V+	activates, stimulates		14,16
	E+	increases expression/production		21,22
	E	affects expression/production (neutral)		7,17
	I	signaling pathway		24
	H	same protein or complex		25
	Rg	regulation		28,30
	Q	production by cell population		1,2,6

The first major cluster, cluster 3, refers to inhibition (the chemical, C, is an inhibitor of the protein, G). This is mostly reported in a static context in patterns such as "C, a G inhibitor" and "G inhibition by C". The mechanism behind the inhibition is usually unclear from these descriptions - is C inhibiting the activity of the protein G or the expression of G's mRNA? It's difficult to tell.

Clusters 5 and 6 specifically describe effects on protein activity, with 6, the larger cluster, referring mainly to situations where C is an agonist or antagonist of G. Antagonists are often referred to as "blockers" or "inhibitors", while agonists are referred to as "activators" or "ligands".

Clusters 8, 9, 10 and 11a all describe effects on mRNA and protein *levels*, rather than protein activity. Cluster 10 specifically refers to inhibition, while the effects in clusters 8 and 9 were mixed: some positive, some negative, some neutral. Cluster 11a sometimes refers to a treatment response, as though C is being administered in a therapeutic manner or G's response to C is being specifically investigated.

Clusters 14-16 all describe the binding of C to a protein, G, which is usually a receptor for C. Many of the chemicals connected to proteins via these dependency paths are endogenous compounds, such as amino acids and hormones.

Clusters 11c and 19-21 reverse the directionality of the relationship. Until this point, most of the relationships we have described involve situations where the chemical, C, acts on the protein, G, perhaps by inhibiting it, inducing its activity, or raising or lowering its expression/synthesis. But there are also relationship classes where the protein acts on the chemical: enzymes that modify chemical structures, transporters that shuttle chemicals across cell membranes, and a variety of other pharmacokinetic relationships.

Cluster 11c contains most of the pharmacokinetic relationships, including the effect of G on C's metabolism and situations where C is actually a metabolite produced by G after acting on some other chemical. Some transport relationships, which are also pharmacokinetic relationships but are described more specifically in clusters 19 and 21, are also found here; note that pharmacokinetic relationships are a superset of these. Cluster 20, the last of the clusters assigned a theme, refers to enzymatic modification of a chemical, usually by an enzyme that specifically targets that chemical and thus

has the chemical's name as part of its own name.

While cluster 11c contained some fine-grained local structure – paths specifically referring to metabolism or secretion tended to cluster close together in the dendrogram, for example – it was surprisingly difficult to distinguish different classes of pharmacokinetic relationships from within this cluster. We comment on this issue further in the discussion.

We did not assign themes to the last major group of clusters in the dendrogram (clusters $23 - 30$) because these mainly reflected a major error where part of a protein (an amino acid or a particular binding domain such as a zinc finger) was misidentified as a chemical. While amino acids and elements such as zinc are chemicals, the relationships reflected here are whole-part, not interactions between distinct entities.

7.3.2 Chemical-Disease Relationships

Figure 7.2 shows the hierarchical clustering of the chemical-disease dependency paths. Full descriptions of the labeled clusters can be found in Table A.3 and simplified themes are in Table 7.2.

By far the largest set of chemical-disease relationships (from clusters 8g, 8h and 9) are treatment relationships, in which a chemical is described as a treatment or potential treatment for a disease. Similar to cluster 3 from Figure 7.1, these relationships are mostly described in a static context: we don't know why C is a useful treatment for D, but it is described as such without further elaboration. While we did not choose to differentiate clusters 8g, 8h and 9 in terms of their relationship theme, there are subtle differences among these three clusters. Cluster 8g mostly refers to evaluation of efficacy, where C is being investigated as an experimental agent for treating D, or patients are described as receiving C for D without an indication as to whether C is useful. Dependency paths in cluster 8h tend to go further, indicating that the treatment was efficacious for D. Finally, cluster 9, a small cluster with only 14 dependency paths, includes statements about using C to prevent or reduce D, which is slightly different than treating D. However, due to the substantial similarities of paths among these three clusters (some variant of the phrase "treatment for" appears in all three), we

labeled all of them with the same theme.

The word "treat" (treated, treatment, etc.) also appears quite often in clusters 3 and 6, but its meaning here is different. Cluster 3 refers not to the treatment of patients, but the treatment of cell lines, usually cancer cell lines. Clusters 2 and 3 reflect a theme where a chemical reduces the proliferation of cells exhibiting a particular disease phenotype.

Cluster 6, which also involves the word "treat", refers mainly to the evaluation of side effects in C-treated patients. Despite its proximity to clusters 8 and 9 in the dendrogram, it is semantically more related to clusters 15 and 16, which describe side effects. In these clusters, D is not a disease that C is used to treat, but a side effect or adverse event resulting from treatment with C.

Cluster 20, which is close in meaning to clusters 15 and 16, includes statements implicating C in the pathogenesis of D. Here C is most often an endogenous compound. Whereas in clusters 15 and 16 we tend to see situations where a drug is intentionally administered to a patient or animal, causing an adverse event, cluster 20 refers to cases where levels of C (most often in serum or tissue) are associated with the risk or progression of D. These levels may result from external supplementation or overproduction of an endogenous compound by the body.

Related to cluster 20 are clusters 18 and 19, which describe biomarkers. In these situations, the chemical C is not implicated in the pathogenesis of D, but is instead referred to as an indicator, or marker, of disease progression. There is considerable overlap with the patterns used in cluster 20, but again the shift in meaning is subtle - a substance can be an *indicator* of D without *causing* D.

Finally, several clusters are closely related to the idea of disease treatment, but rather than stating, simply, "C is a treatment for D", they indicate observations about what C, exactly, is doing. Clusters 1, 9, 21, 24, and 28 all refer to situations where C prevents D, or reduces the risk of D (note that cluster 9 appears both in the "prevents" theme, Pr, and in the "treatment/therapy" theme, T, in Table 7.2). In contrast, clusters 26 and 30 refer to cases where C alleviates D, or reduces its effect. The implication here is that C is being used after D has already occurred.

7.3.3 Gene-Disease Relationships

Figure 7.3 shows the hierarchical clustering of the gene-disease dependency paths. Full descriptions of the labeled clusters can be found in Table A.4 and simplified themes are in Table 7.2.

Cluster 20 in Figure 7.2 contains relationships that are quite similar in character to those in clusters 2h, 4, 6, 8, and 9 in Figure 7.3. All of these clusters describe situations where a protein (or chemical, in Figure 7.2 cluster 20) is implicated in the pathogenesis of a disease. Clusters 4 and 6 refer simply to increased levels of G in D, whereas clusters 8 and 9 more directly implicate the protein in the pathogenesis of the disease. Cluster 29 reflects a slightly different theme, where the protein promotes disease progression, rather than disease onset. The two themes share some overlap but are subtly different; cluster 29 focuses on cancers, discussing proteins promoting cell invasion, proliferation, and progression.

Clusters 5 and 7 in Figure 7.3 are similar to clusters 18 and 19 in Figure 7.2 in that they do not ascribe a pathogenic role to the protein (or chemical) but instead refer to it as a biomarker. Cluster 7 contains statements where a protein, G, is described as "a robust diagnostic biomarker for D", or "an indicator of D", without insinuating that it causes D. Cluster 5 is very closely tied to cluster 6, but cluster 6 contains a few statements with causal implications, such as "G is a mediator of D".

Clusters 2j and 3 include therapeutic relationships, where G is described as a treatment or potential treatment of D. Cluster 3 mostly describes trials of G in the treatment of D. While there are a few statements that could perhaps imply efficacy, such as "G therapy for patients with D", the treatment relationships here are not described with anywhere near the definiteness of clusters 8 and 9 in Figure 7.2.

In clusters 10 and 12, the protein, G, is described as a drug target or potential target for the treatment of the disease, D. Often this description does not include the word "target", but it is implied - the statement refers to the utility of G inhibitors in treating D, for example.

Some statements in clusters 10 and 12 refer to mutations in G that have an effect on D. It's implied that disruptions in the activity of G can impact the course of D. Clusters 13 and 14 address the issue of mutations more directly, either by describing

studies that investigate the role of G mutations in the progression of D (cluster 13) or by directly implicating mutations in G as causal risk factors in D (cluster 14).

While clusters 5 and 7 refer specifically to biomarkers, clusters 15, 17 and 30 refer to overexpression of proteins in disease, usually in patient serum. These proteins could represent potential biomarkers as well, although they are not described in that way.

Clusters 18, 19 and 21 focus on regulation, specifically cases where improper regulation of a gene is linked to disease. There is substantial overlap between these ideas and those of overexpression, biomarkers, etc. but again the focus is subtly different.

The last set of clusters, 22, 26 and 27, focus explicitly on polymorphisms that increase disease risk. The terms "polymorphism", "mutation", and "variant" are all present. Cluster 22 focuses almost exclusively on tumor suppressor genes, which, when mutated, can cause cancers. Note that in this case it is mutations in the gene (the DNA) that are increasing risk, rather than the level or activity of a protein. There is some semantic overlap with clusters 13 and 14.

7.3.4 Gene-Gene Relationships

Figure 7.4 shows the hierarchical clustering of the gene-gene dependency paths. Full descriptions of the labeled clusters can be found in Table A.5 and simplified themes are in Table 7.2.

The cluster themes in Figure 7.4 were the most difficult to parse among all the dendrograms. The vast majority of protein-protein relationships reflect some kind of change in activity or expression in the second protein based on the action of the first protein. Many of the relationships are similar to chemical-gene relationships in that a protein binds to another protein (cluster 10), increases its expression (clusters 21 and 22), or affects its expression in some other way that is not stated (clusters 7 and 17). All of these themes also appear in Figure 7.1.

However, there are a few other themes that are specific to protein-protein interactions. One protein can enhance the response of another to some stimulus (cluster 13), or activate or stimulate another protein by itself (clusters 14 and 16). A protein

can be produced by a cell population expressing another protein, as in the case of lymphocytes (i.e. proteins produced by CD4-bearing T-cells), which is reflected in clusters 1, 2 and 6.

Clusters 24, 25, 28 and 30 all reflect similar relationships involving regulation and pathways, but are subtly different. Cluster 24 explicitly refers to signaling, with both protein members forming part of the same signaling pathway. Cluster 25 is a cluster of patterns reflecting abbreviations, where the two proteins involved are literally identical or part of the same protein complex. Clusters 28 and 30 speak more specifically of regulation, but contain several patterns that also refer to co-membership in the same pathway. All of these concepts are related.

7.4 Creating a Global Relationship Network

7.4.1 Assigning Remaining Paths to Themes

The themes in Table 7.2 are based on only the most frequent $\approx$ 700 connecting dependency paths for each combination of entity types. From here on, we will call the paths represented in Table 7.2 the *flagship paths* for each theme. However, from the full datasets described in Table 7.1, we can see that there are vastly more dependency paths than this in the full dataset. The distributions of dependency path frequencies are Zipfian, meaning that there are a large number of paths that occur only once or twice. We would like to make intelligent guesses about the theme memberships of these remaining paths.

To obtain an estimate of how much each path supports each theme, we counted the number of times each path cooccurred with the flagship paths in Figures 7.1, 7.2, 7.3, and 7.4. We define a co-occurrence as a situation where both the unassigned path and a flagship path connect the same entity pair. We calculated cooccurrence frequencies for the flagship paths as well as the non-flagship paths.

We refer to the number of cooccurrences of each path with flagship paths for a particular theme as that path's *support* for that theme. For example, one of the

dependency paths[2] corresponding to the pattern "the G agonist C", co-occurs with paths from the B theme (Table 7.2) 18 times. It also co-occurs with flagship paths from the A+ theme (agonists/activators) 18 times, but never co-occurs with a path from any other theme. In this way, the same path can support both theme B and theme A+. And in fact, an agonist does both bind to and activate a protein. In general, although we have assigned different symbols to the different themes in Table 7.2, the semantics of these themes are not mutually exclusive.

In the end, we were able to assign theme distributions to 37,491 chemical-gene dependency paths (13.6% of total), 2,021,192 chemical-disease dependency paths (33.3%), 136,206 gene-disease dependency paths (20.0%), and 41,418 gene-gene dependency paths (11.0%) to themes. The rest of the dependency paths never co-occurred with a single flagship path for any theme, so we could not make a call as to their meaning.

7.4.2 Description of the Final Network

Our final dataset contains two parts. Part I of the dataset contains the support of each dependency path toward each theme. Each record contains a [lowercased] dependency path, followed by a set of columns that is 2× as long as the number of themes for that relationship type. For example, the record for the Chemical-Gene path `trial|appos|START_ENTITY trial|nmod|inhibitor inhibitor|amod|END_ENTITY` contains 21 fields (10 possible themes; see Table 7.2). The record contains the dependency path, followed by the supports (co-occurrence frequencies with flagship paths) for each theme, and an indicator of whether or not the path itself is part of the flagship path sets for that theme. This particular path is not a flagship path for any theme; it has a support of 3 for theme N (inhibition) and zero support for any other theme.

Part II consists of four sets of dependency paths, along with the original sentences from which they derive and associated metadata. A single record contains the following information:

[2] [*nummod*, agonist, *nummod*]

15161679	PubMed ID
0	Sentence number (0 = title)
zosuquidar_trihydrochloride	First entity name, formatted
54,81	First entity name, location in abstract
P-glycoprotein	Second entity name, formatted
28,42	Second entity name, location in abstract
zosuquidar trihydrochloride	First entity name, raw string
P-glycoprotein	Second entity name, raw string
MESH:C095179	First entity, database identifier(s)
5243	Second entity, database identifier(s)
Chemical	First entity, type
Gene	Second entity, type
trial|appos|START_ENTITY trial|nmod|inhibitor inhibitor|amod|END_ENTITY	Dependency path
A Phase I trial of a potent P-glycoprotein inhibitor , zosuquidar_trihydrochloride -LRB- LY335979 -RRB- , administered intravenously in combination with doxorubicin in patients with advanced malignancy .	Sentence, tokenized

Part II contains 4,451,661 records, of which 92,465 (2.1%) represent chemical-gene dependency paths, 3,875,209 (87.1%, i.e. the vast majority) are chemical-disease paths, 338,306 (7.6%) are gene-disease paths, and 145,681 (3.3%) are gene-gene paths. We have arranged the paths in alphabetical order of the entity pairs, so that different sentences referring to the same two entities appear next to each other in the file.

7.5 Summary, Limitations & Future Work

We have shown how EBC can be used in conjunction with hierarchical clustering to uncover semantically-related sets of dependency paths, and developed an approach that produces thematic labels for dependency paths both within and outside of these clusters. One important property of our approach is that a single dependency path can provide support for multiple themes, so the themes can be reconfigured, and new themes can be introduced at any time, without altering the support for existing themes. The dependency paths for different themes can also overlap each other.

7.5.1 The Limits of Co-occurrence

One downside of this approach, and of EBC more generally, is that it relies on the co-occurrence of different dependency paths within entity pairs to establish the meaning of rarer paths. But there are a large number of dependency paths that (a) never co-occur with another path, and (b) occur with only one entity pair. These orphan paths are impossible to assign to themes using the current method. Out of 556,487 chemical-gene dependency path connections in the literature, we are currently able to assign theme supports to only 92,465 (16.6%). For chemical-disease connections, of which there are 13,658,821 in Medline, we can assign themes to only 3,875,209 (28.4%). For gene-disease connections, we can assign themes to 338,306 out of 1,071,043 (31.6%), and for gene-gene, we can assign themes to 145,681 out of 1,274,010 (11.4%).

Depending on the application, it is possible that the missing paths do not matter. After all, we are capturing the paths and entity pairs that have the most support in the literature. However, this does imply that if we need to capture these rarer paths, we need to have some way of connecting them to the more frequent dependency paths that does not rely on co-occurrence within entity pairs.

7.5.2 Common Sources of Error

We have observed several sources of error that can lead to incorrect or misleading theme assignments. We plan to improve these iteratively in future releases of this network.

One common source of error arises from the use of dependency paths as features. A dependency path can only capture the relationship between two entities in a sentence, but many relationships involve more than two entities. An example would be a situation where the levels of a chemical in a cell culture (C1) affect the ability of a drug (C2) to exert its effect on a particular receptor (G). Thus, we may choose to flag sentences where > 2 entities (and thus, dependency paths) are present and analyze these separately.

An issue particular to gene-gene relationships, or any type of symmetric relationship, is that our method treats each direction separately. We were interested to see

whether the dendrogram in Figure 7.4 would fragment into two halves, each containing relationships of a particular directionality, but this did not occur. Many of the gene-gene relationships in Table 7.2 are symmetric (binding, for example), but at this time, we are unable to distinguish directionality in, for example, activation relationships. All we can say is that one of the proteins increases (or decreases) the activity of the other protein, which is somewhat unsatisfying.

The named entity recognition provided by PubTator, while state-of-the-art, is also not perfect. While the multi-word entity recognition provided by PubTator is a huge improvement over our earlier method (see Chapter 6) and captures many more entities, we have also observed several situations where only parts of entity names are captured, or where entities are assigned to the wrong type (proteins labeled as chemicals, etc.). As long as we continue to rely on PubTator for NER, this will be an issue, but we expect that NCBI will continue to refine their algorithms as we refine ours.

7.5.3 On Evaluation and Applications

In building this network, we have created a resource that we hope will prove useful for biomedical scientists in a variety of disciplines. Each edge in this network represents one discovery, made by some scientist in a particular time and place. By combining the discoveries of thousands of scientists, we can potentially predict new drug-drug interactions, uncover pathways for new drugs, break down complex biochemical interactions mechanistically, and begin to understand the genetic and chemical similarities underlying complex phenotypes.

It is impossible to evaluate the quality of the network itself without specifying an application. For example, we have chosen not to obsess over the gene-gene bidirectionality "error" at this point because we don't know whether fixing it would have any impact. In the next two chapters, we present two applications of the network to real biomedical problems: curating pharmacogenomic pathways, and predicting (and explaining) drug-drug interactions.

Chapter 8

Building Pharmacogenomic Pathways

Here we describe how the global relationship network from Chapter 7 can be applied to learn pharmacogenomic pathways from the raw text of the biomedical literature, a task that is currently performed by human curators at PharmGKB [62]. We discuss the benefits and limitations of this approach and provide examples of how text mining can be used to expand existing pathways and discover new ones.

8.1 Pharmacogenomic Pathways

8.1.1 Pharmacokinetic and Pharmacodynamic Pathways

A pharmacogenomic pathway contains the complete set of biochemical reactions, transport and catalysis events that happen to a drug once it enters the body.

A *pharmacokinetic (PK)* pathway is the subset of these events through which a drug is absorbed, distributed, metabolized, and excreted. In other words, it represents what the body does to the drug. Examples of PK pathway components include metabolic reactions in which enzymes alter the chemical structure of the drug or break it down into multiple smaller components. They also include the binding and transport events by which a drug is transferred out of cells and eventually, excreted

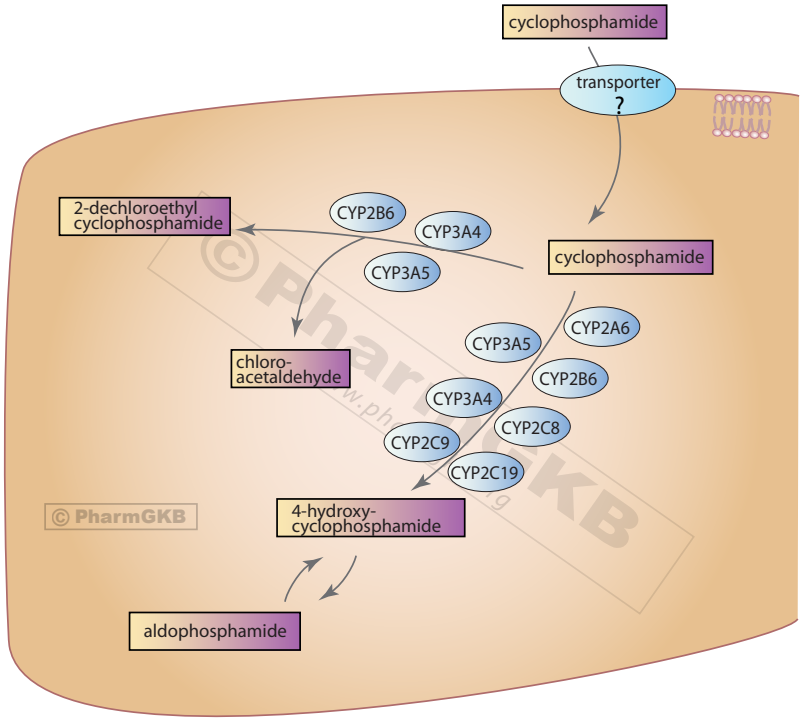


Figure 8.1. The pharmacokinetic pathway for cyclophosphamide, an anti-cancer agent. Copyright PharmGKB and Stanford University (2016). Reprinted with permission.

from the body.

A *pharmacodynamic (PD)* pathway, in contrast, is the set of events through which the drug acts on the body. The drug may, perhaps, bind to a receptor, either on a cell within the body or on a microorganism (bacteria, virus, fungus, etc.) or parasite within the body. It may be transported across the nuclear membrane into the nucleus or it may bind to proteins or other molecules within the cytoplasm. These are just a few of the possible types of events that constitute the pharmacodynamic pathway.

Pathways consist of much more than just chemical-protein or protein-protein interactions, but these form two important components of pathways. For example, Figure 8.1 shows the pharmacokinetic pathway for cyclophosphamide, an important anticancer compound. In the figure, the liver cytochrome CYP2B6 is depicted as part of the initial step in the metabolism (PK pathway) of cyclophosphamide. Once cyclophosphamide enters a liver cell through an unknown transport mechanism,

CYP2B6 and other liver cytochromes work together to convert it to 4-hydroxy-cyclophosphamide. The interaction of CYP2B6 with the cyclophosphamide molecule represents an important step in its metabolism, and it constitutes one type of drug-protein interaction that we would like to uncover automatically through text mining.

Protein-protein interactions represent another important component of pharmacogenomic pathways. For example, Figure 8.2 shows the pharmacodynamic pathway for selective serotonin reuptake inhibitors (SSRIs), an important class of drugs widely used in the treatment of depression. The two cells shown are neurons, one of which is releasing the neurotransmitter serotonin (5HT) into the synaptic cleft (the space between neurons, though which they communicate using neurotransmitters). The serotonin molecules then bind to a set of receptors on the postsynaptic neuron - for example, the HTR1 receptor. Once the serotonin molecule binds to HTR1, it activates a signaling cascade that begins with coupling of the internal part of the HTR1 receptor and the Gi/o protein alpha subunit (GNAI) [126]. This is a direct protein-protein interaction that is important for serotonin's effect on neurons, and in turn, SSRIs' clinical effects on humans.

8.1.2 The PharmGKB Pathways

The PharmGKB pathways database[1] [62] contains, as of this writing, 108 unique pharmacodynamic and pharmacokinetic pathways for drugs or drug classes. Pictures of two of these pathways are shown in Figures 8.1 and 8.2.

The PharmGKB pathways are constructed manually by PharmGKB's curators and are limited to drugs with relevant (or potential) pharmacogenetic (PGx) associations[2]. The drugs for which pathways are built are chosen after careful review of the FDA's biomarker list[3] and the Clinical Pharmacogenetics Implementation Consortium (CPIC) nominations[4]. Pathway interactions are supported by manually curated evidence from the biomedical literature, and relevant PubMed IDs are provided for each pathway

[1] https://www.pharmgkb.org/view/pathways.do

[2] As mentioned in Chapter 6, PGx associations are situations where mutations in the gene are known to affect a patient's clinical response to a drug.

[3] http://www.fda.gov/Drugs/ScienceResearch/ResearchAreas/Pharmacogenetics/ucm083378.htm

[4] https://www.pharmgkb.org/page/cpic

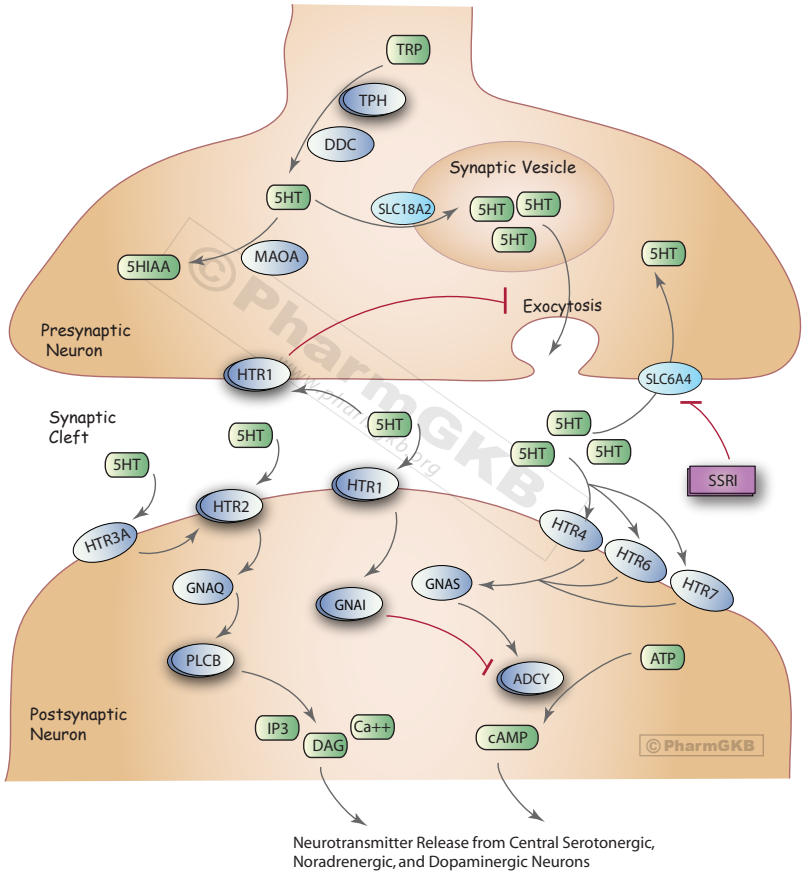


Figure 8.2. The pharmacodynamic pathway for selective serotonin reuptake inhibitors (SSRIs), which are used in the treatment of depression. SSRIs work by blocking reuptake of serotonin (5HT) from the synapse by the SLC6A4 transporter, which is shown in the diagram by the red blocking symbol between the SSRI molecule and the SLC6A4 transporter, which lives in the presynaptic cell's membrane. Copyright PharmGKB and Stanford University (2016). Reprinted with permission.

component.

8.2 Identifying Pathway Candidates

The global relationship network from Chapter 7 contains labeled relationships among a large number of chemicals and genes. Our hypothesis is that certain types of relationship patterns among chemicals, genes, and phenotypes in the network will occur preferentially between chemicals and genes that are connected within a pathway. If this is true, we want to learn to recognize these patterns so we can begin building pathways for new drugs.

8.2.1 Binarizing Pathway Interactions

A pathway file contains several lines of information, each of which includes the following fields: *From, To, Reaction Type, Controller, Control Type, Cell Type, PubMed Id, Genes, Drugs,* and *Diseases*. The *Reaction Type* can be one of the following: *Biochemical Reaction, Complex Assembly, Conversion,* or *Transport*. Each line in a pathway file represents a single biochemical event, but can include more than two entities.

Although multiple entities can be present in each event, the network from Chapter 7 contains edges that connect pairs of entities. To extract the relevant pairwise interactions among drugs, chemicals and genes in each pathway, we applied the following steps to each line in a pathway file:

1. *Preprocessing.* We preprocessed the line to remove extra whitespace and quotes, which surrounded some of the entity names. We split fields containing multiple entity names so we could handle each name separately.
2. *Extracted the Genes, Drugs, From, To, and Controllers fields.* The From and To fields were most likely drugs, metabolites, or chemicals, while the Controllers could be drugs/chemicals or proteins. We removed all recognized drug and gene names from the From, To and Controllers fields and added the rest to both the chemical list and the gene list.

3. *Converted multiple entities to sets of pairs.* We took all binary combinations of chemical and gene entities and created a separate record for each.
4. *Obtained synonyms from PharmGKB lexicons, and restricted to strings in network.* Using the drug and gene lexicons from PharmGKB, we added synonyms for each drug or gene name if any were available, keeping only those strings that were present in the network[5].

The end result was a list of chemical-gene and gene-gene interactions for each of the 108 pathways. Each interaction was labeled with its type, its unique identifier (since multiple pairs of strings could correspond to the same interaction), and (if it was a chemical-gene interaction) whether or not the chemical in question was a recognized drug or another type of chemical.

For example, here are some interactions for the gemcitabine pathway, which includes both pharmacokinetics and pharmacodynamics:

entity 1	type 1	entity 2	type 2	interaction type	id	is drug
gemcitabine	Chemical	cda	Gene	BiochemicalReaction	0	1
gemcitabine	Chemical	cytidine_deaminase	Gene	BiochemicalReaction	0	1
gemcitabine	Chemical	dnt1	Gene	Conversion	7	1
rrm2b	Gene	rrm2	Gene	Conversion	9	NA
cnt1	Gene	ent2	Gene	Transport	24	NA
cnt3	Gene	ent1	Gene	Transport	25	NA

8.2.2 A Strategy for Building Pathways

Pathways contain both nodes (biomedical entities) and edges (relationships). To build a pathway for a new drug, we need to be able to identify both. To identify likely pathway nodes, we begin at the drug for which the pathway is being built (the *origin*) and traverse the network radially to a fixed distance. Based on the patterns of labeled edges connecting the origin to its neighbors, we build a classifier to distinguish likely pathway candidates from other chemicals and genes. Once the likely pathway nodes are identified, we use the network to identify relationships among all of the pathway

[5]We ignored entities not found in the network because it is *a priori* impossible to identify pathway entities using text mining if they are not in the network at all. Even a perfect text mining algorithm could not find these. This problem will be addressed naturally in the future as more and more new discoveries are reported, and as we move beyond abstracts to full text articles.

entities and label them with their most likely interaction types.

Because each pathway has a different character (pharmacokinetic vs. pharmacodynamic, for example), we do not necessarily expect the patterns connecting the origin to other pathway chemicals and genes to be the same for all pathways. We therefore build separate classifiers for each of the 103 pathways for which at least one origin drug was listed and present in the network.

This strategy reflects the practical use case for these classifiers. If I want to build a pharmacokinetic pathway for an anticancer drug, I might expect to choose different candidates than if my goal is to build a pharmacodynamic pathway for an antidepressant. I might therefore choose to use the classifier for pathway PA2001 (irinotecan pharmacokinetics) to predict new candidates for my novel agent, or some combination of classifiers relevant to cancer and pharmacokinetics. Any increased ability to narrow down the pool of possible pathway candidates is worth it, since there are potentially tens of thousands of new pathway candidates within a certain radius of the origin drug in the network.

8.2.3 Pathway Representation in the Network

Before we begin building our classifiers, it is important to establish a baseline for performance based on the representation of pathway nodes and edges in the Chapter 7 network. We need to know how far out to search from the origin, as well as the fraction of pathway edges that are likely to appear in the network.

Table 8.1 shows the fraction of pathway nodes connected to the origin within an edge radius of 1, 2, or 3. Nearly all chemical and gene pathway entities in the network are reachable within a radius of 3, but then again, so are most of the chemicals and genes in the network at large. At least half of all pathway genes are reachable from the origin at a radius of 2 for 98/103 pathways (95.1%). We therefore elected to build our classifiers for pathway entities by considering only those entities at a radius of ≤ 2 from the origin.

Table 8.2 shows the fraction of pathway edges directly connected within the

Table 8.1. Pathway nodes connected to the origin drug in the network at a radius of 1, 2, and 3 edges from the origin, for all 103 pathways. The pathways are sorted in order of the frequency of pathway gene nodes that occur within a radius of 2 from the origin, from greatest to smallest. PD = pharmacodynamics, PK = pharmacokinetics. Note that a radius of 1 is impossible for chemicals, because there are no direct chemical-chemical interaction edges in the network.

	Genes				Chemicals			
Pathway	Total	r1	r2	r3	Total	r1	r2	r3
PA145011113-Warfarin_PK	8	0.12	1.00	1.00	0	-	-	-
PA145011120-Aromatase_Inhibitor_Breast_Cell_PD	6	0.17	1.00	1.00	1	0.00	1.00	1.00
PA152325160-Gefitinib_PK	9	0.11	1.00	1.00	0	-	-	-
PA152530846-Proton_Pump_Inhibitor_PK	1	0.00	1.00	1.00	0	-	-	-
PA154426155-Taxane_PK	13	0.31	1.00	1.00	0	-	-	-
PA161749012-Fluoxetine_PK	6	0.50	1.00	1.00	0	-	-	-
PA162359940-Imipramine_Desipramine_PK	4	0.25	1.00	1.00	0	-	-	-
PA164713429-Citalopram_PK	5	0.20	1.00	1.00	0	-	-	-
PA165291507-Fluoropyrimidine_PD	1	0.00	1.00	1.00	4	0.00	1.00	1.00
PA165816736-Celecoxib_PK	7	0.00	1.00	1.00	0	-	-	-
PA165816969-Rosiglitazone_PK	3	0.33	1.00	1.00	0	-	-	-
PA165884757-Caffeine_PK	7	0.14	1.00	1.00	1	0.00	1.00	1.00
PA165980774-Uric_Acid_Lowering_Drugs_PD	5	0.80	1.00	1.00	4	0.00	1.00	1.00
PA165986114-Tacrolimus_Cyclosporine_PK	3	1.00	1.00	1.00	0	-	-	-
PA166014758-Venlafaxine_PK	5	0.00	1.00	1.00	0	-	-	-
PA166121347-Paroxetine_PK	7	0.14	1.00	1.00	0	-	-	-
PA2034-Cyclophosphamide_PK	7	0.14	1.00	1.00	2	0.00	1.00	1.00
PA165292163-Doxorubicin_Cancer_Cell_PD	21	0.19	0.95	1.00	0	-	-	-
PA2042-Sympathetic_Nerve_Neuroeffector_Junction	21	0.38	0.95	1.00	0	-	-	-
PA153627758-Potassium_Channel_Inhibitors_PD	36	0.22	0.94	0.97	1	0.00	1.00	1.00
PA2025-Etoposide_PK_PD	18	0.22	0.94	1.00	3	0.00	1.00	1.00
PA166121942-Ibuprofen_PD	34	0.24	0.94	1.00	8	0.00	1.00	1.00
PA2023-ACE_Inhibitor_PD	31	0.29	0.94	1.00	2	0.00	1.00	1.00
PA165110622-Renin_Angiotensin_Agents_PD	30	0.30	0.93	0.97	2	0.00	1.00	1.00
PA2039-Methotrexate_Cancer_Cell_PD	14	0.29	0.93	1.00	10	0.00	1.00	1.00
PA150981002-Vinka_Alkaloid_PK	13	0.00	0.92	0.92	0	-	-	-
PA165291575-Antimetabolite_Folate_Cycle_PD	13	0.31	0.92	1.00	9	0.00	1.00	1.00
PA165980399-Oxidative_Stress_Reg_Erythrocyte	13	0.54	0.92	1.00	6	0.00	1.00	1.00
PA165959584-Sorafenib_PD	40	0.12	0.90	0.95	0	-	-	-
PA145011109-Atorvastatin_Lovastatin_Simvastatin_PK	19	0.26	0.89	0.95	0	-	-	-
PA165817070-Carbamazepine_PK	19	0.21	0.89	1.00	0	-	-	-
PA145011114-Warfarin_PD	18	0.06	0.89	0.94	0	-	-	-
PA164713427-Imatinib_PK_PD	16	0.31	0.88	0.94	0	-	-	-
PA145011110-Pravastatin_PK	15	0.13	0.87	1.00	0	-	-	-
PA165378192-Artemisinin_and_Derivatives_PK	7	0.00	0.86	1.00	0	-	-	-
PA165948259-Metformin_PK	7	0.29	0.86	1.00	0	-	-	-
PA2037-Ifosfamide_PK	7	0.00	0.86	1.00	1	0.00	1.00	1.00

PA2031-Statin_PD	34	0.12	0.85	1.00	7	0.00	1.00	1.00
PA154444041-Platelet_Aggregation_Inhibitor_PD	66	0.23	0.85	0.98	7	0.00	1.00	1.00
PA2030-Sympathetic_Nerve_Pre/Post_Ganglionic_Jct	13	0.62	0.85	0.92	0	-	-	-
PA166041114-Ibuprofen_PK	19	0.11	0.84	1.00	0	-	-	-
PA2032-VEGF_Signaling_Pathway	81	0.14	0.84	0.99	1	0.00	1.00	1.00
PA164713428-Losartan_PK	6	0.33	0.83	1.00	1	0.00	1.00	1.00
PA165292177-Doxorubicin_PK	30	0.17	0.83	0.97	0	-	-	-
PA146123006-Codeine_and_Morphine_PK	11	0.09	0.82	0.82	2	0.00	1.00	1.00
PA154426903-Erlotinib_PK	11	0.18	0.82	1.00	0	-	-	-
PA165292164-Doxorubicin_Cardiomyocyte_Cell_PD	22	0.18	0.82	0.95	0	-	-	-
PA165816270-Methotrexate_Brain_Cell_PK	11	0.09	0.82	1.00	0	-	-	-
PA165964265-Valproic_Acid_PK	16	0.12	0.81	1.00	0	-	-	-
PA162356267-EGFR_Inhibitor_PD	91	0.07	0.80	0.90	3	0.00	0.33	0.67
PA145011108-Statin_Generalized_PK	30	0.00	0.80	1.00	0	-	-	-
PA165947317-Leukotriene_modifiers_PD	10	0.10	0.80	1.00	5	0.00	1.00	1.00
PA165950411-Nevirapine_PK	5	0.00	0.80	1.00	0	-	-	-
PA165958541-Theophylline_PK	5	0.00	0.80	1.00	2	0.00	1.00	1.00
PA165985892-Tacrolimus_Cyclosporine_PD	57	0.02	0.79	1.00	0	-	-	-
PA162355621-Nicotine_Dopaminergic_Neuron_PD	52	0.19	0.79	0.92	1	0.00	1.00	1.00
PA145011118-Estrogen_Metabolism_Pathway	14	0.14	0.79	1.00	2	0.00	1.00	1.00
PA165986279-Acetaminophen_therapeutic_doses_PK	28	0.18	0.79	1.00	1	0.00	1.00	1.00
PA145011115-Phenytoin_PK	18	0.06	0.78	1.00	0	-	-	-
PA152241951-Celecoxib_PD	63	0.17	0.78	0.90	5	0.00	1.00	1.00
PA165984799-Diuretics_PD	36	0.11	0.78	0.94	1	0.00	1.00	1.00
PA152530845-Proton_Pump_Inhibitor_PD	44	0.11	0.77	0.93	5	0.00	0.80	0.80
PA153627759-Repaglinide_PK	4	0.25	0.75	1.00	0	-	-	-
PA154423659-Nateglinide_PK	4	0.25	0.75	1.00	0	-	-	-
PA154424674-Clopidogrel_PK	12	0.33	0.75	1.00	0	-	-	-
PA162355620-Nicotine_Chromaffin_Cell_PD	4	0.00	0.75	1.00	0	-	-	-
PA150653776-Fluoropyrimidine_PK	22	0.09	0.73	0.95	1	0.00	1.00	1.00
PA165964832-Mycophenolic_acid_PK_PD	11	0.09	0.73	1.00	1	0.00	1.00	1.00
PA2038-Ifosfamide_PD	18	0.06	0.72	0.94	4	0.00	1.00	1.00
PA161749006-SSRIs_PD	32	0.19	0.72	1.00	5	0.00	1.00	1.00
PA145011119-Tamoxifen_PK	21	0.19	0.71	0.95	0	-	-	-
PA165816349-Methotrexate_PK	21	0.10	0.71	1.00	0	-	-	-
PA165946349-Tramadol_PK	7	0.14	0.71	0.86	0	-	-	-
PA165959537-Sorafenib_PK	7	0.14	0.71	1.00	0	-	-	-
PA165971634-Pentose_Phosphate_Erythrocyte	7	0.29	0.71	0.86	6	0.00	1.00	1.00
PA165860384-Lamivudine_PK_PD	27	0.00	0.70	0.96	0	-	-	-
PA165948566-Metformin_PD	43	0.14	0.67	0.81	0	-	-	-
PA145011117-Aromatase_Inhibitor_PD	6	0.17	0.67	1.00	1	0.00	1.00	1.00
PA165111375-Benzodiazepine_PK	12	0.25	0.67	0.92	0	-	-	-
PA165960076-Clomipramine_PK	6	0.00	0.67	1.00	0	-	-	-
PA165981686-Doxepin_PK	6	0.00	0.67	1.00	0	-	-	-
PA165980050-Vemurafenib_PD	62	0.02	0.66	0.92	0	-	-	-
PA2029-Irinotecan_PD	17	0.12	0.65	1.00	1	0.00	1.00	1.00
PA165859361-Zidovudine_PK_PD	25	0.00	0.64	1.00	0	-	-	-

PA166114721-Uricosurics_PD	11	0.00	0.64	1.00	1	0.00	1.00	1.00
PA2001-Irinotecan_PK	19	0.21	0.63	1.00	1	0.00	1.00	1.00
PA145011111-Fluvastatin_PK	16	0.06	0.62	0.88	0	-	-	-
PA155028030-Tenofovir_Adefovir_PK	8	0.12	0.62	0.88	0	-	-	-
PA166117881-Acetaminophen_toxic_doses_PK	26	0.12	0.62	1.00	1	0.00	1.00	1.00
PA145011112-Rosuvastatin_PK	5	0.00	0.60	1.00	0	-	-	-
PA2024-Beta_agonist_Beta_blocker_PD	57	0.04	0.60	0.91	3	0.00	1.00	1.00
PA166123135-Efavirenz_PK_PD	17	0.00	0.59	0.88	0	-	-	-
PA2036-Gemcitabine_PK_PD	12	0.17	0.58	0.92	1	0.00	1.00	1.00
PA165374494-Busulfan_PD	14	0.00	0.57	0.93	3	0.00	0.67	0.67
PA165959313-Valproic_Acid_PD	14	0.07	0.57	0.79	4	0.00	1.00	1.00
PA166122732-Succinylcholine_PK_PD	16	0.00	0.56	0.94	1	0.00	1.00	1.00
PA2011-Nicotine_PK	11	0.09	0.55	1.00	0	-	-	-
PA2026-Glucocorticoid_HPA_Axis_PD	13	0.08	0.54	0.69	0	-	-	-
PA165980834-Methylene_Blue_PD	10	0.00	0.50	1.00	5	0.00	1.00	1.00
PA166115250-Gemtuzumab_ozogamicin_PK_PD	27	0.00	0.30	0.67	0	-	-	-
PA166104634-Abacavir_PK_PD	16	0.00	0.19	0.94	0	-	-	-
PA165815256-Amodiaquine_PK	4	0.00	0.00	0.75	0	-	-	-
PA166126086-PegIFN_alpha_2a/2b_Hepatocyte_PD	24	0.00	0.00	0.00	0	-	-	-

network[6]. A median 11.0% of chemical-gene pathway edges are connected in the network (range: 0 to 70.0%) and 16.0% of gene-gene pathway edges are found (range: 0 to 100%). More than half of chemical-gene pathway edges are present in the network for only 1.9% of pathways, and more than half of gene-gene pathway edges are present for 16.5% of pathways.

The only way to increase the representation of the pathway edges in the network is to increase the volume of text used in generating the network (in the hope that it will provide more edges) and/or to increase the number of dependency paths we are able to recognize and assign to themes. Until then, it is important to be conscious of the fact that we will almost certainly do better at recognizing pathway *entities* than pathway *edges*. The entities are at least present in the network and accessible from the origin, while most of the pathway edges are not represented directly in the network.

[6]We measured this using unique identifiers, not strings. So if two strings corresponded to the same chemical entity and only one was connected in the network to a gene, we would count that interaction as "found".

Table 8.2. Pathway interactions connected by network edges. Interactions of type chemical-gene and gene-gene are reported. The pathways are sorted in decreasing order of frequency of gene-gene interactions connected by network edges. PD = pharmacodynamics, PK = pharmacokinetics.

	Chemical-Gene			Gene-Gene		
Pathway	Found	Total	Fraction	Found	Total	Fraction
PA145011120-Aromatase_Inhibitor_Breast_Cell_PD	1	13	0.08	1	1	1.00
PA162359940-Imipramine_Desipramine_PK	1	8	0.12	3	3	1.00
PA165815256-Amodiaquine_PK	0	3	0.00	1	1	1.00
PA165960076-Clomipramine_PK	0	4	0.00	3	3	1.00
PA165986114-Tacrolimus_Cyclosporine_PK	15	22	0.68	10	10	1.00
PA166014758-Venlafaxine_PK	0	6	0.00	4	4	1.00
PA165981686-Doxepin_PK	0	5	0.00	5	6	0.83
PA166121347-Paroxetine_PK	1	6	0.17	5	6	0.83
PA161749012-Fluoxetine_PK	7	10	0.70	13	16	0.81
PA154426155-Taxane_PK	5	17	0.29	22	28	0.79
PA164713427-Imatinib_PK_PD	7	20	0.35	27	35	0.77
PA152325160-Gefitinib_PK	1	12	0.08	3	4	0.75
PA154426903-Erlotinib_PK	2	9	0.22	5	7	0.71
PA145011113-Warfarin_PK	3	15	0.20	7	10	0.70
PA164713429-Citalopram_PK	1	6	0.17	2	3	0.67
PA165111375-Benzodiazepine_PK	5	27	0.19	8	14	0.57
PA145011119-Tamoxifen_PK	10	27	0.37	26	51	0.51
PA150981002-Vinka_Alkaloid_PK	0	17	0.00	10	20	0.50
PA165816736-Celecoxib_PK	0	3	0.00	1	2	0.50
PA165950411-Nevirapine_PK	0	10	0.00	5	10	0.50
PA145011110-Pravastatin_PK	2	13	0.15	6	13	0.46
PA2025-Etoposide_PK_PD	3	22	0.14	4	9	0.44
PA166117881-Acetaminophen_toxic_doses_PK	1	19	0.05	13	30	0.43
PA150653776-Fluoropyrimidine_PK	2	20	0.10	3	7	0.43
PA165860384-Lamivudine_PK_PD	0	18	0.00	9	21	0.43
PA165817070-Carbamazepine_PK	3	10	0.30	14	34	0.41
PA165986279-Acetaminophen_therapeutic_doses_PK	3	29	0.10	14	34	0.41
PA165958541-Theophylline_PK	0	12	0.00	2	5	0.40
PA2001-Irinotecan_PK	6	38	0.16	6	15	0.40
PA145011115-Phenytoin_PK	1	2	0.50	23	64	0.36
PA145011118-Estrogen_Metabolism_Pathway	2	26	0.08	6	18	0.33
PA154424674-Clopidogrel_PK	2	5	0.40	6	18	0.33
PA165816349-Methotrexate_PK	3	20	0.15	6	18	0.33
PA2026-Glucocorticoid_HPA_Axis_PD	3	7	0.43	4	12	0.33
PA2034-Cyclophosphamide_PK	2	20	0.10	8	24	0.33
PA166115250-Gemtuzumab_ozogamicin_PK_PD	0	1	0.00	6	19	0.32
PA165374494-Busulfan_PD	0	12	0.00	3	10	0.30
PA146123006-Codeine_and_Morphine_PK	2	27	0.07	6	21	0.29
PA145011108-Statin_Generalized_PK	0	26	0.00	15	54	0.28

PA165884757-Caffeine_PK	4	22	0.18	6	22	0.27
PA145011109-Atorvastatin_Lovastatin_Simvastatin_PK	8	48	0.17	30	123	0.24
PA165292163-Doxorubicin_Cancer_Cell_PD	4	13	0.31	6	25	0.24
PA2037-Ifosfamide_PK	0	15	0.00	4	18	0.22
PA165859361-Zidovudine_PK_PD	0	21	0.00	6	33	0.18
PA166041114-Ibuprofen_PK	1	26	0.04	4	22	0.18
PA165959584-Sorafenib_PD	3	16	0.19	25	139	0.18
PA166121942-Ibuprofen_PD	8	47	0.17	3	18	0.17
PA165948566-Metformin_PD	1	22	0.05	33	202	0.16
PA165292177-Doxorubicin_PK	4	25	0.16	8	50	0.16
PA2029-Irinotecan_PD	2	31	0.06	4	25	0.16
PA145011111-Fluvastatin_PK	1	13	0.08	3	21	0.14
PA165292164-Doxorubicin_Cardiomyocyte_Cell_PD	0	24	0.00	2	15	0.13
PA162356267-EGFR_Inhibitor_PD	4	28	0.14	88	695	0.13
PA165816270-Methotrexate_Brain_Cell_PK	2	11	0.18	1	8	0.12
PA165110622-Renin_Angiotensin_Agents_PD	7	46	0.15	4	33	0.12
PA2023-ACE_Inhibitor_PD	6	43	0.14	4	33	0.12
PA165378192-Artemisinin_and_Derivatives_PK	0	18	0.00	2	17	0.12
PA2011-Nicotine_PK	4	17	0.24	2	17	0.12
PA165964832-Mycophenolic_acid_PK_PD	2	27	0.07	1	9	0.11
PA153627758-Potassium_Channel_Inhibitors_PD	1	24	0.04	11	100	0.11
PA2038-Ifosfamide_PD	1	13	0.08	2	19	0.11
PA165980050-Vemurafenib_PD	1	6	0.17	39	387	0.10
PA165985892-Tacrolimus_Cyclosporine_PD	0	2	0.00	112	1214	0.09
PA2032-VEGF_Signaling_Pathway	12	97	0.12	35	406	0.09
PA166126086-PegIFN_alpha_2a/2b_Hepatocyte_PD	0	3	0.00	8	108	0.07
PA2031-Statin_PD	9	35	0.26	2	27	0.07
PA166123135-Efavirenz_PK_PD	0	8	0.00	4	75	0.05
PA154444041-Platelet_Aggregation_Inhibitor_PD	12	68	0.18	9	268	0.03
PA166104634-Abacavir_PK_PD	0	5	0.00	1	31	0.03
PA152241951-Celecoxib_PD	8	62	0.13	1	115	0.01
PA162355621-Nicotine_Dopaminergic_Neuron_PD	13	56	0.23	3	405	0.01
PA152530845-Proton_Pump_Inhibitor_PD	3	29	0.10	1	145	0.01
PA145011112-Rosuvastatin_PK	0	5	0.00	0	3	0.00
PA145011114-Warfarin_PD	1	9	0.11	0	28	0.00
PA145011117-Aromatase_Inhibitor_PD	1	31	0.03	0	3	0.00
PA153627759-Repaglinide_PK	1	11	0.09	0	6	0.00
PA154423659-Nateglinide_PK	3	8	0.38	0	5	0.00
PA155028030-Tenofovir_Adefovir_PK	1	13	0.08	0	5	0.00
PA161749006-SSRIs_PD	16	38	0.42	0	61	0.00
PA162355620-Nicotine_Chromaffin_Cell_PD	0	2	0.00	0	5	0.00
PA164713428-Losartan_PK	4	13	0.31	0	5	0.00
PA165291575-Antimetabolite_Folate_Cycle_PD	4	47	0.09	0	3	0.00
PA165816969-Rosiglitazone_PK	3	7	0.43	0	3	0.00
PA165946349-Tramadol_PK	1	4	0.25	0	8	0.00
PA165947317-Leukotriene_modifiers_PD	2	26	0.08	0	4	0.00
PA165948259-Metformin_PK	4	9	0.44	0	2	0.00

PA165959313-Valproic_Acid_PD	1	25	0.04	0	6	0.00
PA165964265-Valproic_Acid_PK	1	20	0.05	0	34	0.00
PA165971634-Pentose_Phosphate_Erythrocyte	3	19	0.16	0	3	0.00
PA165980399-Oxidative_Stress_Reg_Erythrocyte	6	53	0.11	0	17	0.00
PA165980774-Uric_Acid_Lowering_Drugs_PD	7	18	0.39	0	1	0.00
PA165980834-Methylene_Blue_PD	1	20	0.05	0	6	0.00
PA165984799-Diuretics_PD	2	42	0.05	0	87	0.00
PA166122732-Succinylcholine_PK_PD	0	11	0.00	0	77	0.00
PA2024-Beta_agonist_Beta_blocker_PD	1	148	0.01	0	235	0.00
PA2030-Sympathetic_Nerve_Pre/Post_Ganglionic_Jct	11	30	0.37	0	22	0.00
PA2036-Gemcitabine_PK_PD	2	10	0.20	0	12	0.00
PA2039-Methotrexate_Cancer_Cell_PD	3	42	0.07	0	3	0.00
PA2042-Sympathetic_Nerve_Neuroeffector_Junction	7	49	0.14	0	20	0.00
PA152530846-Proton_Pump_Inhibitor_PK	0	1	0.00	0	0	-
PA165291507-Fluoropyrimidine_PD	1	5	0.20	0	0	-
PA165959537-Sorafenib_PK	1	8	0.12	0	0	-
PA166114721-Uricosurics_PD	8	41	0.20	0	0	-

8.2.4 Building Classifiers for Pathway Entities

For each of the 103 pathways for which at least one origin drug was listed and present in the network, we traversed the network radially from the origin and collected all chemical and gene entities within a radius of 2. If an entity was part of the pathway, we labeled it as a positive training example and if it was not, we labeled it as a negative training example. Since there were many more negative than positive training examples and we wished to build our classifiers on balanced training sets, we selected a random sample from the negative training set that was equal in size to the positive training set.

We used the network from Chapter 7 to extract all network paths of length ≤ 2 connecting the positive and negative training examples to origin, removing the names of any intervening entities. The theme (and theme combination) scores for these paths became features for a random forest classifier.

Obtaining theme scores for the various paths often required averaging the scores for multiple dependency paths. Our procedure for this was as follows:

- *Scores in parallel were averaged.* Say we had a chemical, C, and a gene, G, that were directly connected by two different dependency paths. The first path had a support of 4 for the "N" (inhibition) theme and a support of 2 for the "B"

(binding) theme (see Table 7.2). The second path had a support of 3 for the "N" theme. Then the pair (C, G) would have a score of 3.5 for the "N" theme and 2 for the "B" theme.

- *Scores in series were also averaged.* Say C and G were each connected by a single dependency path to an intermediate gene, G2. The dependency path connecting C to G2 had a support of 5 for the A- theme (antagonism/blocking) and 4 for the N theme (inhibition). The path connecting G to G2 had a support of 6 for the W (enhances response) theme. Then the length-2 trail between C and G would receive a score of $(5+6)/2 = 5.5$ for the "A-|W" theme combination and $(4+6)/2 = 5.0$ for the "N|W" combination. The identity of the intervening gene, G2, was not recorded.

For each of the 103 pathways, separately for chemical and gene entities, we built random forest classifiers using the theme scores as features, as long as the number of training examples was ≥ 20. We used the random forest implementation in Python's *scikit-learn* package and evaluated the performance of the classifiers using 10-fold cross-validation.

8.2.5 Classifier Performance

There were 79 pathways with sufficient training examples to build a classifier for either chemical or gene entities. Table 8.3 shows their 10-fold cross-validation average AUC, which varied substantially by pathway.

Although chemical classifiers could not be built for all pathways, their performance tended to be better (median: 0.79, range: 0.53-1.00) than that of the gene classifiers (median: 0.68, range: 0.33-0.92). There were a few pathways in particular for which classifier performance was high, including the leukotriene modifiers'[7] pharmacodynamic pathway, and the imatinib and methotrexate pharmacokinetic pathways.

[7]Leukotriene modifiers are a set of anti-asthmatic drugs, including zafirlukast and montelukast.

Table 8.3. Classifier performance at distinguishing pathway chemicals and genes from other chemicals and genes close to the origin, based on theme patterns in the network. Only the best 20 classifiers [at recognizing genes] are shown. PD = pharmacodynamics, PK = pharmacokinetics.

	Chemicals		Genes	
Pathway	Training Set Size	AUC	Training Set Size	AUC
PA165947317-Leukotriene_modifiers_pathway_PD	42	0.95	90	0.92
PA164713427-Imatinib_PK_PD	0	-	136	0.84
PA165816270-Methotrexate_Brain_Cell_PK	0	-	72	0.81
PA162359940-Imipramine_Desipramine_PK	0	-	34	0.80
PA2001-Irinotecan_PK	6	-	112	0.79
PA165984799-Diuretics_PD	22	0.93	398	0.77
PA165110622-Renin_Angiotensin_Agents_PD	8	-	566	0.77
PA161749006-SSRIs_PD	110	0.90	558	0.76
PA153627758-Potassium_Channel_Inhibitors_PD	18	-	546	0.76
PA2026-Glucocorticoid_HPA_Axis_PD	0	-	50	0.76
PA166122732-Succinylcholine_PK_PD	4	-	44	0.76
PA165986279-Acetaminophen_therapeutic_doses_PK	26	0.97	588	0.75
PA162356267-EGFR_Inhibitor_PD	4	-	1164	0.74
PA145011118-Estrogen_Metabolism_Pathway	28	0.75	208	0.74
PA145011120-Aromatase_Inhibitor_Breast_Cell_PD	26	0.57	192	0.74
PA2023-ACE_Inhibitor_PD	8	-	328	0.74
PA166114721-Uricosurics_PD	36	0.69	136	0.73
PA2038-Ifosfamide_PD	16	-	56	0.73
PA165111375-Benzodiazepine_PK	0	-	160	0.73
PA150981002-Vinka_Alkaloid_PK	0	-	136	0.73

8.3 Two Examples of Pathway Building

Although performance estimates are important, the real power of the network from Chapter 7 is that it allows us to use the scientific literature in a way it has not been used before - to assist human curators in performing a task that, until now, has been entirely manual. Two representative examples help illustrate potential use cases. One uses the classifiers we just developed; the other uses the network directly.

8.3.1 Tyrosine Kinase Inhibitors: Expanding a Pathway

The class of drugs called EGFR inhibitors target the epidermal growth factor receptor, which is expressed on multiple tissues, including those of the lung. These drugs,

Table 8.4. The full list of tyrosine kinase inhibitors found in the network from Chapter 7.

afatinib	erlotinib_hydrochloride	nilotinib
alectinib	flumatinib	oclacitinib
allitinib	foretinib	pacritinib
amuvatinib	fostamatinib	pelitinib
apatinib	fruquintinib	ponatinib
axitinib	gefitinib	quizartinib
bafetinib	ibrutinib	rociletinib
baricitinib	icotinib	ruxolitinib
binimetinib	imatinib	saracatinib
bosutinib	imatinib-mesylate	selumetinib
cabozantinib	imatinib_mesilate	semaxinib
canertinib	imatinib_mesylate	sunitinib
cerdulatinib	lapatinib	sunitinib_malate
ceritinib	lapatinib_ditosylate	tandutinib
crizotinib	lenvatinib	tasocitinib
dacomitinib	lestaurtinib	telatinib
dasatinib	linsitinib	tivantinib
dovitinib	masatinib	tofacitinib
erlotinib	masitinib	tofacitinib_citrate
erlotinib_hcl	neratinib	trametinib

which include the agents Tarceva (erlotinib) and Iressa (gefitinib), are often used to treat small-cell lung cancer, as well as several other cancers. PharmGKB has pharmacokinetic pathways for erlotinib and gefitinib, and a pharmacodynamic pathway for the full class of EGFR inhibitors, which includes erlotinib, gefitinib, and lapatinib, along with the identifiers used when these were experimental agents (Iressa = ZD1839, for example).

However, EGFR inhibitors are also part of a broader set of drugs called tyrosine kinase inhibitors. Tyrosine kinases are a class of enzymes important for the activation of proteins in signal transduction cascades. The drug name ending "-inib" is representative of this class of drugs. If we search for the ending "inib" in the network from Chapter 7, we retrieve a list of 60 different agents (some of which are different forms of the same active ingredient), which are listed in Table 8.4. PharmGKB currently does not have pathways for most of these agents.

We apply the the PD classifier for EGFR inhibitors (PA162356267-EGFR_Inhibitor_PD in Table 8.3) to the complete set of genes connected by network paths of length ≤ 2 to at least one drug in Table 8.4. For each drug, we rank all of the genes by the classifier's estimated probability that they belong in the pathway for that drug. We take the top 20 genes for each drug and restrict our set of genes to those nominated by 3 or more drugs. We then use the network to find all direct connections among the

set of 35 gene names that fulfill these criteria.

Figure 8.3 shows the result of this process. In it, we see the 35 genes, along with the connections among them from the Chapter 7, for four different themes: B (binding), E+ (increased expression), Rg (regulatory), and V+ (activation). Many of these connections could be verified using informal PubMed searches; for example, EGFR/ErbB2 are known to bind together to form a heterodimer, and they share a binding relationship in Figure 8.3a. The gene p53 codes for a protein that regulates the cell cycle and is important for tumor suppression. It binds to the promoter for p21, which is probably why the p53-p21 interaction appears in Figure 8.3a (binding) and 8.3c (regulation). The type of binding represented here is not direct protein-protein binding, which is what we had envisioned for the "B" theme, but it is binding. Akt (a serine/threonine kinase) does indeed increase the expression of NFkappaB (Figure 8.3b). All of these proposed interactions would require checking by a human curator before they could be incorporated into a pathway, but diagrams like these reduce the space of tens of thousands of potential interactions down to a few dozen.

One of the most interesting things about the set of genes in Figure 8.3 is not just that they do, indeed, all appear to interact with tyrosine kinases (whether by regulating their transcription, activating them, etc.) but that only *two* of those genes are directly connected to any of the drugs from Table 8.4 in the Chapter 7 network. Instead, the classifier is reasoning about longer-range connections. Of the top 5 most informative features for the trained random forest classifier for the PA162356267-EGFR_Inhibitor_PD pathway, 4 describe situations where improper regulation of a gene (theme L) is linked to a disease, and the origin drug is also connected to that disease. When the classifier sees that pattern, it will probably nominate the gene to be part of the pathway for the drug.

A final interesting note: one might assume that many of the genes in the curated EGFR inhibitor pathway also play a role in the pathways for other tyrosine kinase inhibitors. If we were only finding genes that already appeared in that pathway, this process would not be very interesting, as we would only be recapitulating what we already know. However, despite the fact that the EGFR inhibitor pathway contains 673 unique strings corresponding to gene names, our 35 genes from Figure 8.3 contain

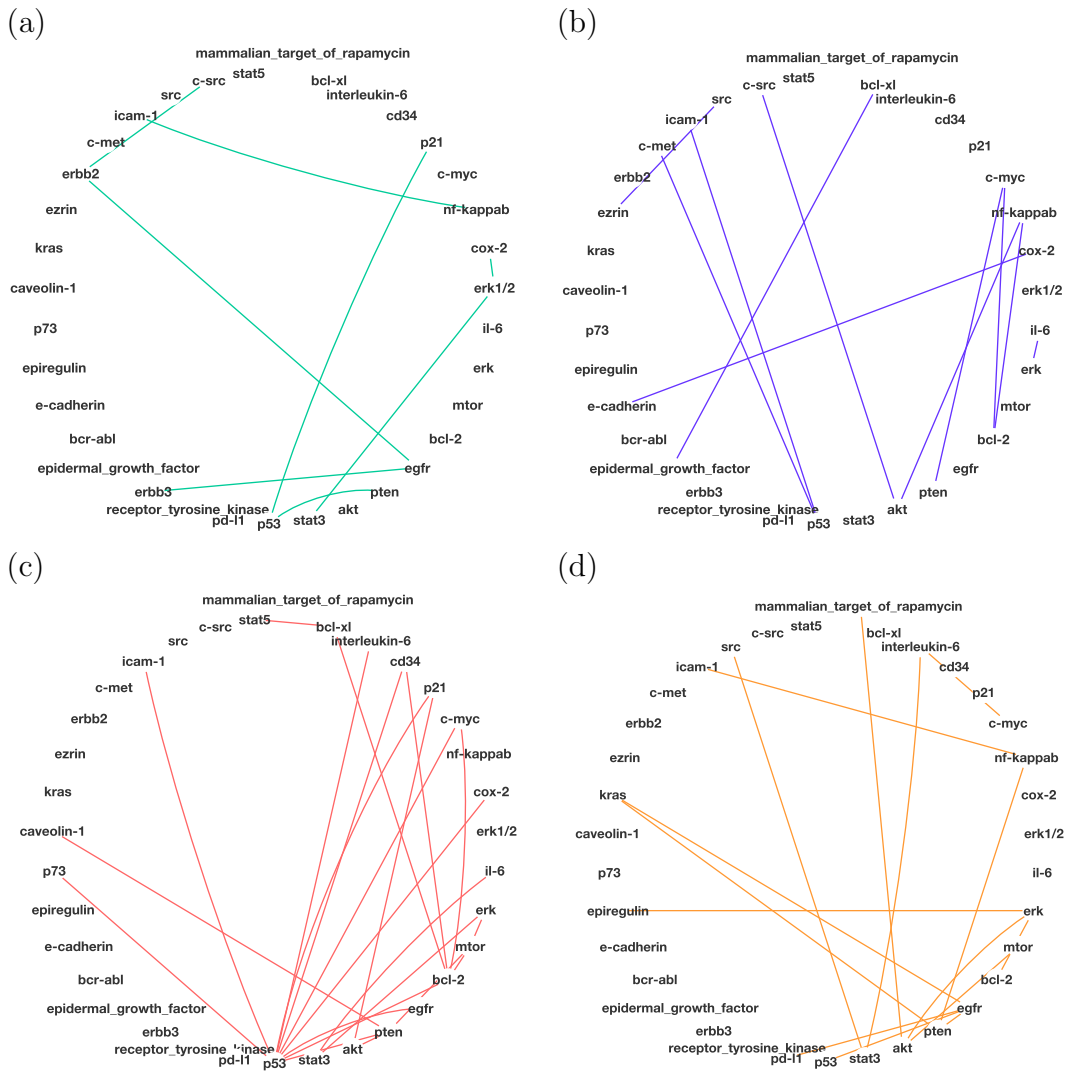


Figure 8.3. The most likely elements of the combined pharmacodynamic pathway for 60 tyrosine kinase inhibitor drugs, and their most likely relationships, based on the network from Chapter 7. (a) B (binding) relationships; (b) E+ (increased expression) relationships; (c) Rg (regulatory) relationships; (d) V+ (activation) relationships.

19 brand new gene candidates for the expanded tyrosine kinase inhibitor pathway that do not appear in the curated EGFR inhibitor pathway.

8.3.2 NSAIDs: Comparing Pathways for Related Drugs

One of the most commonly-used drugs in the United States is ibuprofen (Advil, Motrin). Ibuprofen works by inhibiting the enzyme cyclooxygenase (COX), which is required for the synthesis of prostaglandins (mediators of pain, inflammation and fever). It is part of a class of drugs called NSAIDs (Non-Steroidal Anti-Inflammatory Drugs). Other members of this class include aspirin and naproxen (Aleve, Naprosyn), and celecoxib (Celebrex). Currently, PharmGKB contains PK and PD pathways for ibuprofen and celecoxib.

One of the things that the network from Chapter 7 allows us to do is quickly compare related drugs based on their patterns of interaction with genes and other chemicals. By helping us isolate the differences in how these drugs behave, the network could tell us why they produce different side effects or treat some diseases more effectively than others.

For example, the drug celecoxib (Celebrex) differs from the other three in that it is a selective COX-2 inhibitor. There are actually two forms of COX: COX-1 and COX-2. Recently, pharmaceutical manufacturers have focused on selective COX-2 inhibitors (anything with the ending "-coxib"), believing that they might reduce the risk for peptic ulcers that is associated with nonselective COX inhibitors.

Table 8.5 shows the connections for celecoxib, focusing on properties that are not shared with the other NSAIDs. One connection in particular stands out. Celecoxib downregulates the expression of MDR1, an important cell membrane transporter. (We confirmed this fact via literature search.) Bacterial and cancer cells acquire drug resistance largely through the activity of MDR1, which shuttles drug molecules out of the cell before they have the chance to act. Because it downregulates MDR1, celecoxib has recently been tested as an adjuvant in cancer therapy; the hope is that it will keep cancer cells from evolving resistance to other drugs. Currently, MDR1 is not part of celecoxib's PD pathway, but perhaps it should be.

Table 8.5. Dependency path themes for celecoxib. Interactions that did not include celecoxib were excluded. If an interaction had the theme "E" and also a more specific theme ("E+" or "E-"), the "E" was also excluded. The "in pathway" column has a "1" if the gene is part of the PharmGKB celecoxib PD pathway and a "0" if not.

Gene	Theme	# Drugs Interacting	Drugs (Besides Celecoxib)	In Pathway
cox-2	N	3	aspirin,naproxen	1
cox-2	E-	2	aspirin	1
vascular_endothelial_growth_factor	E	2	aspirin	0
5-lipoxygenase	E-	1		0
akt	E+	1		1
bcl-2	E	1		0
carbonic_anhydrase_ii	E-	1		0
cox-2	K	1		1
cyclo-oxygenase-2	N	1		0
cyclooxygenase-2	E-	1		0
cyclooxygenase-2	K	1		0
cyclooxygenase-2	N	1		0
cyclooxygenase_2	N	1		0
cytochrome_p450_2c9	E	1		0
e-cadherin	E	1		0
heme_oxygenase-1	E+	1		0
matrix_metalloproteinase-10	E-	1		0
mdr1	E-	1		0
rad51	E	1		0
smn	E+	1		0
stat5	E+	1		0
sulfotransferase_2a1	E-	1		0
urokinase-type_plasminogen_activator	E-	1		0
vegf	A+	1		1

8.4 Discussion and Future Work

Exploring the use of the Chapter 7 network for pathway building has raised several important questions and concerns that we will need to consider as we continue with this work.

First, in Table 8.1, we have established that most of the PharmGKB pathways' chemical-gene and gene-gene connections are not represented in the network. This led us to build classifiers that could use longer-range connections to predict pathway membership, but that practice has the disadvantage of being less transparent. The meaning of Table 8.5 is very clear: I can see each connection and its associated theme. But in Figure 8.3, the meaning is less clear. The gene-gene connections and their themes are present, but where did those genes come from in the first place? They were selected by the classifier's examining longer-range connections that are not as easily interpretable. It is possible that, in the end, we would prefer to focus on expanding our text corpus to capture more connections, rather than allowing longer-range network connections to function as features in our pathway classifier.

This chapter has also driven home another difficulty associated with all of the work in this thesis: the difficulty of evaluating text mining algorithms using sources other than annotated text. We were fortunate to have the PharmGKB pathways database to use for evaluation, but the evaluation itself was still difficult, and ambiguous in many cases. Was a relationship not picked up correctly because the mapping of dependency paths to themes in Table 7.2 was wrong? Or was it actually described incorrectly, or vaguely, in the text itself? Or was it simply never mentioned, or mentioned with a variant of one or both entity names that PubTator's NER system did not recognize? Without an annotated corpus, it's difficult to distinguish these sources of error.

Finally, Figure 8.3 illustrates a subtle complicating factor that is not necessarily apparent from Table 7.2: whether or not it is worth distinguishing themes at all. Although B (binding) relationships tend to be described differently in the text from Rg (regulatory) relationships, as evidenced by their positioning in Figure 7.4, the two themes are not mutually exclusive. The same goes for "E+" and "V+" relationships: does it matter if the relationship is described as "activation" or "increased expression"?

Even if the two are semantically distinct, for many applications, we are only interested in direction: does administering this drug increase or reduce the ability of this enzyme to do its job? How exactly that happens may not matter. For some applications, we could actually hurt performance by creating too fine-grained a distinction between similar themes.

Chapter 9

Drug-Drug Interactions Revisited

At long last, we return to the project that first inspired this dissertation: the prediction and mechanistic explanation of drug-drug interactions (Chapter 2). In Chapter 2, we found it remarkably straightforward to predict drug-drug interactions (DDIs) based on drug-gene relationships described in the biomedical literature. However, our ability to do so at scale was limited by our use of a manually-constructed ontology to normalize the drug-gene relationships. That problem inspired the rest of this dissertation.

We begin this final chapter by reexamining our findings from Chapter 2 and discovering that while they were correct from a technical standpoint, they were also much less generalizable than we anticipated. We explore the reasons why, the implications for our pathway results from Chapter 8, and what these findings teach us about how best to apply the network from Chapter 7 to drive biomedical research forward. We show how the network can used to create two specific mechanisms for different types of DDIs, and end the chapter with a summary of our future plans for the network.

9.1 A Fresh Look at Old Findings

Our approach to DDI prediction in Chapter 2 can be summarized as follows: we identified several hundred thousand *drug-gene-drug* connecting paths (potential DDI mechanisms), we extracted the features along these paths (verbs, context words, gene

name), and we used these features in a random forest classifier, taking the out-of-bag estimate of prediction accuracy as our measure of success.

Looking at this again, we notice two things, both of which arise because the training examples were connecting paths and not entity pairs:

- The same entity pair could be represented multiple times.
- Only entity pairs connected by *drug-gene-drug* paths in the network were represented.

We chose mechanisms instead of entity pairs for our training examples because the same drug pair can be connected by multiple paths. We treated each path separately because we were interested both in predicting whether two drugs would interact and in finding the most likely mechanism of interaction. However, this could also lead us to misinterpret the accuracy of our predictions.

For example, as we observed in Chapter 8, most drug-drug pairs are not connected by genes. This includes drug-drug pairs that interact - we estimate that only about 18% of interacting drug pairs are connected by a gene. Therefore, by restricting our training examples to paths via a connecting gene, we are implicitly eliminating many interactions that occur via longer-range paths. In addition, allowing the same entity pair to be represented multiple times in the training set means we could be overfitting to features specific to that entity pair (such as unusual connecting genes) that do not represent true mechanisms of interaction.

9.1.1 Revisiting Prediction

We used the same list that was used in our 2012 project [107] to create a list of 5000 known DDIs and 5000 pairs of drugs that, while both individually present in the list of DDIs, were not known to interact. Drug pairs did not need to be connected by a path in the network to be included. We found all network theme combinations between the two drugs in each pair, as well as all of the entities connecting the pair, binarized all the features, and built a random forest classifier (again with 100 trees) to distinguish DDIs from non-DDIs. The results are shown in Table 9.1.

We see that when the analysis is performed at the level of drug-drug pairs rather

Table 9.1. Performance of several random forest classifiers, each with 100 trees, trained on 10,000 training examples. The mean AUC from 10-fold cross validation is reported. The feature definitions are as follows: "themes only" means that if two drugs are connected by dependency paths with the themes "N" and "B" to a common gene, say CYP3A4, the feature "N|B" is included but the identity of the gene is not. In contrast, "entities only" means that CYP3A4 would be included but "N|B" would not. "Entities + themes" means both types of feature are included separately. "Combined" means that the connecting path "N|CYP3A4|B" is included as a single feature.

Connecting Entities	Feature Type	Number of Features	Mean AUC
genes + diseases/phenotypes	entities only	5943	0.940
	themes only	143	0.615
	entities + themes	6086	0.798
	combined	55,132	0.740
genes only	entities only	245	0.552
	themes only	94	0.531
	entities + themes	339	0.542
	combined	1896	0.542

than mechanisms, performance drops considerably, at least when only *drug-gene-drug* connecting paths are included. Mainly this has to do with the fact that the majority of training examples have no connecting *drug-gene-drug* path whatsoever. When *drug-phenotype-drug* paths (which were not available at the time we performed the analysis in Chapter 2, but were created in Chapter 7) are included, performance increases dramatically, even surpassing the performance in Chapter 2. However, including theme information actually hurts performance relative to just including the connecting entities.

The evidence is clear and corroborates our ordering of feature importance in Figure 2.6: it is the identities of the connecting entities, much more than the themes, that determines whether or not two drugs interact. In addition, if no connecting entity exists, it is impossible for us to make a prediction about the likelihood that the two drugs interact without considering paths beyond length 2.

9.1.2 From Prediction to Mechanism

If the theme labels in the network do not contribute to our ability to predict DDIs, and show only a weak correlation with pathway membership for most pathways (Chapter 8), why bother creating a labeled network at all? Why do I, as a human reader, find the

diagram in Figure 8.3, and the information in Table 8.5, helpful in predicting pathway membership, while my computer (apparently) does not?

The answer, I believe, is that I bring a great deal of background knowledge to my reasoning about biomedical entities. The computer does not. If there is a rare cell membrane transporter that is responsible for shuttling a drug into the cell, and another drug inhibits the expression of that transporter, I know that the effect of the first drug will be reduced. However, it is not the case that every single situation in which a drug inhibits the expression of a protein that binds another drug will lead to a DDI - the identity of that protein matters. The problem is, aside from common proteins that interact with a lot of drugs, it's difficult to build a training set that accounts for all of the complex/rare interaction types that mediate DDIs. I can reason by analogy ("This rare transporter has a similar function to this other transporter...") but the computer cannot.

In light of this, it seems to make the most sense to use the network for comparison and the generation of mechanistic hypotheses, rather than large-scale prediction. We can find DDIs that share similar mechanisms by examining the drugs' patterns of connection with intervening genes or phenotypes. We provide two different examples of use cases below.

9.2 Two Mechanistic Examples

9.2.1 Metoprolol and Dextromethorphan

In my PhD defense, I presented an example of a drug interaction that had been predicted by our classifier in Chapter 2 but was not part of our training set. The classifier predicted that the drug metoprolol, a beta-blocker (used to treat hypertension), would interact with the cold medicine dextromethorphan (found in Robitussin, for example), because both drugs were metabolized by the same enzyme, CYP2D6. We later confirmed this association via literature search [140].

We see this exact relationship reflected in our network from Chapter 7, despite the fact that it used an entirely different named entity recognition system (PubTator) and

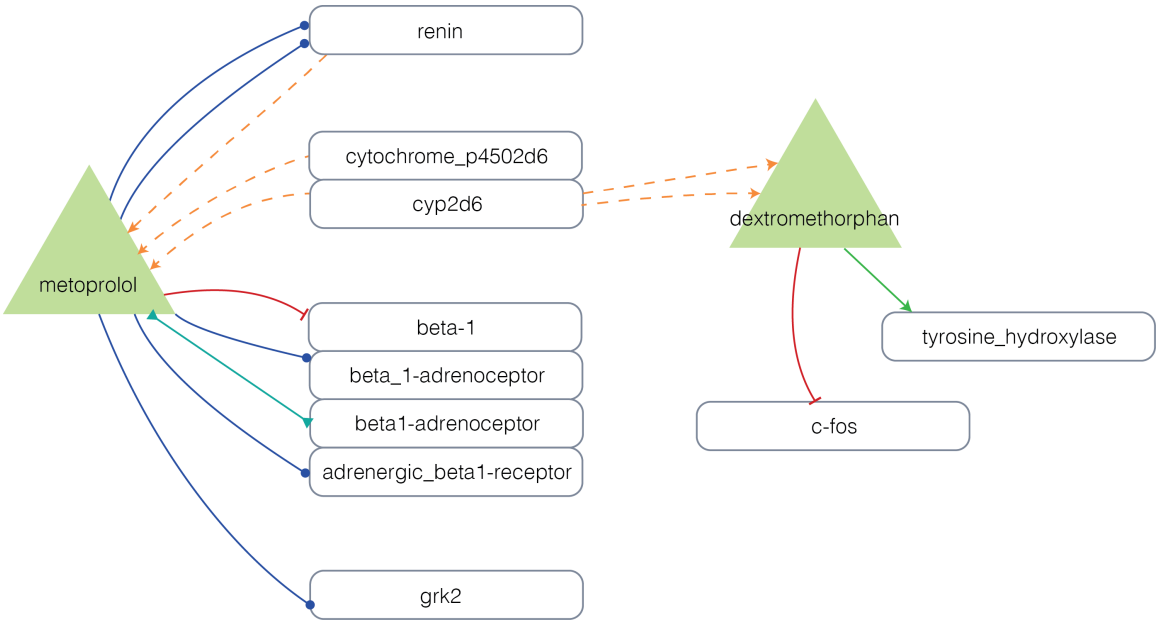


Figure 9.1. Diagram of the DDI between metoprolol and dextromethorphan, with labels corresponding to dependency paths in the network from Chapter 7. The edge styles correspond to different themes, the meanings of which can be found in Table 7.2. Blue with circle: E, orange dashed arrow: K, red with hash: E-, green with arrow: E+, turquoise with backward arrows: B.

relationship extraction technique (EBC) from the network in Chapter 2. A diagram of all of the drug-gene interactions for metoprolol and dextromethorphan, derived from the labeled edges in the Chapter 7 network, can be found in Figure 9.1. The only link between the two drugs is their shared metabolic relationship with CYP2D6. In the diagram, we also see metoprolol's mechanism of action - its effect on the beta-1 adrenoceptor, to which it binds.

Since our new network includes information on gene-disease relationships, we can also investigate likely side effects of this DDI. This type of reasoning is important for recognizing DDIs in the clinic, since we cannot directly observe the inhibition or induction of proteins, etc.

The paper reporting the interaction between metoprolol and dextromethorphan [140] stated that a patient who had been taking metoprolol for a long time and was given dextromethorphan developed a severe case of myoclonus (muscle twitching).

Looking at all of the genes in Figure 9.1 and their associations in our network, we observe that tyrosine hydroxylase activity is indeed associated with myoclonus. In fact, familial tyrosine hydroxylase deficiency is associated with a chronic form of myoclonus called myoclonus-dystonia syndrome [138].

If the long-term administration of metoprolol had essentially "used up" all of the available CYP2D6 in the patient's liver, there would be little left over to metabolize the newly-administered dextromethorphan. This could cause dextromethorphan to build up in the body and, in turn, for tyrosine hydroxylase activity to increase, causing myoclonus[1].

The interaction between metoprolol and dextromethorphan is an example of a *pharmacokinetic* (PK) DDI, which occurs when one drug modifies the absorption, metabolism, distribution or excretion of another drug relative to what would happen if the second drug were administered on its own. Two other examples of pharmacokinetic DDIs are shown in Figure 9.2.

9.2.2 Melatonin's Effect on Dopamine

Situations where one drug alters the ability of another drug to perform its function are called *pharmacodynamic* (PD) DDIs. This type of DDI is extraordinarily difficult to uncover, given the multiplicity of effects that drugs typically have on the body, many of which are unknown. Two examples of pharmacodynamic DDIs are shown in Figure 9.3. One common situation where PD DDIs occur is when a drug binds to a receptor that is not part of its intended mechanism of action. That receptor may be the intended target of another drug, causing an interaction if the two drugs are coadministered.

Many psychoactive drugs affect the dopamine pathway. For example, rasagiline, an anti-Parkinsons agent, is an irreversible inhibitor of the enzyme monoamine oxidase B (maoB), a key player in the breakdown of dopamine. Both of these relationships are

[1]The effect in the paper on familial tyrosine hydroxylase deficiency actually works in the reverse direction - a decrease in tyrosine hydroxylase activity causes myoclonus. However, we don't know what happens when tyrosine hydroxylase activity increases beyond normal bounds, and we do know that its activity is associated with myoclonus, so we might reasonably assign it as the culprit.

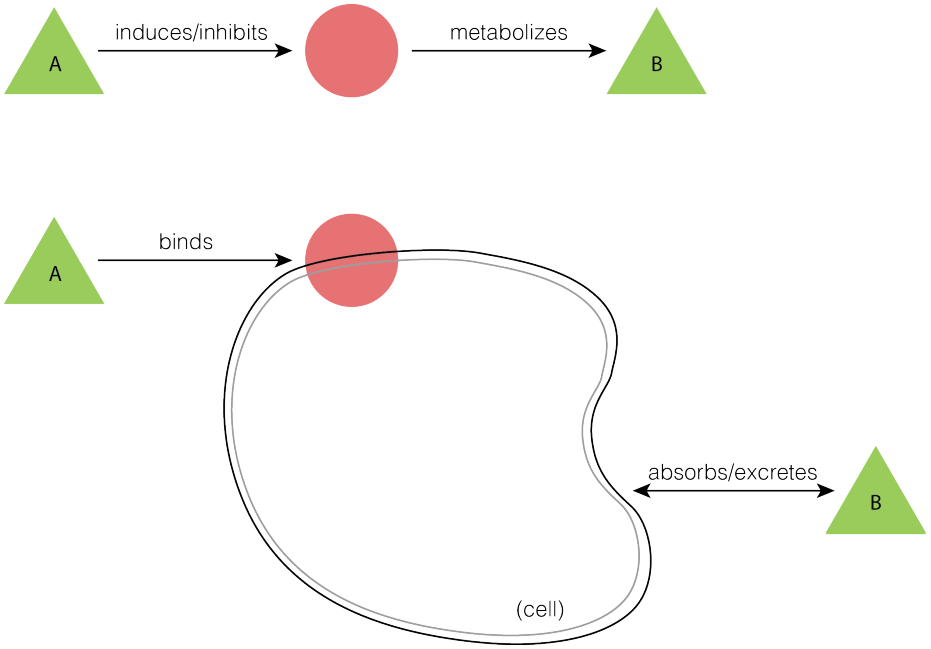


Figure 9.2. Two possible mechanisms for pharmacokinetic drug interactions. In the first example, drug A affects the activity of a protein that is responsible for metabolizing drug B. In the second example, drug A binds to a receptor on a cell that is responsible for absorbing or excreting drug B. Drug A's binding somehow affects the ability of the cell to perform its absorption/excretory function, which affects the levels of drug B in the body.

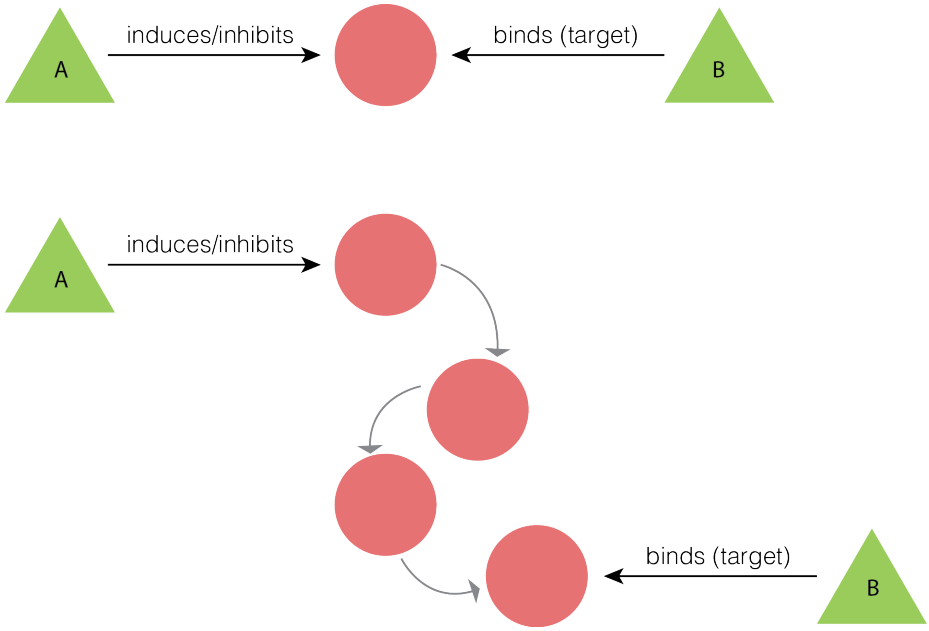


Figure 9.3. Two possible mechanisms for pharmacodynamic drug interactions. In the first example, drug A induces or inhibits the production (translation, transcription) of a protein that is the target of another drug. If there more of this protein, the dosage of drug B may need to be increased to have the same effect. If there is less, we may need to decrease the dose of drug B. In the second example, drug A affects one member of a genetic pathway (see Chapter 7) and somewhere down on that pathway is the target of drug B.

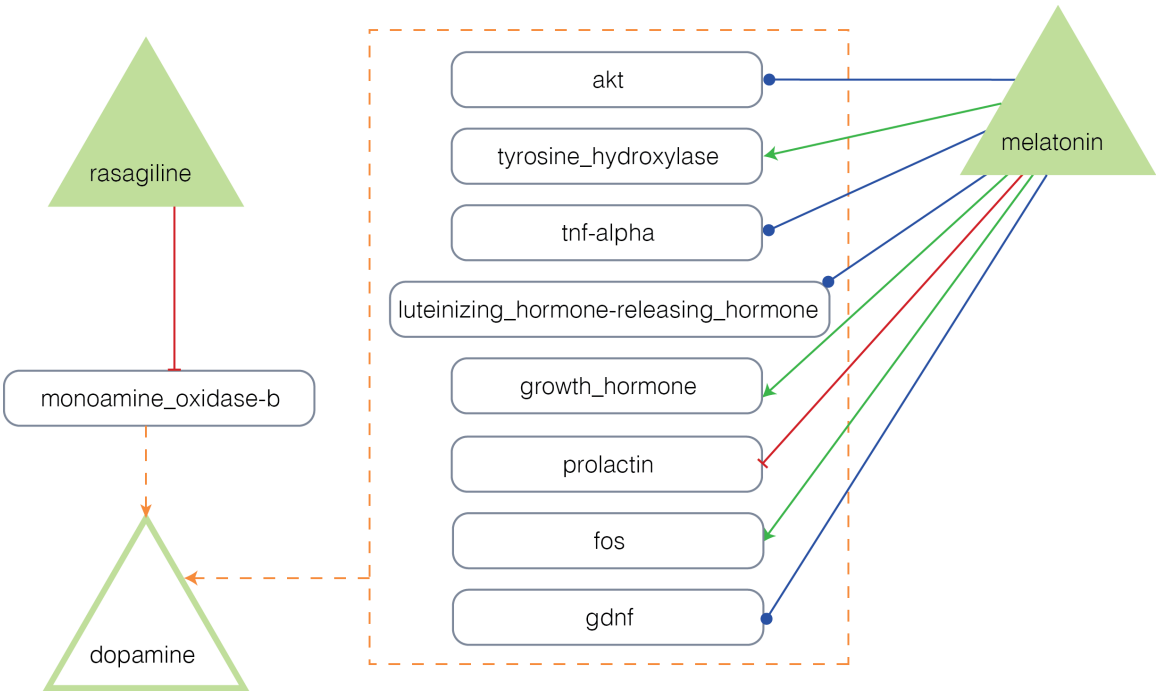


Figure 9.4. Diagram of potential PK DDI mechanisms for melatonin and drugs like rasagiline, which affect dopamine levels. The only relationships with melatonin that are included are those that affect the expression of proteins responsible for the pharmacokinetics of dopamine. Labels correspond to dependency paths in the network from Chapter 7. The edge styles correspond to different themes, the meanings of which can be found in Table 7.2. Blue with circle: E, orange dashed arrow: K, red with hash: E-, green with arrow: E+.

captured by our network, and are shown in Figure 9.4.

The drug melatonin, which is also an endogenous compound produced by the body, has long been known to have a lowering effect on dopamine levels in parts of the brain. The exact mechanism behind this, however, is unclear [158]. Melatonin, therefore, is known to engage in a pharmacodynamic interaction with nearly every drug that affects dopamine levels. Sometimes this effect is synergistic, as in the case of some first-generation antipsychotics that work by suppressing dopamine, and sometimes, as with rasagiline, it is antagonistic. As more information about melatonin's role in the dopamine pathway becomes known, more edges in Figure 9.4 will be filled in. The important thing here is that the key relationships underlying melatonin's PD interactions with rasagiline and other dopamine-affecting drugs are captured by our

network.

9.3 Summary and Future Work

It is clear just from these two examples that, due to their relatively predictable and short-range mechanisms, pharmacokinetic DDIs will constitute the low-hanging fruit of our network. In the metoprolol-dextromethorphan example, we were able both to characterize the interaction and to attach likely side effects. Pharmacodynamic DDIs are less well-defined, and will probably require us to improve and expand the gene-gene portion of our network.

These two case studies do not prove that the network is broadly useful for identifying DDIs. However, they do give us confidence that the themes produced by our network correspond (at least in two cases) to reality. That is both surprising and exciting given where these themes came from: unsupervised clustering of dependency paths using similarity scores derived from EBC, using no human annotated text whatsoever.

In the end, we believe that a complete, labeled network of biomedical relationships derived from the literature will prove a valuable resource to the biomedical research community, since it allows us to connect facts from across the literature quickly and easily. The network appears to have more power as a tool for understanding than for prediction, but that could change as our relationship classes become more well-defined and as we expand the network to cover ever larger sources of text.

Appendix A

Explanations of Clusters

A.1 Drug-Gene Clusters from Chapter 6

The following table is a detailed explanation of the clusters of drug-gene pairs shown in Figure 6.3.

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
1a	Synthesis	34	synthase	aldosterone, P450aldo	0.0	17.6	Many of the drugs in this cluster are endogenous compounds.
11 beta-Hydroxylase (P45011 beta) and aldosterone synthase (P450aldo) were situated in the inner mitochondrial membrane of the zona fasciculata-reticularis cells and in that of the zona glomerulosa cells, respectively. (9617077)							
1b	Activation	134	increased activity	curcumin, caspase-8	9.0	6.7	In this cluster, activation is frequently associated with phosphorylation.
Curcumin also stimulated the activity of caspase-8, which initiates Fas signalling pathway of apoptosis. (11396178)							
2	Enzyme activity	45	activity	estradiol, E2DH	6.7	6.7	The gene is typically an enzyme that chemically modifies the drug. A few transporter pairs are also present, such as (ornithine, ORNT1).
A fraction of the estradiol 17 beta-oxidoreductase (E2DH) activity in the vesicle remained associated to the membrane after disruption and treatment with 2 M NaCl. (3459941)							

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
3a	Substrates	64	substrate	aminopterin, hOAT1	29.7	7.8	Relatively few mentions of "metabolism" compared to 3b and 3c. Reference to transporters such as P-gp, hOAT1, SERT.
These findings show that both aminopterin and methotrexate are substrates of hOAT1 and hOAT3, and that there are differences between the antifolates in terms of their transport characteristics. (20460822)							
3b	Metabolism	131	metabolized	rosiglitazone, CYP2C8	37.4	0.8	Frequent reference to liver cytochromes such as CYP3A4 and CYP2D6.
Rosiglitazone, a thiazolidinedione antidiabetic medication used in the treatment of Type 2 diabetes mellitus, is predominantly metabolized by the cytochrome P450 (CYP) enzyme CYP2C8. (15606443)							
3c	Substrates that (often) also affect activity	70	substrate	efavirenz, CYP2B6	37.1	5.7	The drug-gene pairs in this sub-cluster are mentioned together less frequently in the literature than those in 3a or 3b.
Efavirenz is extensively metabolized by CYP2B6, and associations between CYP2B6 polymorphisms and plasma efavirenz exposure have been reported. (20639527) Our results confirm that efavirenz induces CYP2B6 enzyme activity in vivo, as demonstrated by an increase in bupropion hydroxylation after 2 weeks of efavirenz administration. (18989234)							
4	Third party involvement	28	inhibits... to/	rapamycin, PHAS-I	3.6	3.6	All of the drug-gene pairs in this cluster are connected by exactly one path, and the paths are unusual. They often refer to the involvement of a third molecule of some kind, raising the possibility of three-way interactions among drugs and genes.
Rapamycin may inhibit translation initiation by increasing PHAS-I binding to eIF-4E. (7629182)							
6	Co-administration	172	in presence of	sunitinib, IFN-alpha	0.6	0.6	This cluster illustrates the blurry line between drugs and genes (proteins) since many drugs (in this case, IFN-alpha) are also proteins.
Herein, we report the results of a phase I dose-finding study of sunitinib in combination with IFN-alpha as first-line treatment in patients with metastatic RCC. (19213665)							

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
7c	Increased production	141	induced, production, increase	PGE2, VEGF	1.4	1.4	Cluster 7 is distinguished by the presence of many proteins that act as drugs. These include IL-2, gp120, and PGE2.
These findings raise the possibility that endogenous PGE2 stimulates VEGF and bFGF mRNA expression in Mueller cells in vivo under conditions in which PGE2 production is increased, such as in injury. (9501870)							
7d	Raised levels	52	levels, production	cisplatin, Rad51	5.8	3.8	Similar in theme to 7a-c, descriptions from this cluster involve drugs that raise protein levels. Sentences mostly report experimental results.
In addition, gefitinib decreased cisplatin- or MMC-elicited Rad51 protein levels by increasing Rad51 protein instability. (18544565)							
8a	Antagonists	101	antagonist, blocker	plerixafor, CXCR4	11.9	39.6	Cluster 8 references inhibition more generally. EBC learns that antagonism (cluster 8a) is a subclass of inhibition.
Plerixafor is a selective antagonist of CXCR4 used for mobilization of hematopoietic stem cells (HSCs) for autologous stem cell transplantation (SCT) in patients with multiple myeloma (MM) and non-Hodgkin lymphoma (NHL). (19748593)							
8c	Inhibition	380	inhibitor of, inhibits	sildenafil, PDE5	18.7	37.9	Cluster 8c is large and includes some interesting smaller subclusters, such as antibodies against particular proteins, and inhibition, specifically, of protein activity or phosphorylation.
Although active sites of PDEs are apparently structurally similar, PDE4 is specifically inhibited by selective inhibitors such as rolipram, while PDE5 is preferentially blocked by sildenafil. (15224132)							
9	Specific drug-protein interactions	56	target, kinase, protein	hyaluronate, GHAP	3.6	14.3	These are pairs where the protein is named for its function, which involves a particular action on the drug in question. In the second sentence, the pair is pyridoxal/Pdxk.
Cells were probed with the glial hyaluronate binding protein (GHAP) which was itself then visualized by conventional indirect immunofluorescence. (2070821) Transcriptome profiling revealed pyridoxal kinase (Pdxk) as a target gene of PAR bZip proteins in both liver and brain. (15175240)							

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
10	Inhibitors and substrates	70	inhibitor, substrate, metabolized	verapamil, P-gp	30.0	4.3	Many drugs act as both inhibitors and substrates of proteins, including ritonavir/CYP3A4, quinidine/P-gp, and omeprazole/CYP2C19, all found in cluster 10.
It has been reported that verapamil and atorvastatin are inhibitors of both P-glycoprotein (P-gp) and microsomal cytochrome P450 (CYP) 3A4, and verapamil is a substrate of both P-gp and CYP3A4. (18193210)							
11	Inhibition	148	inhibitor of; G inhibitors, such as D; inactivator	miglitol, alpha-glucosidase	12.2	27.0	There is little difference in meaning between this cluster and cluster 8c, except that there are variations in phrasing that are more common to one or the other cluster.
alpha-Glucosidase inhibitors, such as miglitol, are drugs that have greater affinity towards this enzyme in comparison to carbohydrates. (19563873)							
12	Receptors	80	receptor(s), gene, antagonist	urokinase, uPAR	1.3	32.5	Cluster 12 contains a subcluster primarily composed of antagonist pairs, and a larger subcluster involving pairs where the gene is described as the "receptor" for the drug.
The urokinase receptor urokinase-type plasminogen activator receptor (uPAR) is a surface receptor capable of not only focalizing urokinase-type plasminogen activator (uPA) - mediated fibrinolysis to the pericellular micro-environment but also promoting cell migration and chemotaxis. (22285761)							
13	Activation	112	activated, increased expression	simvastatin, Rac1	0.0	0.0	This is the largest cluster with zero representation of either PGx or drug-target relationships. The pair in the second sentence is estradiol/HO-1.
The small GTPase Rac1 was activated by simvastatin, and this was required for both PKB activation and IL-1beta secretion. (18684863) Estradiol increased HO-1 expression by 2- to 3-fold, an effect blocked by SU5416, and PPT mimicked the effects of estradiol on HO-1. (20644008)							
14a	Agonists	129	agonist, hormone, analog	sumatriptan, 5-HT1B	7.0	33.3	

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
We compared the vasoconstrictor effects of 5-HT with those of the selective 5-HT1B/1D-receptor agonists sumatriptan and rizatriptan in human isolated cranial (middle meningeal) arteries. (9862247)							
14b	Activation / stimulation	138	activates, induced, stimulates	resveratrol, AMPK	1.4	4.3	Focus is similar to cluster 13 but notably, there is relatively little reference to expression.
Moreover, resveratrol activated AMPK and inhibited phosphorylation of 4E-BP1 and S6 in diabetic rat kidneys. (20332614)							
15	Protein binding	28	binds to; binding to	glibenclamide, SUR1	7.1	35.7	
ATP, in the presence of an ATP-regenerating system to oppose hydrolysis during incubation, inhibited glibenclamide binding to SUR1 and SUR2B (Y1206S) by approximately 60%, to SUR2A (Y1206S) by 21%. (12145099)							
17d	Experimental methods	151	treatment, concentration, toxicities, mice, cells	dasatinib, STAT3	1.8	2.4	This cluster includes many sentences describing observed effects on expression/activity, but not as many as other nearby clusters. Cluster 17d is also home to one insidious error: the term "DLTS" ("dose-limiting toxicities") identified as a gene.
We hypothesized that the reactivation of STAT3 after dasatinib treatment represents the engagement of a compensatory signal for cell survival that blocks the antitumor effects of SFK inhibition. (17634553) Treatment of cultured cells from WT or Delta 18 COX-2 mice with flurbiprofen, which blocks substrate-dependent degradation, attenuates COX-2 degradation, and treatment of normal mice with ibuprofen increases the levels of COX-2 in brain tissue. (19758985)							
17e	Effect on expression	148	investigate effect on G expression; alter, affect, decrease, regulated	colchicine, MEFV	1.3	0.0	If directionality of effect is reported in cluster1 17e, it is most often inhibition.
To investigate the effect of colchicine (the main therapeutic agent for FMF patients) and certain inflammatory cytokines (IL-1 beta, TNF-alpha, IFN-alpha, IFN-gamma) on MEFV expression and C5a inhibitor activity in neutrophils and primary peritoneal fibroblast cultures. (11802319)							
18a	Induction of expression	123	increased / induced expression	imatinib, CXCR4	1.6	1.6	Typically experimental results reporting a positive effect of the drug on gene expression.

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
In KBM5 and K562 cells, imatinib, INNO-406, or IFN-alpha increased CXCR4 expression and migration. (18202009)							
18b	Effect on expression, usually induction	65	by expression, inducer of, was induced by	melatonin, bcl-2	1.5	1.5	In many sentences, we know only that the effect of the drug on the expression of the gene was investigated. If directionality of effect is reported, it is most often induction.
Melatonin given before the ischemia enhanced the expression of bcl-2 in the penumbra area and had no significant effect on the expression of bax. (10678086)							
19	Inhibition of activation	41	inhibited / suppressed activation (of G)	fluvastatin, NF-kappaB	4.9	4.9	This is another set of three-way interactions where the drug is suppressing activation of the protein by some other molecule.
Interestingly, fluvastatin suppressed IFN-gamma-induced NF-kappaB activation in parallel with p38 MAPK phosphorylation. (19594754)							
20a	Effect on expression, usually inhibition	54	expression by, expression of, inhibited expression, decreased, reduced	montelukast, iNOS	0.0	3.7	There is a fairly even split in this cluster between methods and results.
This study investigated the effects of montelukast (a leukotriene receptor antagonist) on iNOS expression and activity in a Brown Norway (BN) rat allergic inflammation model and in L2 lung epithelial cell. (14559427)							
20b	Decreased levels	59	decreased levels, inhibited expression, suppression	gefitinib, Rad51	1.7	0.0	Note that the example sentence here is identical to that in cluster 7d, but the drug in question is different. This single sentence describes two separate relationships with different characters.
In addition, gefitinib decreased cisplatin- or MMC-elicited Rad51 protein levels by increasing Rad51 protein instability. (18544565)							
21	Inhibited activity / expression	76	inhibited activity, inhibited expression	minocycline, MMP-2	3.9	10.5	Focus is experimental observations, as opposed to stated prior knowledge (the dominant theme in cluster 8c).

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
Intraperitoneal minocycline at 45 mg/kg concentration twice a day (first dose immediately after the onset of reperfusion) significantly reduced gelatinolytic activity of ischemia-elevated MMP-2 and MMP-9 ($p < 0.0003$). (16846501)							
22	Inhibition	78	inhibited; G inhibitors, such as...	trastuzumab, HER2	10.3	17.9	There are some subtle differences between cluster 22 and cluster 8. Most notably, cluster 22 never references antagonism. Cluster 22 also contains some descriptions that never occur in cluster 8, such as "inhibited induction of" and "inhibited activation". Similarly, cluster 8 contains some descriptions (besides those of antagonists) that never occur in cluster 22, such as "inhibitors of G, such as...", "decreased activity", and "inhibit activity".
The humanized anti-HER2 monoclonal antibody trastuzumab inhibits the activation of HER2 and its multiple downstream signaling pathways, including the Ras/mitogen-activated protein kinase pathway. (18451248)							
23	Protein binding (and) affects activity	33	activity, protein, binds	gp120, DC-SIGN	0.0	12.1	This small cluster actually contains two smaller subclusters, one of which focuses on protein activity and the other on binding. The descriptions of these drug-gene pairs include some different variants of those in clusters 15 and 25f.
gp120 additionally binds to DC-SIGN, a C-type lectin expressed on immature dendritic cells. (11825572) Moreover, exposure of hippocampal neurons to dexamethasone significantly increased caspase-3 activity, which was inhibited by co-treatment with agmatine. (16777341)							
24	Patients with disease (error)	92	treatment, patients, disease	glyburide, NIDDM	3.3	2.2	This cluster illustrates one problem associated with using simple string matching to lexicons to identify drugs and genes: COPD and NIDDM are both gene names. Notably, however, these types of errors are "quarantined" together in the dendrogram.

	Theme	Cluster size	Key word /phrase	Example drug-gene pair	% PGx	% Drug-Target	Comment
140 NIDDM patients being treated with either glyburide (n = 70) or glipizide (n = 70) were randomly selected from the populations of patients receiving either drug using computerized pharmacy records. (1421641)							
25c	Affects secretion / release	50	secretion	octreotide, calcitonin	0.0	0.0	Genes (proteins) in this cluster are generally hormones or cytokines, such as gastrin, lactogen, IL-1RA, and IL-13.
The inhibitory effect of octreotide on rGRF-induced calcitonin secretion was partially abolished by pretreating the cells with pertussis toxin. (1355052)							
25d	Expression	252	on expression, by expression, inhibited / increased expression	indomethacin, MCP-1	2.0	2.0	The directionality of the drug's effect on expression varied within this cluster. The sentences mostly report experimental findings.
We found that, in murine podocytes, expression of monocyte chemoattractant protein 1 (MCP-1) in response to TNF-alpha was suppressed by indomethacin but not by ibuprofen. (18799549)							
25f	Affects activity	38	activity, on activity	amitriptyline, EAAT3	2.6	10.5	
Our results suggested that amitriptyline at clinically relevant concentrations reversibly reduced EAAT3 activity via decreasing its maximal velocity of glutamate transporting function. (19405995)							

A.2 Dependency Path Clusters from Chapter 7

Table A.2. Chemical-gene clusters from Figure 7.1.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / G)
3	36	inhibition	"C, a G inhibitor"	ARRY-614 / p38
			"G specific inhibitor, C"	naringenin / Smad3
			"C, an inhibitor of G"	PSC_833 / P-glycoprotein
			"G inhibition by C"	NVP-AUY922 / Hsp90
			"effects of the G inhibitor, C, on . . . "	SCH_34826 / enkephalinase
5	12	effect on protein activity	"[chemical]-dependent effects of C on G activity"	fenfluramine / renin
			"effect of C on G activity"	donepezil / acetylcholinesterase
			"inhibition of G activity by C"	plumbagin / Nox-4
			"study on interaction of C with G"	caffeine / myoglobin
			"G activity in patients on C"	tacrolimus / CYP3A4
6	29	agonism / antagonism	"effect of C, a selective G antagonist"	MTEP / mGluR5
			"C, a G agonist"	roxindole / 5-HT1A
			"inactivation of G by C"	mitomycin_C / DT-diaphorase
			"G agonist, C, . . . "	ciglitazone / PPAR-gamma
			"study of a G antagonist, C, . . . "	CI-988 / CCK-B_receptor
8	42	secretion, production, synthesis	"effect of G on C metabolism"	dopamine / cholecystokinin
			"C inhibits G secretion"	Dasatinib / TNF-alpha
			"effects of G on C metabolism"	steroid / angiotensin_ii
			"C stimulates G synthesis"	prostaglandin_e2 / interleukin-1_beta
			"C regulates G expression"	baicalin / tlr4
9	76	affects expression	"C inhibits G expression"	AG490 / NFATc1
			"effect of C on G production"	neopterin / erythropoietin
			"C induces the expression of G"	Nicotine / C-reactive_protein
			"C upregulates G expression"	Dexamethasone / Kv1

Table A.2. Chemical-gene clusters from Figure 7.1.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / G)
			"effects of C on the expression of G"	letrozole / HOXA10
10	20	inhibition of activity / expression	"the new G inhibitors, C and . . . " "the effect of G inhibition by C" "inactivation of G by C" "C effects on G: . . . " "the effect of C on G activity in . . . "	rofecoxib / COX-2 tolcapone / COMT carbodiimides / thrombin Naloxone / beta-endorphin aspartame / acetylcholinesterase
11a	62	response to treatment	"G responses to C" "effect of C on G" "effect of G with C therapy" "influence of C on G response" "effects of C on G release"	cimetidine / Prolactin gossypol / LDH-X ribavirin / interferon_alpha_2b clofibrate / insulin amines / renin
11c	96	metabolism, secretion/uptake	"effect of G on C metabolism" "effects of G on C formation" "effect of G on the secretion of C" "control of G by C" "G stimulates C uptake" "[chemical] may reduce G concentration via C" "G stimulates C transport" "effect of C on G release"	dopamine / beta-endorphin cyclic_AMP / adrenomedullin omeprazole / intrinsic_factor retinoic_acid / c-jun phenylalanine / Insulin catecholamines / leptin calcium / Prolactin fenfluramine / growth_hormone
11d	23	binding (uptake/release)	"C release from G" "binding of C to G"b "C uptake by G" "enhancement of action of G by C" "controlled release of G by C"	iron / transferrin calcium / troponin_C potassium / HKT1 glucose / insulin-like_growth_factor_I polyurethane / IGF-1
13	18	modulation of expression, substrates	"C modulates [event] through G" "C binding to G" "C induced by G"	Metformin / SIRT1 cyanide / myeloperoxidase nitric_oxide / iNOS

Table A.2. Chemical-gene clusters from Figure 7.1.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / G)
			"C is a G substrate"	caffeine / cytochrome_p450_1a2
			"C mediates [event] by G"	superoxide / c-Src
14	24	receptor binding	"antagonist of the G C receptor"	tachykinin / NK1
			"effect of C receptors, G and ..."	steroid / pS2
			"[chemical] antagonism of a G C agonist"	dopamine / D-2
			"interaction of G with C receptors ..."	estrogen / DYX1C1
			"a new selective G agonist, C ..."	procaterol / beta_2-adrenoceptor
15	14	receptors	"G, a C receptor ..."	free_fatty_acid / GPR40
			"deletion of the C G gene ..."	adenosine / A1_receptor
			"G, a major C receptor ..."	somatostatin / SSTR4
			"the C domain of G"	zinc / SIP1
			"analysis of G C channels ..."	potassium / KCNQ2
16	16	receptor [subunit]	"the C carrier subunit (G) of ..."	acyl / NDUFAB1
			"increased expression of C receptor (G)"	benzodiazepine / PBR
			"the G subunit of the C receptor"	NMDA / GluN2B
			"human C receptor subunit (G)"	acetylcholine / CHRNA4
			"C receptor G subunits"	AMPA / GluR1
19	38	channels / transporters	"regulation of G transporters C and ..."	sterol / ABCG5
			"G is a C channel that modulates ..."	chloride / MOD-1
			"G C transporter expression"	glucose / GLUT4
			"C transporter, G"	glutamate / VGLUT1
			"distribution of G C channel subunits"	potassium / Kv4
20	15	synthase, dehydrogenase, reductase	"neuronal C synthase (G)"	nitric_oxide / nNOS
			"C transporter (G) polymorphism"	serotonin / 5-HTT
			"porcine C reductase (G)"	thiol / GILT
			"C dehydrogenase (G)"	Aldehyde / Ald4p

Table A.2. Chemical-gene clusters from Figure 7.1.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / G)
			"C synthase (G) gene"	5-aminolevulinate / ALAS1
21	10	transporters	"G, a C transporter, ..."	ribavirin / ENT1
			"low-affinity C cotransporter (G)"	sodium_glucose / SGLT2
			"a C binding protein, G"	methyl_CpG / Mecp2
			"C transfer protein (G) polymorphism"	Cholesteryl_ester / CETP
			"C binding protein (G)"	fatty_acid / hFABP
24	24	sequence, factor, moiety	"complete C sequence of G"	amino_acid / GSTM4
			"G is C exchange factor"	guanine_nucleotide / Rab3GEP
			"binding of [chemical] to the C moiety of G"	heme / cytochrome_P-450
			"structural analysis of the C finger of G"	zinc / THAP1
			"C binding domains of G"	nucleotide / CFTR
27	74	phosphorylation, kinases	"expression of receptor C kinase, G"	tyrosine / Trk
			"G receptor C kinases"	tyrosine / ErbB
			"[gene name] (G) C phosphorylation"	tyrosine / GIT1
			"G C phosphorylation pathway"	serine / STAT3
			"implication of G and C kinase in . . . "	creatine / esterase_D
29	13	phosphorylation, phosphatases	"G induces C phosphorylation"	tyrosine / Oncostatin_M
			"G C phosphatase"	tyrosine / Shp2
			"C phosphorylation sites on G"	Serine / IRS2
			"conserved C residues in G"	histidine / lipoxygenase
			"a critical C residue in G"	lysine / apolipoprotein_B-100
30	20	inhibition / activation (via phosphorylation?)	"[other chemical] inhibits C activation of G"	phenylephrine / phospholipase_A
			"efficacy of G C kinase inhibitors"	tyrosine / EGFR
			"surface of G C domain"	zinc / TFIIB
			"discovery of C G inhibitors"	glycine_hydrazide / CFTR
			"G induces rapid C phosphorylation"	tyrosine / Prolactin

Table A.3. Chemical-disease clusters from Figure 7.2.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / D)
1	13	prevents, reduces incidence	"C and [other drug] reduce [adverse event] after D" "C decreased levels of [substance] after D" "D of patients treated with C" "[women, men] receiving C to prevent D" "intravenous C reduces the incidence of D"	Isoflurane / cerebral_ischemia estrone / brain_injury triptans / coronary_spasm nevirapine / HIV-1_vertical_transmission magnesium / arrhythmias
2	20	inhibits growth / proliferation	"C significantly inhibited the growth of D" "C inhibits proliferation of D cells" "C inhibited [event(s)] in D cells" "C inhibited D growth" "C inhibits D growth in vitro"	celastrol / osteosarcoma Darbepoetin / hepatic_cancer NVP / RCC sorafenib / tumor Zebularine / acute_myeloid_leukemia
3	46	induction of effects in cells, esp. resistance; chemotherapy	"[event] induced by C in D cells" "C therapy for D" "C resistance in D" "D resistant to both C and [other drug]" "chemotherapy agents like C in D treatment"	fenretinide / neuroblastoma cisplatin / thoracic_malignancies Tamoxifen / breast_cancer imatinib / GIST doxorubicin / hepatocellular_carcinoma
6	15	treatment evaluations (esp. safety)	"C was measured in patients with D" "we evaluated the effects of C on D" "C is indicated for D" "C administered before/after D reduced [event]" "treatment of D with C"	Glutamic_acid / ischemic_stroke diphenidol / chronic_constriction_injury Bicillin_C-R / streptococcal_infections nicardipine / coronary_artery_occlusion sulfasalazine / juvenile_spondyloarthropathies
8g	125	treatment of disease (esp. evaluation of efficacy)	"C therapy for the treatment of D" "patients who received C for treatment of D" "D patients were treated with C" "effectiveness of C in D" "comparison of C and [other drug] in D"	indomethacin / PDA tigecycline / Acinetobacter_infections DMSO / amyloid_A_amyloidosis warfarin / atrial_fibrillation timolol / angle-closure_glaucoma
8h	80	treatment of disease (indication of efficacy)	"C may be useful for the treatment of D" "evaluate the protective efficacy of C in D"	OPC-18790 / congestive_heart_failure FTY720 / cerebral_ischemia

Table A.3. Chemical-disease clusters from Figure 7.2.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / D)
			"C is a promising treatment option for patients with D"	bosutinib / CML
			"C is approved for the treatment of D"	anidulafungin / intra-abdominal_abscesses
			"C is commonly prescribed for D"	Colchicine / gout
9	14	treatment of disease (prophylactic)	"C may be used for the prevention of D"	melatonin / premature_aging
			"in [children, patients] with D following C treatment"	MPH / ADHD
			"C reduces [event] [during, before] D"	thiazolidinedione_ciglitazone / pneumonia
			"C prevents [event] [during, in] D"	Mibefradil / atrial_tachycardia
			"C reduces the risk of D by X%"	raloxifene / vertebral_fractures
15	37	side effects (association)	"D associated with C therapy"	clozapine / tachycardia
			"the use of C has been associated with D"	moxalactam / thrombocytopenia
			"C intake was associated with D"	caffeine / shorter_nocturnal_sleep_duration
			"incidence of D in patients receiving C"	oxaliplatin / hypersensitivity_reaction
			"D occurred after C"	alfentanil / hypotension
16	67	side effects (causal implications), studies inducing effect	"administration of C resulted in D"	vincristine / thrombocytopenia
			"C induces D"	Taxol / myalgias
			"D was induced by administration of C"	lidocaine / Hypotension
			"D was/were induced by infusion of C"	ouabain / Cardiac_arrhythmias
			"patient developed D after receiving C"	ceftaroline / eosinophilic_pneumonia
18	12	potential biomarkers	"C levels of D patients were significantly [lower/higher] . . . "	homocycteine / hyperthyroid
			"monitoring of C in D rats"	homocysteine / hypertensive
			"reduced C in D subjects"	selenium / asthmatic
			"significant elevations of C in D subjects"	leucine / MSUD
			"effect of C on [biomarker level / event] in D patients"	clozapine / schizophrenic
19	15	potential biomarkers	"effect of C supplementation in D"	vitamin_D3 / Autism_Spectrum_Disorder
			"we studied the effect of C on D"	rosiglitazone / angiogenesis
			"C was well tolerated in [patient group] with D"	tolterodine / incontinence
			"blood C concentrations in patients with D"	vitamin_C / diabetes_mellitus

Table A.3. Chemical-disease clusters from Figure 7.2.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / D)
			"examine the C status of our D patients"	magnesium / chronic_ambulatory_peritoneal_dialysis
20	63	levels associated with disease risk / progression	"high C levels are associated with increased risk of D"	cholesterol / coronary_heart_disease
			"C implicated in D"	bisphosphonates / osteonecrosis
			"effect of D on serum C levels"	testosterone / prostate_cancer
			"patients with D and increased C concentrations"	triglyceride / unstable_angina
			"C has been implicated in the pathogenesis of D"	Serotonin / migraine
			"C intake may be associated with [lower/higher] risk of D"	PUFA / colorectal_neoplasia
			"C supplementation and incidence of D: . . . "	beta-carotene / cancer
21	13	changed incidence / risk	"C use was associated with [increased/decreased] risk of D"	Warfarin / ICH
			"C reduce(s) the risk of D"	Bisphosphonates / osteoporotic_fractures
			"C may reduce the incidence of D in . . . "	Eicosapentaenoic_acid / cardiovascular_disease
			"C was associated with a [lower/higher] risk of D"	Preconception_O3 / GDM
			"relation of C to risk of D"	cholesterol / coronary_heart_disease
24	22	inhibits, suppresses	"C inhibited [other event] in D"	Ki23057 / gastric_tumours
			"the D action of C"	diltiazem / hypotensive
			"C suppresses D through [mechanism]"	Evodiamine / hyperalgesia
			"influence of C on D development"	histamine / seizure
			"C significantly suppressed D"	AS1069562 / allodynia
26	48	inhibited / blocked disease progression	"the effects of C on the progression of D"	minocycline / encephalopathy
			"C may protect against D"	Eicosapentaenoic_acid / atherosclerotic_disease
			"C blocked D in organ culture"	phenethyl_caffeiate / hyperplasia
			"C antagonized [other drug-induced] D"	procyclidine / seizures
			"C attenuates D in mice"	Simvastatin / pulmonary_fibrosis
			"C ameliorated D by [mechanism]"	EGB / endothelial_dysfunction
			"C alleviates D in [disease model]"	Propentofylline / hypersensitivity
28	17	preventive effects evaluated	"examine the effects of C on D"	metformin / cytotoxicity
			"study was carried out to evaluate the effect of C on D"	atorvastatin / inflammation

Table A.3. Chemical-disease clusters from Figure 7.2.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (C / D)
			"investigated possible beneficial effects of C on D"	AdCbl / atopic_dermatitis
			"to assess the effect of C on D"	nebivolol / endothelial_dysfunction
			"C effective for the prevention of D"	dronedarone / atrial_fibrillation
30	23	reduced, abolished, prevented	"C prevents D"	Itraconazole / fungal_infections
			"C, a [description], prevented D"	AMD3100 / anxiety_behaviors
			"C is beneficial in D"	lithium / tauopathies
			"D was reduced by C"	gabapentin / Pain
			"C was effective in reducing D"	buspirone / overall_anxiety_symptoms

Table A.4. Gene-disease clusters from Figure 7.3.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G / D)
2h	44	therapeutic effects, esp. drug sensitivity, resistance	"G and response to [drug] in patients with D" "G resistance in patients with D" "serum G levels are associated with D" "G sensitivity in D" "comparison of G and [other drug] for detection of D"	TCF7L2 / type_2_diabetes insulin / systemic_lupus_erythematosus leptin / hepatic_steatosis insulin / hypertension cardiac_troponin_I / ischemic_myocardial_injury
2j	33	influences disease treatment (some adjuvant therapies)	"the use of G in the treatment of D" "D in patients treated with G" "effect of G on [event] in D patients" "G therapy in patients with D" "efficacy of G in D"	parathyroid_hormone / osteoporosis interferon_alpha_2b / Acute_renal_failure prolactin / systemic_lupus_erythematosus Erythropoietin / chronic_renal_failure S-1 / colorectal_cancer
3	13	therapy, trial, treatment	"G gene therapy of D" "study of G in D" "trial of G in the treatment of D" "relationship between G and [substance] in D patients" "G treatment for D"	Connexin_26 / bladder_cancer epidermal_growth_factor / gastric_carcinoma VP-16 / chronic_granulocytic_leukemia apolipoprotein_H / stroke Epoetin / anaemia
4	24	protein causes change in disease status	"injected G induces D" "G promotes D" "regulation of [event] by G in D" "G inhibits D" "G exacerbates D"	IL-1 / anorexia VEGF-D / metastasis TDP-43 / frontotemporal_lobar_degeneration High-mobility_group_box_1 / ulcer_healing VDUP1 / bacteremic_shock
5	12	levels / expression in disease	"G levels in D patients" "expression of G in D" "increased G levels in patients with D" "[regulation/function] of G system in D" "G level in D"	Interleukin-6 / headache SFRP4 / primary_serous_ovarian_tumours thyroglobulin / nontoxic_goiter interleukin-6 / stroke C-reactive_protein / atopic_dermatitis
6	28	levels / expression in disease	"G levels in patients with D" "G levels in D patients"	interleukin-6 / glomerulonephritis Interleukin-2 / multiple_sclerosis

Table A.4. Gene-disease clusters from Figure 7.3.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G / D)
			"effects of [drug] on G in D patients"	insulin / hypertensive
			"serum G levels in D"	E-selectin / Kawasaki_disease
			"expression of G in D"	E-cadherin / carcinomas
7	26	biomarkers, diagnostic	"G is a robust diagnostic biomarker for D"	TLE1 / synovial_sarcomas
			"G is an independent predictor of D"	Proinsulin / coronary_heart_disease
			"G as an indicator of D in patients with . . . "	Plasma_hyaluronidase / atherosclerosis
			"prognostic significance of G in D patients"	TGFbeta-1 / breast_cancer
			"effects of [situation/event] on G levels in D"	chromogranin-A / neuroendocrine_tumors
			"G is a potential marker of D"	SERPINA3 / preeclampsia
8	11	role in pathogenesis	"association of G with [event] in patients with D"	FCGR2A / rheumatoid_arthritis
			"effects of G on D"	interleukin-5 / acute_myeloid_leukemias
			"role of G in the development of D"	IL-4 / transplant_arteriosclerosis
			"role of G in the pathogenesis of D"	leptin / thyroid_cancer
			"a novel gene, G, is associated with D"	THSD7A / obesity
9	24	role in disease course / pathogenesis	"clinical impact of circulating G in D"	miR-18a / oesophageal_squamous_cell_carcinoma
			"G attenuates D"	Wnt5a / pulmonary_arteriolar_remodeling
			"G predicts [event] in patients with D"	LTBP2 / acute_dyspnoea
			"evidence for role of G in D"	BRCA1 / gastric_cancer
			"G: the link between D and [other disease]"	HMGB1 / diabetes_mellitus
10	32	inhibitors used as therapies	"G inhibitors in D: . . . "	ACE / aortic_stenosis
			"D with G mutation(s)"	TARDBP / amyotrophic_lateral_sclerosis
			"response to G inhibitors in patients with D"	EGFR / squamous_cell_carcinoma
			"G testing and management of D"	EGFR / NSCLC
			"G gene amplification in D"	c-erbB-2 / nasopharyngeal_carcinoma
12	17	drug targets (esp. cancer)	"G signaling in D cells"	Akt / colon_cancer
			"G inhibitors in the treatment of D"	MEK1/2 / malignancies
			"G as a strategic target in D therapy"	ErbB1 / breast_cancer

Table A.4. Gene-disease clusters from Figure 7.3.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G / D)
			"G: an attractive target for D therapy"	Angiopoietin-2 / tumor
			"[drug]: a C inhibitor for the treatment of D"	tumor_necrosis_factor_alpha / rheumatoid_arthritis
13	26	evaluation of role of mutations in disease	"G mutations in D"	KRAS / lung_adenocarcinoma
			"mutations in G in D"	GUSB / mucopolysaccharidosis_VII
			"characterization of G expression in D"	MUC1 / papillary_thyroid_carcinoma
			"G mutations are associated with [event] in D"	KRAS / colorectal_cancer
			"role of G in D development"	RSK2 / osteosarcoma
14	91	causal mutations	"mutation of G in a patient with D"	STK11 / Peutz-Jeghers_syndrome
			"G mutation is associated with D"	MTHFR / arterial_stroke
			"novel mutation in G gene associated with D"	MYH7 / distal_myopathy
			"characterization of G mutations causing D"	GALC / Krabbe_disease
			"mutations of the G gene in patients with D"	COL1A2 / osteogenesis_imperfecta
			"D: a novel G mutation . . . "	CISD2 / Wolfram_syndrome
			"D: novel G mutations and . . . "	NPC1 / Niemann-Pick_type_C_disease
			"the recurrent mutation of G in C patients"	BRCA1 / breast_cancer
			"G mutations can cause D"	HIBCH / Leigh-like_disease
15	13	levels, concentrations, expression	"G levels in patients with D"	renin / thoracic_neuroblastoma
			"serum G concentrations in D"	leptin / hyperinsulinemia
			"G expression in D cell lines"	TIMP-1 / prostate_tumor
			"diagnostic value of G in D patients"	interleukin_17 / lung_cancer
			"prognostic relevance of G in D"	CCN3 / Ewing_sarcoma
17	12	levels, overexpression	"serum G concentrations in patients with D"	erythropoietin / anemia
			"serum G level in patients with D"	thyroglobulin / subacute_thyroiditis
			"G overexpression in D"	cyclin_D3 / follicular_thyroid_carcinoma
			"G is overexpressed in D"	FOXG1 / hepatoblastoma
			"G expression in D patients"	SPARC / pancreatic_cancer
18	43	expression, mutations correlated	"presence of G gene mutation in D patients"	BRAF / melanoma

Table A.4. Gene-disease clusters from Figure 7.3.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G / D)
		with disease	"frequency of G mutations in D"	PTEN / thyroid_cancer
			"association of D with G mutations"	PDH / cerebral_dysgenesis
			"association of G expression with D"	FcRn / lung_abnormalities
			"correlation between G expression and [event] in D"	COX-2 / colon_cancer
19	32	gene expression, regulation	"down-regulation of G in D cells"	E-cadherin / breast_cancer
			"expression of G mRNA in D"	CerbB-2 / nasopharyngeal_carcinomas
			"mRNA expression of G in patients with D"	KCNQ1 / long_QT_syndrome_type_1_and_2
			"D cells expressing G"	P-gp / acute_myeloid_leukemia
			"regulation of G expression in D cells"	CYP1A1 / medulloblastoma
21	66	gene expression in cell lines	"G expression in D"	c-mpl / hematologic_disorders
			"G expression in patients with D"	trypsinogen-1 / ulcerative_colitis
			"analysis of G expression in D"	SLC34A2 / ovarian_tumors
			"effects of G on D cells"	p53 / hepatocellular_carcinoma
			"G expression in D cells"	MMP2 / prostate_cancer
22	28	tumor suppressor genes	"G as a D suppressor"	Caspase-2 / tumour
			"G acts as a D suppressor"	ECRG4 / tumor
			"the gene G is a functional D suppressor"	GADD45G / tumor
			"G, a novel D suppressor"	SynCAM / tumor
			"G: a mediator of D"	P-glycoprotein / melanoma_invasion
26	26	polymorphism	"association of variants of G with D"	factor_V_Leiden / thrombosis
			"association of the G polymorphisms with D"	interleukin-18 / type_1_diabetes
			"genetic polymorphisms at G are associated with D"	SIRT1 / carotid_atherosclerosis
			"mutations in the G gene in patients with D"	P-protein / encephalopathy
			"G polymorphisms are associated with D"	Chromogranin_A / hypertensive_renal_disease
27	28	polymorphism	"association of G gene polymorphism with D"	vascular_endothelial_growth_factor / colon_cancer
			"polymorphism of G in D"	angiotensin-converting_enzyme / sarcoidosis
			"mutation of the G gene in D"	endothelin-3 / Waardenburg-Hirschsprung_disease

Table A.4. Gene-disease clusters from Figure 7.3.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G / D)
			"G polymorphism is associated with D"	tumor_necrosis_factor_a / cystic_fibrosis
			"mutation in the G gene in a family with D"	connexin_32 / Charcot-Marie-Tooth_neuropathy
29	20	promotes progression (cancers)	"G promotes D cell invasion"	DLK1 / lung_cancer
			"G promotes D cell proliferation"	CD97 / gastric_cancer
			"expression of G in [disease] correlates with D"	Apaf-1 / lymph_node_metastasis
			"G promotes D progression by . . . "	HDAC6 / hepatocellular_carcinoma
			"G expression is associated with D in [disease]"	Gli-1 / lymph_node_metastasis
30	25	overexpression associated with disease (cancers)	"regulation of G gene expression in D"	CD44 / neuroblastoma
			"prognostic value of G in D"	Gli-1 / gastric_cancer
			"secretion of G by D in vitro"	cathepsin_B / gliomas
			"G overexpression in D"	TRIB1 / acute_myeloid_leukemia
			"correlation between G expression and D"	p27Kip1 / esophageal_squamous_cell_carcinoma

Table A.5. Gene-gene clusters from Figure 7.4.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G1 / G2)
1	90	cell populations	"G1 induction of human G2"	C5a / interleukin_1
			"increased induction of G2 in G1 lymphocytes"	CD8 / interferon-gamma
			"G1 + G2 T-cell population"	CD25 / Foxp3
			"G2 induction of G1"	NfkappaB / Interleukin-1beta
			"G1 expression [on, in] G2 T-cells"	CD161 / CD8
			"an enriched G1 + G2 T-cell subset"	CD4 / CD8beta
			"G1-dependent G2 activation"	ERK / CREB
2	14	cell populations, regulation	"regulation of G2 expression by G1"	SOX10 / MITF
			"G1 induces G2 gene transcription"	TNF-alpha / MUC1
			"regulation of G2 by G1"	RECK / matrix_metalloproteinase-9
			"G2 expression in the G1 + cells"	CD34 / Bcl-2
			"G1 / G2 ratio"	CD39 / CD8
6	39	cell populations, protein production / gene expression	"G1 production by G2 + T cells"	IL-17A / CD146
			"G1 producing G2 + T cells"	IL-10 / CD8
			"G1 signaling in G2 + T cells"	IFN-gamma / CD4
			"G1 expression on G2 + T cells"	CXCR3 / CD8
			"the role of G1 in the function of G2 + T cells"	CD28 / CD25
7	15	inhibits / induces expression	"G1 induces G2 expression"	Fos / Neurotensin
			"G1 inhibits G2 expression"	IL-15 / IL-7Ra
			"effect of G2 on G1 production"	MMP-9 / calcitonin-gene-related_peptide
			"G1 secretion in G2 cells"	cholecystokinin / STC-1
			"G1 induced G2 production"	TNF-alpha / TARC
10	76	binding, regulation of activity	"G1 binds G2"	HJURP / CENP-A
			"G2 interaction with G1"	Bcl-xL / Clusterin
			"G1 is a receptor for G2"	CD96 / CD155
			"G1 binding to G2"	Haptoglobin / apolipoprotein_A-I
			"G1 mediates activation of G2"	Bcl10 / NF-kappaB

Table A.5. Gene-gene clusters from Figure 7.4.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G1 / G2)
13	14	enhances response (esp. hormones)	"G1 enhances [event] via G2"	Glypican-4 / insulin_receptor
			"changes in the G1 response to G2"	prolactin / thyrotropin-releasing_hormone
			"G1 and G2 responses to [event]"	Prolactin / TRH
			"G1 in G2 receptor signaling"	Fc_gamma_RI / p72syk
			"exaggerated G2 response of G1"	thyrotropin-releasing_hormone / prolactin
14	67	activation, stimulation, signaling	"G2 activates [protein] via G1"	fucosyltransferase_1 / Calreticulin
			"G1 stimulates G2"	Akt / SREBP1c
			"G1 modulates G2 signaling"	Hsp27 / p53
			"G2 stimulates G1 expression"	EGFR / MUC1
			"G1 induces phosphorylation of G2"	Thrombopoietin / STAT5
16	23	activation, targeting	"function of G2 in G1 receptor activation"	TNFR1 / Ubc13
			"G2 promotes [event] by targeting G1"	EPB41L3 / miRNA-223
			"G1 phosphorylation by G2"	NuMA / CDK1
			"role of G1 in the activation of G2"	PP4 / JNK-1
			"regulation of G1 expression by G2"	FGF8 / androgen_receptor
17	13	affects production (mostly induces)	"G2 induces the production of G1"	IgG1 / IL-27
			"[protein] stimulates G2 production via G1"	ERK1/2 / granulocyte_colony-stimulating_factor
			"regulation of G2 production by G1"	IFN_gamma / IL-18
			"downregulation of G2 by G1"	miR-25 / mitochondrial_calcium_uniporter
			"enhancement of G2 by G1"	TNF-alpha / IFN-gamma
21	28	induces expression / production	"G2 induces G1 production"	beta-defensin-2 / Tat
			"G1 modulates G2 expression"	Stat3 / heat_shock_27kDa_protein
			"induction of G1 expression by G2"	iNOS / IL-1beta
			"G2 upregulates G1 expression"	p16INK4a / p33ING1b
			"G1 stimulates G2 secretion in [cell type] cells"	Angiotensin_II / endothelin-1
22	56	induces release / production	"G1 induces G2 expression"	CXCL12 / connective_tissue_growth_factor
			"G2 stimulates G1 secretion"	atrial_natriuretic_peptide / Thrombin

Table A.5. Gene-gene clusters from Figure 7.4.

Cluster Number	Cluster Size	Theme	Selected Descriptive Patterns	Entity Pair with Pattern (G1 / G2)
			"G1 stimulates G2 release"	Bradykinin / tissue_plasminogen_activator
			"G2 stimulates G1 production"	MCP-1 / Angiotensin_II
			"effect of G2 on G1 secretion"	renin / neuropeptide_Y
24	62	signaling, receptor binding	"G2 signaling via G1"	SMOC-1 / TGF-beta
			"G1 / G2 costimulatory interactions"	ICAM-1 / LFA-1
			"coactivator G1 in G2 transcriptional activation"	CBP / p53
			"G2 G1 signaling"	TCF / beta-catenin
			"the G2 G1 receptor"	TNF / p55
			"binding of G2 to the G1 receptor"	interleukin-1 / interleukin-18
25	26	same or related protein: abbreviations	"G1 (G2) inhibitor"	mammalian_target_of_rapamycin / mTOR
			"expression of G1 (G2) protein"	pentraxin_3 / PTX3
			"G2 (G1) activity"	PON1 / paraoxonase-1
			"G2 (G1) expression"	AURKA / Aurora_kinase_A
			"G1 / G2 complexes"	PAI-1 / vitronectin
28	26	regulation of expression / activity	"the roles of G1 / G2 in [event]"	MMP-2 / TIMP-2
			"G2 (G1) expression"	M-CSF / macrophage_colony-stimulating_factor
			"binding of G1 / G2 proteins"	NF-kappa_B / Rel
			"G2 regulates G1 activity"	RhoA / Shp-2
			"synergistic effect of G1 / G2"	IL-6 / BSF-2
30	28	regulation of expression / activity	"upregulation of G2 activity by G1"	CD28 / interleukin-4
			"regulation of G1 expression by G2"	TNF-alpha / TGF-beta
			"G1 regulation of G2"	miR-133b / Connective_Tissue_Growth_Factor
			"G2 regulation by the G1 pathway"	JNK / ATF2
			"prognostic significance of G1, G2, . . . "	bcl-2 / PCNA

Bibliography

1. Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. [7, 3.4.2]

2. J Thomas August, Ferid Murad, MW Anders, Joseph T Coyle, and Albert P Li. *Drug-Drug Interactions: Scientific and Regulatory Perspectives*. Academic Press, 1997. [2.3]

3. Matthijs L Becker, Marjon Kallewaard, PW Caspers, Loes E Visser, HG Leufkens, and BH Stricker. Hospitalisations and emergency department visits due to drug-drug interactions: a literature review. *Pharmacoepidemiology and Drug Safety*, 16(6):641, 2007. [2.1]

4. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003. [3.2.1]

5. Jacob Bien and Robert Tibshirani. Hierarchical clustering with prototypes via minimax linkage. *Journal of the American Statistical Association*, 106(495):1075–1084, 2011. [5.4.2, 6.4.1]

6. Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Suppl 1):D267–D270, 2004. [4.1, 4.3]

7. Danushka Tarupathi Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relational duality: unsupervised extraction of semantic relations between entities

on the web. In *Proceedings of the 19th International Conference on World Wide Web*, pages 151–160. ACM, 2010. [3.2.2]

8. Evan E Bolton, Yanli Wang, Paul A Thiessen, and Stephen H Bryant. PubChem: integrated platform of small molecules and biological activities. *Annual Reports in Computational Chemistry*, 4:217–241, 2008. [4.3]

9. Andrew P Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. [6.5.1]

10. Paul Broadhurst and Anthony W Nathan. Cardiac arrest in a young woman with the long QT syndrome and concomitant astemizole ingestion. *British Heart Journal*, 70(5):469–470, 1993. [2.2, 2.1]

11. Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. The extraction of pharmacogenetic and pharmacogenomic relations–a case study using PharmGKB. In *Pacific Symposium on Biocomputing*, pages 376–387, 2012. [6]

12. Hao Chen and Burt M Sharp. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics*, 5(1):147, 2004. [2.3]

13. Georges Cheymol. Effects of obesity on pharmacokinetics. *Clinical Pharmacokinetics*, 39(3):215–231, 2000. [2.2]

14. Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. [3.2.2]

15. Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71, 2005. [2.3]

16. Trevor Cohen and Dominic Widdows. Empirical distributional semantics: methods and biomedical applications. *Journal of Biomedical Informatics*, 42(2):390–405, 2009. [3]

17. International Warfarin Pharmacogenetics Consortium et al. Estimation of the warfarin dose with clinical and pharmacogenetic data. *The New England Journal of Medicine*, 360(8):753, 2009. [2.2]

18. A Coulet, RB Altman, MA Musen, and NH Shah. Integrating heterogeneous relationships extracted from natural language sentences. *Proceedings of the Bio-ontologies SIG, ISBM*, pages 9–10, 2010. [2.3, 5]

19. Adrien Coulet, Yael Garten, Michel Dumontier, Russ B Altman, Mark A Musen, and Nigam H Shah. Integration and publication of heterogeneous text-mined relationships on the Semantic Web. *Journal of Biomedical Semantics*, 2(S-2):S10, 2011. [2.4, 4.2, 4.2]

20. Adrien Coulet, Nigam H Shah, Yael Garten, Mark Musen, and Russ B Altman. Using text to build semantic networks for pharmacogenomics. *Journal of Biomedical Informatics*, 43(6):1009–1019, 2010. [2.3, 2.3.1, 5, 2.4, 4.2, 4.2.1, 6, 6.8]

21. Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: rationale, evaluation and approaches. *Natural Language Engineering*, 15(Special Issue 04):i–xvii, 2009. [3.2.2]

22. Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. *Recognizing textual entailment: models and applications*. Morgan & Claypool Publishers, 2013. [3.2.2]

23. Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006. [4]

24. Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008. [6.2.2, 6.2.2, 6.2.3]

25. Marie-Catherine De Marneffe and Christopher D Manning. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008. [6.2.2, 6.9, 2]

26. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. [3, 3.2.1, 3.2.1, 3, 5.1, 6.7]

27. Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98. ACM, 2003. [(document), 5.2, 5.2, 5.2.1, 5.2.1, 5.2.1, 5.2.1, 5.2, 5.3.2, 6.1, 3]

28. Susan T Dumais. LSI meets TREC: a status report. In *Proceedings of the First Text Retrieval Conference, TREC1*, pages 137–152, 1993. [3.2.1]

29. Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906. Association for Computational Linguistics, 2008. [3.2.1]

30. Kenneth E Ferslew, Andrea N Hagardorn, OC Harlan, and WF McCormick. A fatal drug interaction between clozapine and fluoxetine. *Journal of Forensic Sciences*, 43:1082–1085, 1998. [2.1]

31. John R Firth. *A synopsis of linguistic theory, 1930-1955*. Blackwell, 1957. [3]

32. Centers for Disease Control and Prevention. National Hospital Ambulatory Medical Care Survey 2009. http://www.cdc.gov/nchs/ahcd.htm, 2010. [Online; accessed 16-Jan-2016]. [2.1]

33. National Center for Health Statistics. Health, United States, 2011: with special feature on socioeconomic status and health. http://www.cdc.gov/nchs/data/hus/hus11.pdf, 2012. [2.1]

34. Lisa Francis, Eduardo Bonilla, Ekaterina Soforo, Hom Neupane, Hassan Nakhla, Christine Fuller, and Andras Perl. Fatal toxic myopathy attributed to propofol, methylprednisolone, and cyclosporine after prior exposure to colchicine and simvastatin. *Clinical Rheumatology*, 27(1):129–131, 2008. [2.1]

35. Carol Friedman, Philip O Alderson, JH Austin, James J Cimino, and Stephen B Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161, 1994. [6.8]

36. Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx: relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007. [4.2.11, 6]

37. Yael Garten, Adrien Coulet, and Russ B Altman. Recent progress in automatically extracting information from the pharmacogenomics literature. *Pharmacogenomics*, 11(10):1467–1489, 2010. [2.3]

38. Yoav Goldberg and Omer Levy. word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. Technical Report arXiv:1402.3722v1, Bar Ilan University, February 2014. [3.5.2]

39. Margaret Jean Hall, Carol J DeFrances, Sonja N Williams, Aleksandr Golosinskiy, Alexander Schwartzman, et al. National Hospital Discharge Survey: 2007 Summary. *Natl Health Stat Report*, 29(29):1–20, 2010. [2.1]

40. Ada Hamosh, Alan F Scott, Joanna S Amberger, Carol A Bocchini, and Victor A McKusick. Online Mendelian Inheritance in Man (OMIM): a knowledge

base of human genes and genetic disorders. *Nucleic Acids Research*, 33(Suppl 1):D514–D517, 2005. [1, 6]

41. Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415. Association for Computational Linguistics, 2004. [3.2.2]

42. Saeed Hassanpour and Curtis P Langlotz. Information extraction from multi-institutional radiology reports. *Artificial Intelligence in Medicine*, 2015. [4.4.1]

43. Trevor Hastie, Jerome Friedman, and Robert Tibshirani. *The Elements of Statistical Learning*, chapter 16: Random Forests. Springer Series in Statistics; Springer, Berlin, 2009. [2.3.3, 2.3.3, 2.6]

44. Robert Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence*, pages 43–56, 1994. [3.4.2]

45. Jesús Herrera, Anselmo Penas, and Felisa Verdejo. Textual entailment recognition based on dependency analysis and WordNet. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment: First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, pages 231–239. Springer, 2006. [4.2.11]

46. Micheal Hewett, Diane E Oliver, Daniel L Rubin, Katrina L Easton, Joshua M Stuart, Russ B Altman, and Teri E Klein. PharmGKB: the pharmacogenetics knowledge base. *Nucleic Acids Research*, 30(1):163–165, 2002. [4.2.2]

47. Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl 1):S1, 2005. [6.8]

48. SM Huang, R Temple, DC Throckmorton, and LJ Lesko. Drug interaction studies: study design, data analysis, and implications for dosing and labeling. *Clinical Pharmacology & Therapeutics*, 81(2):298–304, 2007. [2.2]

49. IFN Hung, AKL Wu, VCC Cheng, BSF Tang, KW To, CK Yeung, PCY Woo, SKP Lau, BMY Cheung, and KY Yuen. Fatal interaction between clarithromycin and colchicine in patients with renal insufficiency: a retrospective study. *Clinical Infectious Diseases*, 41(3):291–300, 2005. [2.1]

50. Lars Juhl Jensen, Jasmin Saric, and Peer Bork. Literature mining for the biologist: from information retrieval to biological discovery. *Nature Reviews Genetics*, 7(2):119–129, 2006. [1]

51. Antonio Jimeno, Ernesto Jimenez-Ruiz, Vivian Lee, Sylvain Gaudan, Rafael Berlanga, and Dietrich Rebholz-Schuhmann. Assessment of disease named entity recognition on a corpus of annotated sentences. *BMC Bioinformatics*, 9(Suppl 3):S3, 2008. [4.1]

52. William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(189-206):1, 1984. [3.4.2]

53. Michael N Jones and Douglas JK Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114(1):1, 2007. [3.4.1]

54. Kenneth Jung, Paea LePendu, Srinivasan Iyer, Anna Bauer-Mehren, Bethany Percha, and Nigam H Shah. Functional evaluation of out-of-the-box text-mining tools for data-mining tasks. *Journal of the American Medical Informatics Association*, 22(1):121–131, 2015. [4.1]

55. Pentti Kanerva. *Sparse Distributed Memory*. MIT press, 1988. [3.4]

56. Pentti Kanerva, Jan Kristofersson, and Anders Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, volume 1036, 2000. [3.4]

57. Irene L Katzan and Richard A Rudick. Time to integrate clinical and research informatics. *Science Translational Medicine*, 4(162):162fs41, 2012. [1]

58. BG Katzung, SB Masters, and AJ Trevor. *Basic & Clinical Pharmacology*, chapter 2: Drug Receptors and Pharmacodynamics. Lange Medical Books/McGraw Hill New York, 2009. [2.3]

59. J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. GENIA corpus: a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–i182, 2003. [6.8]

60. Yoon Kim. Convolutional neural networks for sentence classification. *arXiv:1408.5882*, 2014. [3.2.2]

61. Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003. [4]

62. TE Klein, JT Chang, MK Cho, KL Easton, R Fergerson, M Hewett, Z Lin, Y Liu, S Liu, DE Oliver, et al. Integrating genotype and phenotype information: an overview of the PharmGKB project. *Pharmacogenomics Journal*, 1(3):167–170, 2001. [1, 4.2.2, 6.9, 8, 8.1.2]

63. Stanley Kok and Pedro Domingos. Extracting semantic networks from text via relational clustering. In *Machine Learning and Knowledge Discovery in Databases*, pages 624–639. Springer, 2008. [3.2.2]

64. Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, 7(Suppl 1):S1, 2015. [4.1]

65. Martin Krallinger, Miguel Vazquez, Florian Leitner, David Salgado, Andrew Chatr-aryamontri, Andrew Winter, Livia Perfetto, Leonardo Briganti, Luana

Licata, Marta Iannuccelli, et al. The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. *BMC Bioinformatics*, 12(Suppl 8):S3, 2011. [4.1]

66. Thomas K Landauer and Susan T Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211, 1997. [3.2.1, 3]

67. Curtis P Langlotz. RadLex: A new method for indexing online educational materials. *Radiographics*, 26(6):1595–1597, 2006. [4.4.2]

68. Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv:1405.4053*, 2014. [3.2.2]

69. Robert Leaman and Graciela Gonzalez. BANNER: an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, volume 13, pages 652–663, 2008. [4.1, 6.2.1, 6.9]

70. Ulf Leser and Jörg Hakenberg. What makes a gene name? Named entity recognition in the biomedical literature. *Briefings in Bioinformatics*, 6(4):357–369, 2005. [6.2.1, 6.9]

71. Omer Levy and Yoav Goldberg. Dependency based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 302–308, 2014. [3.5.2, 4.3.1]

72. Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014. [3.5.2, 3.8]

73. Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies (NAACL HLT 2015)*, Denver, CO, 2015. [3.2.2]

74. Jiexun Li, Zhu Zhang, Xin Li, and Hsinchun Chen. Kernel-based learning for biomedical relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5):756–769, 2008. [6]

75. Anthony M Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, and Jurgen Del-Favero. BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biology*, 12(6):R57, 2011. [6]

76. Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics, 1998. [3.2.1]

77. Dekang Lin and Patrick Pantel. DIRT – discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328. ACM, 2001. [3.2.2, 6.1, 6.7]

78. Donald A Lindberg, Betsy L Humphreys, and Alexa T McCray. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291, 1993. [4.1]

79. Kaihong Liu, William R Hogan, and Rebecca S Crowley. Natural language processing methods and systems for biomedical ontology learning. *Journal of Biomedical Informatics*, 44(1):163–179, 2011. [4.1]

80. Henry J Lowe, Todd A Ferris, Penni M Hernandez, and Susan C Weber. STRIDE–An integrated standards-based translational research informatics platform. In *AMIA Annual Symposium Proceedings*, volume 2009, page 391. American Medical Informatics Association, 2009. [4.4.1]

81. Zhiyong Lu. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011:baq036, 2011. [1]

82. Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996. [3.2.1]

83. Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33(Suppl 1):D54–D58, 2005. [4.3]

84. Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to Information Retrieval*. Cambridge University Press, 2008. [3.2, 2]

85. Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014. [4.3.2]

86. Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co., 2010. [4.2.3]

87. Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. Simple algorithms for complex relation extraction with applications to biomedical IE. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 491–498. Association for Computational Linguistics, 2005. [6]

88. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013. [3.2.1, 3.2.1, 3.5, 4.3.1]

89. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013. [3.2.2]

90. Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009. [6.1]

91. Michael Mohler, Razvan Bunescu, and Rada Mihalcea. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 752–762. Association for Computational Linguistics, 2011. [4.2.11]

92. Brian P Monahan, Clifford L Ferguson, Eugene S Killeavy, Bruce K Lloyd, James Troy, and Louis R Cantilena. Torsades de pointes occurring in association with terfenadine use. *Journal of the American Medical Association*, 264(21):2788–2790, 1990. [2.1]

93. Alexander A Morgan, Zhiyong Lu, Xinglong Wang, Aaron M Cohen, Juliane Fluck, Patrick Ruch, Anna Divoli, Katrin Fundel, Robert Leaman, Jörg Hakenberg, et al. Overview of BioCreative II gene normalization. *Genome Biology*, 9(Suppl 2):S3, 2008. [4.1]

94. Cristiano Soares Moura, Francisco Assis Acurcio, and Najara Oliveira Belo. Drug-drug interactions associated with length of stay and cost of hospitalization. *Journal of Pharmacy & Pharmaceutical Sciences*, 12(3):266–272, 2009. [2.1]

95. Tsendsuren Munkhdalai, Meijing Li, Khuyagbaatar Batsuren, Hyeon Ah Park, Nak Hyeon Choi, and Keun Ho Ryu. Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. *Journal of Cheminformatics*, 7(Suppl 1):S9, 2015. [4.1]

96. Jessica J Nadler and Gregory J Downing. Liberating health data for clinical research applications. *Science Translational Medicine*, 2(18):18cm6, 2010. [1]

97. Pertti J Neuvonen, Mikko Niemi, and Janne T Backman. Drug interactions with lipid-lowering drugs: mechanisms and clinical relevance. *Clinical Pharmacology & Therapeutics*, 80(6):565–581, 2006. [2.2]

98. U.S. National Library of Medicine. MEDLINE Citation Counts by Year of Publication. https://www.nlm.nih.gov/bsd/medline_cit_counts_yr_pub.html, 2014. [Online; accessed 16-Jan-2016]. [1]

99. U.S. National Library of Medicine. Number of Titles Currently Indexed for Index Medicus and MEDLINE on PubMed. https://www.nlm.nih.gov/bsd/num_titles.html, 2015. [Online; accessed 16-Jan-2016]. [1]

100. Haruhiro Okuda, Takahito Nishiyama, Kenichiro Ogura, Sekio Nagayama, Kazumasa Ikeda, Shuji Yamaguchi, Yoshimasa Nakamura, Yasuro Kawaguchi, and Tadashi Watabe. Lethal drug interactions of sorivudine, a new antiviral drug, with oral 5-fluorouracil prodrugs. *Drug Metabolism and Disposition*, 25(2):270–273, 1997. [2.1]

101. Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007. [3.2.1]

102. Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv:1404.5367*, 2014. [3.2.2]

103. Jeffrey Pennington, Richard Socher, and Christopher D Manning. GLOVE: Global vectors for word representation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014. [3.5.2]

104. Bethany Percha and Russ B Altman. Inferring the semantic relationships of words within an ontology using random indexing: applications to pharmacogenomics. In *AMIA Annual Symposium Proceedings*, pages 1123–1132. American Medical Informatics Association, 2013. [6, 4, 6.10]

105. Bethany Percha and Russ B Altman. Informatics confronts drug–drug interactions. *Trends in Pharmacological Sciences*, 34(3):178–184, 2013. [2, 6]

106. Bethany Percha and Russ B Altman. Learning the structure of biomedical relationships from unstructured text. *PLoS Computational Biology*, 11(7):e1004216, 2015. [5, 5, 8, 6]

107. Bethany Percha, Yael Garten, and Russ B Altman. Discovery and explanation of drug-drug interactions via text mining. In *Pacific Symposium on Biocomputing*, pages 410–421, 2012. [2, 4, 4.2, 9.1.1]

108. Conrad Plake, Torsten Schiemann, Marcus Pankalla, Jörg Hakenberg, and Ulf Leser. AliBaba: PubMed as a graph. *Bioinformatics*, 22(19):2444–2445, 2006. [2.3]

109. Conrad Plake and Michael Schroeder. Computational polypharmacology with text mining and ontologies. *Current Pharmaceutical Biotechnology*, 12(3):449–457, 2011. [2.3]

110. Sue Povey, Ruth Lovering, Elspeth Bruford, Mathew Wright, Michael Lush, and Hester Wain. The HUGO gene nomenclature committee (HGNC). *Human Genetics*, 109(6):678–680, 2001. [4.3]

111. James Pustejovsky, José M Castaño, Jason Zhang, Maciej Kotecki, and Brent Cochran. Robust relational parsing over biomedical literature: extracting inhibit relations. In *Pacific Symposium on Biocomputing*, volume 7, pages 362–373, 2002. [6]

112. Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. *NAACL HLT 2013*, pages 74–84, 2013. [3.2.2]

113. Thomas C Rindflesch and Marcelo Fiszman. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting

hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477, 2003. [6]

114. Thomas C Rindflesch, Lorraine Tanabe, John N Weinstein, Lawrence Hunter, et al. EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing*, volume 5, pages 514–25, 2000. [6.8]

115. Tim Rocktäschel, Michael Weidlich, and Ulf Leser. ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 2012. [4.1]

116. R Rogers and Ross Prpic. Profound symptomatic bradycardia associated with combined mibefradil and beta-blocker therapy. *The Medical Journal of Australia*, 169(8):425–427, 1998. [2.2, 2.1]

117. Benjamin Rosenfeld and Ronen Feldman. Clustering for unsupervised relation identification. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 411–418. ACM, 2007. [3.2.2]

118. Juana María Ruiz-Martínez, Rafael Valencia-García, Jesualdo Tomás Fernández-Breis, Francisco García-Sánchez, and Rodrigo Martínez-Béjar. Ontology learning from biomedical natural language documents using UMLS. *Expert Systems with Applications*, 38(10):12365–12378, 2011. [4.1]

119. Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition, 2003. [2.3.3]

120. Magnus Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE)*, volume 5, 2005. [3.2.1, 3.3, 3.4, 6.10]

121. Magnus Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Institutionen för Lingvistik, 2006. [1, 3.2.1]

122. Magnus Sahlgren, A Holst, and Pentti Kanerva. Permutations as a means to encode order in word space. In *30th Annual Meeting of the Cognitive Science Society*, July 2008. [3.4.1]

123. Gerard Salton. *The SMART retrieval system: experiments in automatic document processing*. Prentice-Hall, Inc., 1971. [3.2]

124. Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. [2]

125. Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. [3.2, 3.2.1]

126. Katrin Sangkuhl, Teri Klein, and Russ Altman. Selective serotonin reuptake inhibitors (SSRI) pathway. *Pharmacogenetics and Genomics*, 19(11):907, 2009. [8.1.1]

127. Hedi Schelleman, Warren B Bilker, Colleen M Brensinger, Xiaoyan Han, Stephen E Kimmel, and Sean Hennessy. Warfarin with fluoroquinolones, sulfonamides, or azole antifungals: interactions and the risk of hospitalization for gastrointestinal bleeding. *Clinical Pharmacology & Therapeutics*, 84(5):581–588, 2008. [2.1]

128. Diana Schmassmann-Suhijar, Roy Bullingham, Rodolfo Gasser, Jörg Schmutz, and Walter E Haefeli. Rhabdomyolysis due to interaction of simvastatin with mibefradil. *The Lancet*, 351(9120):1929–1930, 1998. [2.1]

129. H. Schütze. Dimensions of meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, Supercomputing '92, pages 787–796, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press. [3.2.1]

130. Hinrich Schütze and Jan Pedersen. A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113, 1993. [3.2.1]

131. Isabel Segura-Bedmar, Paloma Martinez, and Cesar de Pablo-Sánchez. Using a shallow linguistic kernel for drug–drug interaction extraction. *Journal of Biomedical Informatics*, 44(5):789–804, 2011. [6]

132. Burr Settles. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005. [4.1]

133. Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003. [4.4.2]

134. Hagit Shatkay and Ronen Feldman. Mining the biomedical literature in the genomic era: an overview. *Journal of Computational Biology*, 10(6):821–855, 2003. [1]

135. Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics, 2006. [3.2.2]

136. Alan R Shuldiner, Jeffrey R O'Connell, Kevin P Bliden, Amish Gandhi, Kathleen Ryan, Richard B Horenstein, Coleen M Damcott, Ruth Pakyz, Udaya S Tantry, Quince Gibson, et al. Association of cytochrome P450 2C19 genotype with the antiplatelet effect and clinical efficacy of clopidogrel therapy. *Journal of the American Medical Association*, 302(8):849–857, 2009. [2.2]

137. David Skillicorn. *Understanding complex datasets: data mining with matrix decompositions*. CRC press, 2007. [3.2.1, 5.1, 6.7]

138. Maria Stamelou, Niccolo E Mencacci, Carla Cordivari, Amit Batla, Nick W Wood, Henry Houlden, John Hardy, and Kailash P Bhatia. Myoclonus-dystonia

syndrome due to tyrosine hydroxylase deficiency. *Neurology*, 79(5):435–441, 2012. [9.2.1]

139. Don R Swanson. Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association*, 78(1):29, 1990. [5]

140. Akio Tanaka, Tadashi Nagamatsu, Makoto Yamaguchi, Atsushi Nomura, Fumiko Nagura, Kayaho Maeda, Tatsuhito Tomino, Tatsuhito Watanabe, Hideaki Shimizu, Yoshiro Fujita, et al. Myoclonus after dextromethorphan administration in peritoneal dialysis. *Annals of Pharmacotherapy*, 45(1):e1, 2011. [9.2.1, 9.2.1]

141. Buzhou Tang, Hongxin Cao, Xiaolong Wang, Qingcai Chen, and Hua Xu. Evaluating word representation features in biomedical named entity recognition tasks. *BioMed Research International*, 2014. [4.1]

142. Luis Tari, Saadat Anwar, Shanshan Liang, James Cai, and Chitta Baral. Discovering drug–drug interactions: a text-mining and reasoning approach based on properties of drug metabolism. *Bioinformatics*, 26(18):i547–i553, 2010. [2.3]

143. Nicholas P Tatonetti, JC Denny, SN Murphy, GH Fernald, G Krishnan, V Castro, P Yue, PS Tsau, I Kohane, DM Roden, et al. Detecting drug interactions from adverse-event reports: interaction between paroxetine and pravastatin increases blood glucose levels. *Clinical Pharmacology & Therapeutics*, 90(1):133–142, 2011. [2.1]

144. Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. [5.3.1]

145. Peter D Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th International Joint Conference on Artificial intelligence*, pages 1136–1141, 2005. [3.2.2, 5.1, 6.7]

146. Peter D Turney. The latent relation mapping engine: algorithm and experiments. *Journal of Artificial Intelligence Research*, pages 615–655, 2008. [3.2.2, 6.1, 6.7]

147. Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010. [3, 3.1, 3.2, 3.2.2]

148. Rafael Valencia-García, Jesualdo Tomás Fernández-Breis, Juana María Ruiz-Martínez, Francisco García-Sánchez, and Rodrigo Martínez-Béjar. A knowledge acquisition methodology to ontology construction for information retrieval from medical documents. *Expert Systems*, 25(3):314–334, 2008. [4.1]

149. Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41:W518–W522, 2013. [4.1, 4.3.2, 1]

150. M Whirl-Carrillo, EM McDonagh, JM Hebert, Li Gong, K Sangkuhl, CF Thorn, RB Altman, and Teri E Klein. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics*, 92(4):414–417, 2012. [1, 4.3, 6, 6.2.1, 1, 6.9]

151. Dominic Widdows and Kathleen Ferraro. Semantic vectors: a scalable open source package and online technology management application. In *LREC*, 2008. [4.2.3]

152. David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. DrugBank: a knowledge base for drugs, drug actions and drug targets. *Nucleic Acids Research*, 36(Suppl 1):D901–D906, 2008. [1, 2.3.3, 4.3, 6, 2]

153. Hong-Guang Xie, Richard B Kim, Alastair JJ Wood, and C Michael Stein. Molecular basis of ethnic differences in drug disposition and response. *Annual Review of Pharmacology and Toxicology*, 41(1):815–850, 2001. [2.2]

154. Rong Xu and QuanQiu Wang. A knowledge-driven conditional approach to extract pharmacogenomics specific drug–gene relationships from free text. *Journal of Biomedical Informatics*, 45(5):827–834, 2012. [6]

155. Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011. [3.2.2]

156. Lei Zhang, Yuanchao Derek Zhang, Ping Zhao, and Shiew-Mei Huang. Predicting drug–drug interactions: an FDA perspective. *The AAPS Journal*, 11(2):300–306, 2009. [2.2]

157. Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Natural Language Processing–IJCNLP 2005*, pages 378–389. Springer, 2005. [3.2.2]

158. Nava Zisapel. Melatonin–dopamine interactions: from basic neurochemistry to a clinical setting. *Cellular and Molecular Neurobiology*, 21(6):605–616, 2001. [9.2.2]

159. Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398, 2013. [3.2.2]