VizMaps: A Bayesian Topic Modeling Based PubMed Search Interface

by

Kirti Kamboj

Program in Statistical and Economic Modeling
Duke University

Date: ________________________
Approved:

____
David Banks, Supervisor

____
Charles Becker

____
Cliburn Chan

Dissertation submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Program in Statistical and Economic Modeling
in the Graduate School of Duke University
2015

Abstract

VizMaps: A Bayesian Topic Modeling Based PubMed Search Interface

by

Kirti Kamboj

Program In Statistical and Economic Modeling
Duke University

Date: ______________________

Approved:

____
David Banks, Supervisor

____
Charles Becker

____
Cliburn Chan

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Master of Science in the Department of Statistics
in the Graduate School of Duke University
2015

Abstract

A common challenge that users of academic databases face is making sense of their query outputs for knowledge discovery. This is exacerbated by the size and growth of modern databases. PubMed, a central index of biomedical literature, contains over 25 million citations, and can output search results containing hundreds of thousands of citations. Under these conditions, efficient knowledge discovery requires a different data structure than a chronological list of articles. It requires a method of conveying what the important ideas are, where they are located, and how they are connected; a method of allowing users to see the underlying topical structure of their search. This paper presents VizMaps, a PubMed search interface that addresses some of these problems. Given search terms, our main backend pipeline extracts relevant words from the title and abstract, and clusters them into discovered topics using Bayesian topic models, in particular the Latent Dirichlet Allocation (LDA). It then outputs a visual, navigable map of the query results.

Contents

List of Tables

List of Figures

Acknowledgements

I'm very grateful to my advisor, Charles Becker, for his advice and support, both in research and non-research matters. He's been a tremendously invaluable mentor for me.

Many thanks to Campbell R. Harvey, for giving me such interesting projects to work on (including one that led me to exploring academic databases and search engines and prompted this project), and for his mentorship. Thanks to Cliburn Chan, whose class taught me the skills I needed to finish this, and for providing timely advice. Thanks also to David Banks, for his prompt feedback and support.

Thanks to Duke's Innovations CoLab, for the grant and technical support that made this project possible, and Michael Faber and Jiehan Zhang in particular, for their invaluable assistance.

Eric Monson, at Duke's Data and Visualization Services, provided feedback that led to a much needed overhaul of both the website and the first draft of this thesis. I'm thankful for his expert guidance into the field of data visualizations, and his helpful comments.

Thanks to my MSEM cohort, especially Michael and Amaze. You're the reason I think back fondly even on late-night study sessions.

I'd like to thank David Banks, Charles Becker, Cliburn Chan again, for serving as my committee members, for being so generous with their expertise and time, and for making my defense such an enjoyable and interesting experience.

And finally, I would like to thank my parents, for their unfailing encouragement and support.

1

Introduction

Academic databases are online collections of citations, abstracts, and sometimes full-text articles from journals, magazines, and newspapers. Some have Application Programming Interfaces (APIs) that allow automated queries, while others demand manual searches. The last two decades have seen an explosive growth in the generation and collection of data, including those in academic databases. One of the most widely used, PubMed, had 20 million citations in 2010. Today, it has grown to over 25 million.

Academic researchers face the challenge of knowledge discovery – identifying, extracting, and understanding useful information to gain new insights. Due to the size and growth of modern databases, efficient knowledge discovery now requires a different data structure than the list of articles that databases output. It requires a list of ideas contained in each article, as well as other articles that have the same types of ideas (Blei and Lafferty, 2009). This mismatch between what researchers need and what most databases currently provide creates a particularly acute challenge for novice researchers in a field, who don't yet have the domain expertise to make focused queries or understand implicit connections. For example, naive queries on

PubMed, such as *obesity* or *autoimmune disorder*, yield reverse-chronological lists of hundreds of thousands of articles with related bibliographic information (titles, authors, abstracts, etc).[1]

This paper presents VizMaps, a PubMed search interface that addresses some of these problems. Given a search query, it grabs the relevant citations from PubMed's API, automatically extracts the main words from each title and abstract, and clusters them into topics. Among the clustering methods tested were variations of Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA), and Hierarchical Dirichlet Processes (HDP). The LDA was chosen for reasons of speed and accuracy.

Once these clusters have been discovered, it outputs a navigable map. This map is a node-and-edges network graph showing the underlying topics associated with the query, as well as their relationships. Each of the topics has an associated list of citations, ranked by order of importance. In this way, users gain an overview of the different branches of research associated with their search query, and can discover the relevant papers associated with each branch.

In designing VizMaps, our main purpose was to improve the quality and efficiency of the knowledge extraction process associated with PubMed informational queries. To this end, each decision made was an attempt to optimize three metrics – statistical and semantic accuracy, computational efficiency, and usability, with an emphasis on the latter.

The goal of this paper isn't to give a comprehensive overview of VizMaps' features. There are many that are straightforward and replicate previous work, such as readability rankings, summary statistics, and citations networks. Rather, in this paper we focus on the main pipeline, which abstracts the metadata from collected citations into topics, and delivers a navigable map.

[1] PubMed has recently added a feature that also sorts articles by relevance. It is unknown what goes into the relevance rankings, though it seems to be articles that are a best match based on keywords.

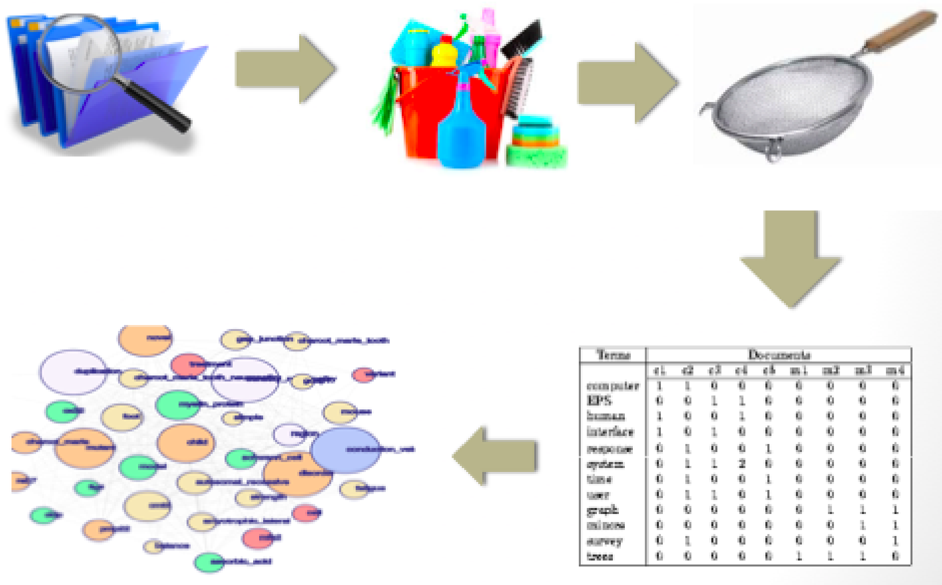

Figure 1.1: This is an overview of the pipeline. The relevant citations are first grabbed from PubMed. The data is cleaned, and then the important words are extracted. A term-document matrix is created, and from that, both the words and the documents are grouped into topics. The final step is the visualizations.

The rest of this paper is divided into four main parts. The first discusses the PubMed database, and provides an overview of the field of information visualization for search interfaces. PubMed's open licensing policies and backend API have in effect created a laboratory in which a wide range of alternative interfaces and third-party add-ons can be tested. Yet, almost all the current interfaces still output PubMed-like lists of results, due to challenges associated with using visualizatons in search interfaces.

The second details the process of feature extraction, in which we identify potential topic words from each title and abstract. There are two main issues that must be resolved in this step. The first is dimension reduction, which involves discarding words that contribute little information; the second is identifying important n-grams, or word phrases, such that word dependencies that significantly contribute to topic discovery are maintained. This feature extraction process highlights one of the limitations of our model. Since we don't utilize the National Library of Medicine's (NLM) more accurate and comprehensive lexical and semantic libraries, but rather Python's Natural Language Toolkit (NLTK),[2] our results are not as accurate as those of some other PubMed interfaces.

The third section covers our clustering algorithm. We use Bayesian topic models – "probabilistic models for uncovering the underlying semantic structure of a document collection" (Blei and Lafferty, 2009) – for extracting the main topics. Although Bayesian nonparametric topic models, such as Hierarchical Dirichlet Processes and Pitman-Yor Processes, would be ideal, the amount of time required for convergence, even for the more straightforward variational methods for inference, prohibits us from using them in our pipeline. Instead, we utilize the more computationally efficient Latent Dirichlet Allocation (LDA) model, using MeSH terms to calculate the number of topics. This is different from existing PubMed interfaces that do network analysis, which mostly use techniques such as citation networks, word-collocations, and Latent Semantic Indexing (LSI).

Finally, we outline our visualizations, and then conclude with a discussion.

[2] http://www.nltk.org

2

PubMed Database and Search Visualizations

2.1 PubMed Overview

PubMed[1] is the largest and most widely used biomedical search tool on the web. It contains over 25 million citations from approximately 5,600 journals in the life sciences, behavioral sciences, chemical sciences, and bioengineering. In addition to serving as a central index of biomedical literature, it also provides quality control in scientific publishing, using guidelines found on the National Library of Medicine (NLM) Fact Sheet.[2] Unlike most academic databases, it is freely accessible to the general public, and is developed and maintained by the National Center for Biotechnology Information (NCBI), a US government organization. Millions of queries are issued each day by users around the globe. While two-thirds of its users are domain experts – scientists, physicians, and other health care professionals – one-third are laypeople, such as those researching their diagnosed medical conditions (Mosa and Yoo, 2013; Lacroix and Mehnert, 2002).

About 3.6 of the over 25 million PubMed citations include links to free full-

[1] http://www.ncbi.nlm.nih.gov/pubmed

[2] https://www.nlm.nih.gov/pubs/factsheets/jsel.html

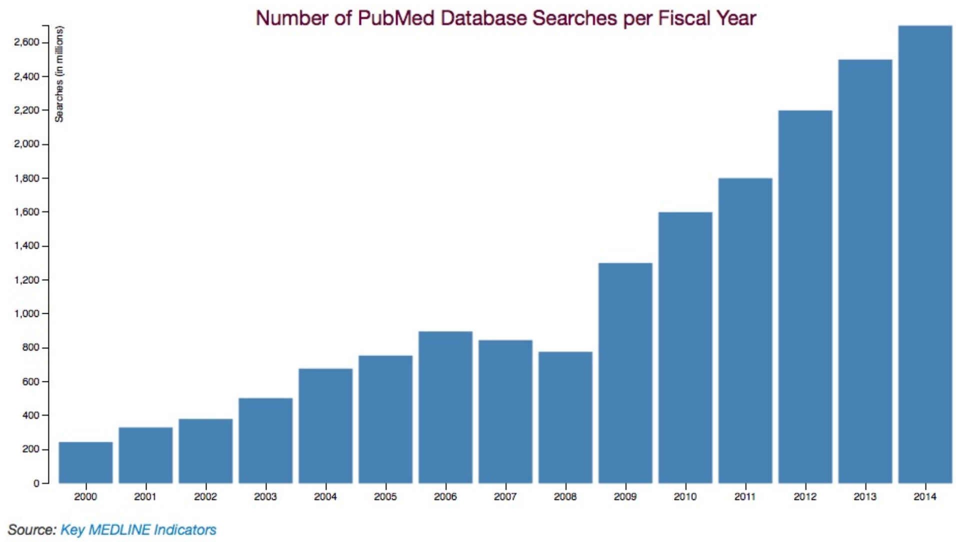

FIGURE 2.1: Growth of the PubMed database, as measured by the number of searches per year

text content from PubMed Central,[3] NCBI's full-text digital archive of biomedical and life sciences journal literature. Others are from MEDLINE, which contains bibliographic information stored in a structured database with 65 fields, such as titles and abstracts, and may include links to the full-text articles on publisher web sites (usually paid or firewalled access) (Mosa and Yoo, 2013; NCBI, 2013).

The MEDLINE database was launched in the 1960s, and for the following three decades, institutional facilities such as university libraries were necessary to access it. This changed in 1997, when the PubMed system made MEDLINE searches available to the public via the web. In the ensuing years, the number of added citations per year almost doubled, from about 400,000 in 1996 to 750,000 in 2015.[4]

This growth in the collection of citations has been a boon for researchers, but has also highlighted some shortcomings in PubMed's current interface. Over one-third of PubMed queries result in over 100 citations (Dogan et al., 2009), and querying topics

[3] http://www.ncbi.nlm.nih.gov/pmc/

[4] https://www.nlm.nih.gov/bsd/stats/cit$_{a}$*dded.html*

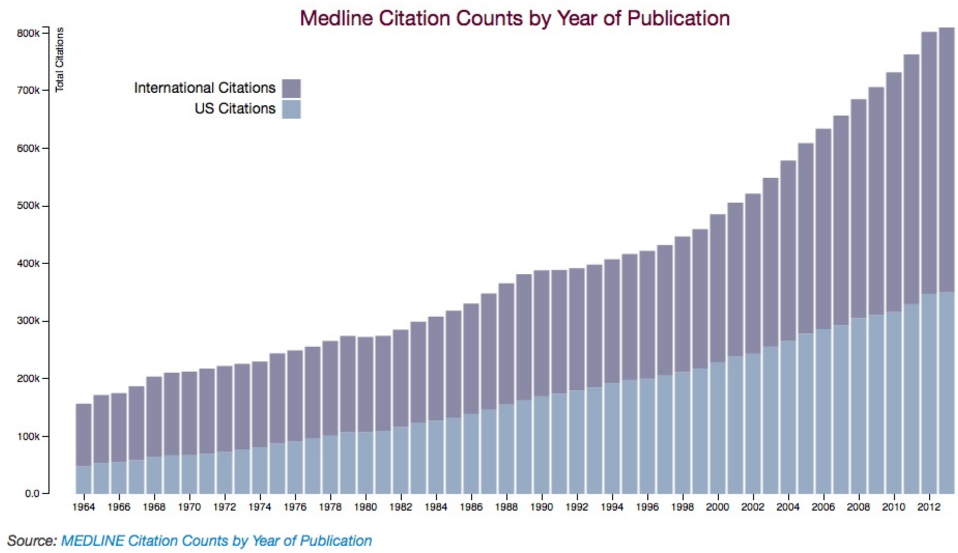

FIGURE 2.2: Growth of the PubMed database, as measured by the number of citations added per year

such as *obesity*, or *autoimmune disease*, whose rates have been rising, yields a reverse-chronological list of over 200,000 and 400,000 articles, respectively. This makes it difficult for researchers to determine which articles contain important concepts, or how they are linked. Even with rarer diseases, stories such as Kim Goodsells', who spent months combing the PubMed database for information about Charcot-Marie-Tooth, her diagnosed genetic disorder,[5] are prevalent.

The NCBI has made efforts to address this problem by suggesting more specific queries (Lu et al., 2009), and there are numerous articles and books focused on guiding PubMed users. A guideline suggested by all resources is utilizing PubMed's advanced search features, such as Medical Subject Heading (MeSH) terms and search field tags.

However, even experienced users have considerable difficulty extracting relevant information from the PubMed database. Over 94% of PubMed queries are performed

[5] http://mosaicscience.com/story/diy-diagnosis-how-extreme-athlete-uncovered-her-genetic-flaw?src=longreads

by users that don't use its advanced features. Google Scholar, with its relevance ranking algorithms, is twice as likely to present relevant search results, and is the most frequently used search engine for patient care (Thiele et al., 2010; Shariff et al., 2013). 65% of physicians utilize Google Scholar and UpToDate, which not just answer more clinical questions correctly, but also more efficiently, than PubMed. Only 13% utilize PubMed (Thiele et al., 2010; Nourbakhsh et al., 2012).

There is, however, a feature of the PubMed database that distinguishes it from competitors such as Google Scholar and UpToDate. Its Entrez Programming Utilities[6] and the free licenses the NLM provides to use the MEDLINE data,[7] in effect act to create a laboratory in which a wide range of alternative interfaces and third party addons, designed to help users more quickly and efficiently search and retrieve relevent publications, can be tested (Lu, 2011).

In the next section, we discuss some of the most utilized PubMed search interfaces. We then defocus, to discuss the broader topic of domain visualization and effective search interfaces. We conclude by discussing some current bibliometrics research tools for analyzing large citations networks, and the similiarities and differences between our work.

2.2 Existing PubMed Search Interfaces

Numerous PubMed derivative systems have been developed in the past fifteen years. The majority are by academic researchers, have an upper limit of citations, and feature a PubMed-like interface that outputs list-based search results. They can be placed in four groups – those that focus on ranking search results; categorizing results into topics; extracting and displaying semantics and relations; and improving search interface and retrieval experience (Lu, 2011). We only provide a glancing overview

[6] http://www.ncbi.nlm.nih.gov/books/NBK25501/

[7] https://www.nlm.nih.gov/databases/journal.html

of these interfaces; for more details, please refer to Zhiyong Lu's 2011 survey.

2.2.1 Ranking search results

For each PubMed search query, the resulting citations are given in reverse chronological order by default. This presents obvious difficulties for the majority of PubMed users, who are engaged in knowledge discovery rather than tracking the latest remotely relevant paper. Therefore, a major subgroup of PubMed Search interfaces presents alternative orderings, sorting results by calculated relevance, from algorithms that utilize direct user feedback (RefMed, MiSearch), to those that use classifiers to rank the relevancy of articles to the user's query (MedlineRanker, MScanner, eTBLAST).

PubFocus, which sorts articles based on various factors, including reference dynamics and authors' contribution level, shares high-level similarities with the ranking algorithm we use.

A major challenge in ranking PubMed articles is that, unlike academic databases such as the Web of Science, the majority of papers in PubMed provide no "cited by" or impact factor information.[8] This is why the majority of existing interfaces focus almost exclusively on relevancy calculations. However, by reversing some of the reference calls in e-utilities, we were able to capture an article's PubMed Central citations, and use this as an input for approximating its general level of influence.

2.2.2 Categorizing search results into topics

With potentially hundreds of thousands of search results, PubMed users are often faced with information overload. Dividing the results into topics makes the knowledge discovery process more manageable and efficient.

Most of these algorithms utilize PubMed's MeSH topics. MeSH is NLM's con-

[8] https://www.nlm.nih.gov/services/cite.html

trolled vocabulary or subject heading. Subject analysts go over papers submitted to the PubMed database and select ten to twelve MeSH terms to describe each paper.[9] One drawback of relying on this feature is that, since these MeSH terms are hand curated, newer publications – usually the most relevant to PubMed users – will not have them.

In these interfaces, papers are grouped into separate topics using either clustering algorithms (McSyBi, ClusterMed), or categorization methods via information contained in the Unified Medical Language System (Dynacat). The results are usually outputted in a PubMed-like list format.

The closest to our clustering algorithm is McSyBi, though it uses LSI rather than LDA for clustering. The differences between LSI and LDA clustering algorithms is discussed in Chapter 3.

2.2.3 Extracting and displaying semantics and relations

These PubMed interfaces analyze search results based on biomedical concepts and their relationships. The search interfaces are varied, from Europe PubMed Central, which integrates PubMed with other biomedical literature resources and highlights keywords such as gene names, organisms, and diseases in the outputted list of citations; to PubNet, which output network graphs based on shared authors, MeSH terms, or location.

2.2.4 Improving search interface and retrieval experience

This group provides alternative interfaces to PubMed. Innovations include interfaces for languages other than English (Babelmesh), PDFs displayed in search results (PubGet), and interactive search interfaces that allow users to search as they type (iPubMed).

[9] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2651214/

2.2.5 *Vizmaps*

Our PubMed search interface differs from those discussed above primarily in its scope, which covers all four of the above groups. We use LDA to group the search results into topics, and Google's pagerank algorithm to determine the importance of said topics. We use ranking algorithms to determine the importance of citations within each topic, and extract and display the relations between the topics in the form of a force-directed network graph. We then use these graphs as a means of allowing researchers to visually map their searches, with the goal of improving the interface and retrieval experience.

This visualization is another major difference between our interface and others – only PubNet outputs a similar visualization, without integrating other search-and-discovery functionalities like we do. There is controversy whether such visualizations add value. In the following sections, we will discuss this in greater detail.

2.3 Search Visualizations Overview

Search visualizations are a limited extension of domain visualizations, which use spatial representations to portray the interrelationships between research fronts, allowing researchers to navigate scientific literature based on the depicted spatial patterns. The relevant literature on domain visualizations is spread across disciplines that traditionally have few connections (Börner et al., 2003). Approaching this, as we do, from the discipline of statistics, in order to develop a search interface for the biomedical discipline, we must admit to having a limited understanding of the whole.

In this section, we will attempt to create a bridge to the fields of bibliometrics, bioinformatics, visualization, and user interface technologies, with the understanding that a thorough analysis is beyond the scope of this thesis.

Since it is important to study information-seeking behavior in order to design search interfaces that allow for efficient knowledge discovery, we begin by introducing some concepts from user interfaces (UI) literature. We then delve into the muti-faceted field of domain visualization, and explore existing bibliometrics software.

2.3.1 Search User Interfaces

An important quality of a user interface, such as our PubMed search interface, is its usability. Usability is determined by five properties – its learnability, efficiency, memorability, error rate and severity, and overall user satisfaction (Herskovic et al., 2007). A user-centered design technique has been developed in order to maximize an interface's usability. It begins with a needs assessment, "in which designers investigate who the users are, what their goals are, and what tasks they have to complete in order to achieve their goals" (Hearst, 2009).

When applied to the PubMed search engine, we find that user behavior is in some respects similar to that of user behavior on web search engines. Like web users, PubMed users favor short queries and issue few queries per session. (Herskovic et al., 2007) We also find that there are two types of queries that users make. Informational queries are intended to satisfy information needs on a topic (for example, "charcot marie tooth"), while navigational queries are intended to retreive a specific document or set of documents (for example, a specific article or articles related to charcot marie tooth). (Herskovic et al., 2007)

PubMed, like most digital libraries, is optimized for the latter task. This reflects its origins as an online counterpart to MEDLARS (Medical Literature Analysis and Retreival System). MEDLARS originated in 1964, when computerized information retrieval was carried out by members of a narrow demograhic, such as university librarians, primarily for navigational queries – to find the names of specific documents containing the requested information, then locate and retrieve their physical paper copies (Hearst, 2009; Rogers, 1964). However, the landscape has since changed.

Today, three quarters of all queries to the PubMed database are informational (Herskovic et al., 2007). As we have discussed in previous sections, PubMed does not provide the additional analysis tools necessary to efficiently carry out these informational queries for the majority of its users.

We believe that the pipeline we've created, and the visual maps we've developed, will improve the information extraction process associated with PubMed's informational queries. However, as Marti Hearst (2009) points out, "the typical search interface today is of the form: type-keywords-in-entry-form, view-results-in-a-vertical list. A comparison of a search results page from Google in 2007 to that of Infoseek in 1997 shows that they are nearly identical."

Designers have learned that they must be careful when introducing novelty and complexity; no matter how intuitive and obvious a new feature is, it will be mystifying, at least initially, to a significant portion of users. This is especially true when introducing visualizations to search interfaces, and part of the reason why the most prominent existing PubMed search interfaces output PubMed-like list-based citations. The other part of the reason, it is theorized, is because search is a means towards an end. "When reading text, one is focused on that task; it is not possible to read and visually perceive something else at the same time" (Hearst, 2009).

To date, there are no widely accepted applications of information visualization for search interfaces, and few positive usability results (Hearst, 2009; Chen, 2010). Our interface has not yet been introduced to the general public, so it is not known whether it will prove an exception.

2.3.2 Domain Visualization

Domain visualization aims to study scientific communication, provide deeper understanding of multidisciplinary and fast-moving knowledge domains, and forecast emerging trends, either by exploring citation paths in scientific literature, or by bibliometric mapping. Its history is rooted in fields such as scientometrics, bibliometrics, citation analysis, and information visualization (Börner et al., 2003).

Scientometrics is the quantitative study of scientific communications, which applies bibliometric methods, such as citations and content analysis, to scientific literature. Though a direct semantic relationship between the citing and cited works is commonly assumed, there tends to be little subject similarity among pairs of cited and citing documents (Börner et al., 2003; Harter et al., 1993). Bibliometric mapping, resting on the assumption that each research field and article can be characterized by lists and sublists of important keywords, addresses this shortcoming. The more important keywords two articles share, the greater their similarity. This work falls in the subfield of bibliometric mapping and content analysis.

In 1987, a NSF panel report (McCormick, Defanti, and Brown) recommended funding research on scientific visualization – mapping phenomenon onto static two or three-dimensional representations. Information visualization techniques differ from this on a technical level by providing more user interactivity. On a thematic level, they reveal interesting phenomena rather than providing clarification of well-known phenomena. Furthermore, while the former is typically used on physically based scientific data, the latter is for abstract, non-physically based data – in our case, bibliographic data sets.

A key driver of advances in information visualization is a quest to improve the effeciency and effectiveness of current information retrieval systems. Notable subfields are hypertext research, geographic information systems, and visualization of educational knowledge domains (Börner et al., 2003; Hook and Börner, 2005). In his seminal 1996 paper, Schneiderman presented the visual information-seeking mantra – "overview first, zoom and filter, then details on demand" – considered the optimal path to knowledge extraction.

An overview is important, since it allows the detection of overall patterns. It is also where the PubMed interface, as well as other widely used search interfaces, with their page upon page of outputted results, are most lacking. Current search engines don't reveal the inherent structure of the information being searched (Hook and Börner, 2005). This is why, despite the poor usability results of past visualization attempts, we still believe this research area is key to making the search interfaces of the future.

The visual maps created by our pipeline fall into an area between bibliometric knowledge discovery visualizations (KDVs) – graphic renderings of bibliometric data designed to provide a global view of a particular domain, as well as its structural details and salient characteristics – and concept maps. They resemble the former in being generated automatically, without explicitly labeled connections or notions of causality. However, KDVs are wider in scope, seeking to represent an entire domain of knowledge rather than that associated with individual search queries (Hook and Börner, 2005).

Our proposed PubMed Search Interface also fulfills the requirements of the zoom and filter and details on demand steps of the visual information-seeking mantra, allowing users to isolate topics and topic networks; retrieve a ranked list of relevant articles associated with each topic; and view abstracts and, when available, full-text links for further exploration.

2.3.3 Bibliometrics and Visualization Software for Citations Analysis

Major differences between these bibliometric tools, and the PubMed search interfaces discussed earlier, is that they are downloadable apps written in Java, requiring users to manually load in the data.

One of the newest, and most comprehensive, is the Action Science Explorer (ASE), designed to rapidly provide a summary of the overall citations, while also identifying key papers, topics, and research groups. In addition to the differences noted above, it has a much broader scale than this project, covering as it does any academic discipline; it more thoroughly incorporates citations information; utilizes different text mining and clustering algorithms; and has a different user interface. However, it presents the same type of network visualization, and has some of the same citations manager features. It's also undergone an evaluation phase in which participant usage was monitored, and feedback collected, though the sample size (six participants) makes us hesitant to draw sweeping conclusions (Gove et al., 2011).

Nevertheless, some relevant findings follow. Its overall node-link diagram view was frequently used by participants to orient themelves, illustrating that such visualizations add value. Its most used features were ranking and filtering operations designed to find important papers. Participants had trouble managing the interaction between citation text and other views. They wanted additional clustering techniques such as topic modeling, and they didn't want to be limited to sets found by the clustering algorithms. A key recommendation made for future researchers was to allow easy import from one or more general data sources (Gove et al., 2011). Our import process is completely automatic, and some other points raised are also fulfilled by our search interface.

Other research tools applying bibliometrics and visualization techniques for knowledge discovery in citations analysis are as follows. Network Workbench is a large-scale network analysis, modeling and visualization toolkit that allows users to construct co-author networks from bibliographic data (NWB Team, 2006). CiteSpace's specialty is identifying trends and intellectual turning points (Chen, 2014). Cytoscape is a popular software for network visualization and analysis, with most of its emphasis on the former, that's able to extend its functionality via apps. Since these research tools are weakly integrated into the document exploration process, they are not widely used in evaluating the output of academic search queries.

2.4 Summary

PubMed provides a useful laboratory for exploring and developing search engine interfaces, as evidenced by the numerous addons and search interfaces that exist. Almost all the most-widely used PubMed interfaces output PubMed-like search results, which reflects a larger trend. Applications of visualization to general search have not yet been widely accepted.

However, as the information-seeking mantra states, the most efficient path to knowledge discovery is overview first; zoom and filter; then details on demand (Shneiderman, 1996), and the first part of this, the general overview, is what all current widely-used search engines lack. Therefore, in building our interface, we begin with the overview step, presenting users with a nodes-and-edges network graph that displays the underlying topics in the outputted results, their components and relationships. If users are interested in a particular topic, they click on it, which brings up a side window view of a list of topics, starting with the clicked topic and going down to its least-related topic, along with additional details. Clicking on one of these topics makes the visual map fade into the background, and brings up a familiar PubMed-like interface, with a ranked list of articles relevant to that topic.

In this way, the overview step is separated from the one that requires reading and parsing text, and users are able to both grasp the underlying structure of their search results, isolate the most relevant topics, and then proceed to examine the most relevant articles related to that topic in a familiar list format.

3

Building The Feature Space

3.1 Introduction: Creating a term document matrix

Entrez Direct is a server-side program that provides an interface to the PubMed, allowing users to search and retreive requested data using a fixed URL syntax[1]. We use its esearch function to directly query the database using inputted terms (i.e., *autoimmune disorder*), and collect a list of the relevant articles, with their title and abstract information, among others.

Once we've collected the data – in particular, the titles and abstracts of relevent citations – we use the Vector Space Model (VSM) to transform the titles and abstracts of each citation into a numerical format that can be stored and used for analysis. Each citation is represented as a vector in a high-dimensional space, its dimensionality determined by the number of unique words in the entire corpus, subject to filtering. The vectors together form a term-document matrix, where each row represents a specific article citation and each column corresponds to a unique word, with the cells representing how many times that word appears in that article citation (subject to

[1] http://www.ncbi.nlm.nih.gov/books/NBK25501/

different weights for title and abstract appearances). As is standard in topic modeling literature, our model does not care about the position or relationship between words (Salton and Michael, 1983). There is an underlying assumption of word independence and exchangeability – a simplification that represents any text as a bag of words, without case or order or underlying grammar.

There are two main problems with this model. The first is that the bag of words assumption reduces computational time and complexity at the expense of potentially significant information loss. The second is that large document collections have correspondingly large row and column dimensions in their term-document matrices. This is a problem because, as discussed earlier, over a third of PubMed searches return over a hundred citations, and some naive searches return hundreds of thousands of citations.

To reduce this information loss, instead of considering the text as just a collection of unique words, we will consider the text as a collection of n-grams – one, two, or n words in a sequence – using algorithms discussed in the following section. However, incorporating n-grams into the term document matrix further increases its already high dimensionality.

To reduce the dimensionality, we notice that the majority of words in the titles and abstracts of citations are either irrelevant or redundant, delivering very little information. Selecting the words, or features, that deliver the most information, and discarding the rest, improves the effectiveness and efficiency of our next-stage clustering algorithms. In the third section, we discuss this dimensionality reduction in greater detail.

3.2 Identifying ngrams

Currently, there are two main methods researchers have of addressing the problem of information loss incurred by representing text as bags-of-words; of, in effect, recapturing word dependencies. The first is topical (semantic) dependency, also known as long-distance dependency, where two words are considered dependent if their meanings are related and they co-occur often, such as *fruit* and *apple*. The second is phrase dependency, also known as short-distance dependency, which requires a method of discovering word collocations (Wang et al., 2007). In this section, we are interested in the latter. We will depend on topic models, covered in the next chapter, to automatically reveal the necessary semantic dependencies.

Learning n-grams requires finding words that appear frequently together, and infrequently otherwise. For example, *Charcot Marie Tooth* should be flagged as a trigram, while *flagged as* shouldn't be flagged as a bigram. Many techniques have been developed to identify ngrams, ranging from identifying noun patterns to utilizing word frequencies and variations. But despite years of research, no single technique has emerged as the best (Pedersen et al., 2011). For this reason, we use one of the simplest and most computationally efficient techniques, where phrases are formed based on word counts. For identifying bigrams, the equation is:

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i)count(w_j)}$$

where δ is the discounting coefficient that prevents too many phrases consisting of infrequent words to be formed, and counts include both the prior and observed information. Bigrams with scores above the chosen threshold are then stored. The extension for identifying trigrams, etc., is straightforward.

3.3 Reducing dimensionality and post-processing

3.3.1 reducing dimensionality

The first step to reducing dimensionality is filtering out non-content-bearing function words, such as common english stop words (and, of, is, ...). Doing so incurs minimal information loss, while at the same time significantly speeding up computation time.

Another such step is to reduce redundancy by merging different forms of the same word (disorder, disorders). There are two main ways of doing so, lemmatization and stemming. Stemming chops off the ends of words; lemmatization returns its base form. We lemmatize the corpus using NLTK and WordNet.

Next, we filter out words that appear too often or too little, and that appear in too few or too many citations. If there are a hundred citations, and there is a word that appears in all of them, then it conveys no information that could be used to differentiate the citations from each other. Alternatively, if it only occurs a handful of times, it conveys very little information that could be used to group the citations together, to find commonalities and the underlying structure of the corpus.

The main building blocks of sentences are noun phrases and verb phrases. Noun phrases are usually objects of the sentence, while verb phrases describe some action between those objects. Since noun phrases, or noun n-grams (n words in a sequence) contain the most information about discovering the underlying themantic sense of a topic, we extract the noun phrases of each citation (or rather, each citation's title + abstract) in a corpus of citations (the full query output, containing a list of indexed titles and abstracts).

NLTK has a tagger (POS-tagger), that attaches a part of speech tag to each word, with an accuracy rate of over 90% (Manning, 2011). We use this to identify and extract the nouns and noun phrases from the corpus.

This results in a list of nouns and noun phrases – alternately called the set of features, or vocabulary – associated with each article, and the corpus – our list of citations – represented in the form of a sparse term-document matrix.

3.3.2 post-processing

As stated above, we weigh words differently depending of whether they appear in the title or the abstract. If the appear in the former, it is counted as two appearances.

The term frequency-inverse document frequency (tf-idf) is a statistic that reflects how important a word is to a document in a corpus. A simple way of determining this is counting the number of times each term occurs in each document and summing them all together – that is, looking at the term frequency. However, this leads us to give more weight to common terms that appear in each document (such as *the*), and less to more unique terms. To correct for this bias, we use inverse document frequency, where the specificity of a term is the inverse function of the number of documents in which it appears.

The tf-idf is calculated by multiplying the term frequency with the inverse document frequency, where the latter is defined as:

$$idf(t, D) = \log \frac{N}{|d \in D : t \in d|}$$

We also use a variant of the tf-idf where we take the log of the frequency counts, rather than the frequency counts themselves.

3.4 Discussion

Our PubMed search interface falls short of others in this section of the pipeline. We use the libraries associated with Python's Natural Language Toolkit (NLTK) to identify stop-words, lemmatizations, and parts of speech. However, a much more comprehensive set of tools and databases exist for the biomedical field in the Unified Medical Language System (UMLS).[2] These tools include a metathesaurus, a multilingual vocabulary database that contains information about biomedical and health-related concepts; a semantic network which categorizes the concepts from the metathesaurus and charts their relationships; and most especially, the SPECIALIST Lexicon, a general English lexicon that includes both common English words and biomedical vocabulary.

[2] http://www.ncbi.nlm.nih.gov/books/NBK9675/

Some other PubMed search interfaces discussed in the previous chapter, such as McSyBi, use the UMLS for both their feature extraction and categorization purposes. Integrating it with the NLTK libraries in our pipeline will yield more accurate results, though doing so is currently beyond the scope of this paper.

In the next chapter, we discuss the clustering algorithms used to extract the underlying topical structure of PubMed search results.

4

Clustering

4.1 Introduction

Now that we've completed feature extraction and post-processing steps and built a term-document matrix, we've reached the most crucial component of this process. As Tyron and Bailey wrote in 1970, "understanding our world requires conceptualizing the similarities and differences between the entities that compose it". In other words, it requires clustering, an unsupervised learning problem where finite unlabeled data sets, in our case the individual citations, are separated into a finite and discrete set of natural, hidden data structures.

Clustering is intuitively simple to understand as dividing data into groups, such that entities in the same group are similar to each other, while entities in different groups are not. However, this simplicity leads to a bewildering number of clustering methods, as the following list of some of the most widely used algorithms shows: hierarchical clustering, such as lineage metrics; partitioning relocation clustering, such as k-means methods; density-based partitioning; grid-based methods; high dimensional clustering, which includes spectral clustering techniques such as LSI; probabilistic clustering algorithms, such as LDA; deep learning techniques, such as artificial neural networks.

Also, while clustering is intuitively simple to understand, analyzing clustering methods isn't. There are many factors that come into play, such as effective similarity measures, criterion functions, and initial conditions (Kotsiantis and Pintelas, 2004). Moreover, no clustering method adequately handles all cluster structures, and relating clustering algorithms to specific problems and data types remains an open and fundamental problem (Kotsiantis and Pintelas, 2004; Jain et al., 1999). As Backer and Jain (1981) point out, "in cluster analysis a group of objects is split up into a number of more or less homogenous subgroups on the basis of an often subjectively chosen measure of similarity (i.e. chosen subjectively based on its ability to create 'interesting' clusters)". For this reason, there isn't a best, or set of best, clustering algorithms, except in well-prescribed sub-domains.

A thorough, or even glancing, overview of different clustering algorithms is beyond the scope of this paper. Instead, in the following sections, we concentrate our attention on the two most widely used clustering algorithms in topic modeling – latent semantic indexing (LSI), and latent dirichlet allocation (LDA), along with variations and innovations.

4.2 Topic Models

4.2.1 Latent Semantic Indexing (LSI)

Topic models are algorithms that uncover the abstract topics in document collections. Latent semantic indexing (LSI) is one of the most common topic models in use today. Developed in 1988, when Dumais and co-workers applied the SVD algorithm to the term-document matrix, it presented a significant improvement over simplistic term matching by accounting for term dependencies, and allowed users to retrieve information on the basis of the conceptual topics. It is also the algorithm used in some of the PubMed search interfaces and bibliometric software reviewed in the first chapter.

However, despite its popularity, the LSI algorithm has drawbacks. Its discovered topics and topical infrastructure (the words making up the topic, topic weightings) are often difficult to interpret. Also, since higher-order co-occurrence paths hidden in the term-term LSI matrix are responsible for the term weights in its term-document matrix, altering a single term can result in a redistribution of weights across the entire matrix. Furthermore, LSI doesn't perform well in the presence of ambiguity (when the same word, such as *apple*, could refer to the fruit or the multinational corporation). Its drawbacks are also evident when considering the results of an LSI transformation. Table 4.1 shows five topics, along with their topic loadings, for the search query *autoimmune disease*, taking the first 1000 citations.

It is difficult to tell what the loadings of the LSI transformation actually mean, to map them into topics. Most words are shared between the topics, such as *determinant*, with negative loadings in the first two topics and positive in the third. This makes the LSI transformation suboptimal for determining and extracting the unique underlying topics in the corpus.

One reason for the similarity in words between topics is that each document can only belong to one topic. For this reason, Hoffman (1999; 2001) introduced the probabilistic topic approach (pLSI), which models each word in a document as a sample from a mixture model. The mixture components are multinomial random variables viewed as representations of topics. However, as Blei (2003) pointed out, pLSI "is incomplete in that it provides no probabilistic topic model at the level of documents. In pLSI, each document is represented as a list of numbers (the mixing proportions for topics), and there is no generative probabilistic model for these numbers." Therefore, the number of parameters in the model tends to grow linearly with the size of the document collection, leading to overfitting.

Table 4.1: Five distinct topics (along with their associated word loadings) that result from the lsi transformation.

Words	T1 Load	T2 Load	T3 Load	T4 Load	T5 Load
immunoglobulin	-0.92	0.107	0.323	-0.054	-0.094
level	-0.231	–	-0.396	0.617	.551
negative	-0.117	-0.038	-0.477	–	-.551
promising	-0.114	-0.049	-0.381	-0.710	.442
virus	-0.104	–	–	-0.066	–
circulating	-0.103	-0.045	-0.455	0.086	.332
yet	-0.101	-0.059	–	–	-0.111
synoviocytes	-0.089	–	–	–	–
patient	-0.086	–	–	-0.184	0.115
determinant	-0.070	-0.988	.119	–	–
growth	–	-0.028	–	–	–
associated	–	-0.022	–	–	–
ivig	–	-0.022	-0.186	-0.121	–
arthiritis	–	-0.017	–	–	–
modulation	–	–	-0.160	–	–
autoimmune	–	–	-0.142	–	-0.139
hiv-infected	–	–	-0.098	0.142	0.117
rheumatoid	–	–	–	-0.076	–
disease	–	–	–	-.063	0.061

These problems are fixed by Blei's (2003) Latent Dirichlet Allocation (LDA) Model, the first fully probabilistic topic model.

4.2.2 Probabilistic Topic Models

In probabilistic topic models, topics are characterized by probability distributions over a dictionary of words. A document can be restricted to belonging to only one topic, or containing several different topics. In the latter case, the document is considered a probability distribution over the set of possible topics (Steyvers and Griffiths, 2007).

The most common topic model in use today is the Latent Dirichlet allocation (LDA), a probabilistic topic model in which documents D are mixtures over latent topics K, and topics are multinomial distributions over a vocabulary of W words. Both the topic distribution in each document, and the word distribution in each topic, are given Dirichlet priors.

The generative process is described as follows. For document j, we first draw a mixing proportion θ_j from a Dirichlet with parameter α, such that $\theta_j \sim D(\alpha)$. For the i^{th} word in the document, a topic $z_{ij} = k$ is drawn with probability $\theta_{k|j}$. Word x_{ij} is drawn from topic z_{ij}, with x_{ij} taking on value w with probability $\beta_{w|z_{ij}}$. A Dirichlet prior with parameter η is placed on the word-topic distributions β_k, such that $\beta_k \sim D(\eta)$.

The distribution is as follows:

$$p(\beta, \theta, z, w) = (\prod_{i=1}^{K} p(\beta_i|\eta))(\prod_{d=1}^{D}(p(\theta_d|\alpha) \prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K}, z_{d,n})) \tag{4.1}$$

Running the LDA algorithm requires defining the number of topics, K. In our case, we obtain an estimate of K by considering the number of distinct MeSH keywords associated with each query, with an upper bound of 100. While this upperbound will bias results for queries with large citation counts, given the limitations of our visualization, it is necessary to ensure usability for the overall interface (it is difficult to parse d3 force-directed graphs whose topics exceed this number).

There are other methods of determining the number of topics, such as stability analysis, that can be used for databases that do not have such indexing.

Once the number of topics has been determined, LDA assigns temporary topics to each word according to the Dirichlet distribution. Making inference involves computing the posterior distributions of the latent variables in a document, which has an intractible closed form solution.

However, posterior inference is straightforward using Collapsed Gibbs Sampling. We iteratively loop through each word in each document, updating its topic assign-

Table 4.2: Words associated with four topics that result from the LDA transformation.

T1 Words	T2 Words	T3 Words	T4 Words
hiv-infected	immunoglobulin	suppresses	synoviocytes
sepsis	synthase	allosteric	traumatic
level	fat	disturbance	functional
association	long	immunoglobulin	experimental
patient	genetic	response	management
systematic	controlled	profile	humoral
circulating	immunological	characteristic	remyelination
hypogammaglobulinemia	hepatitis	virus	pilot
immunoglobulin	modulators	resistance	multiple
variation	child	circulating	iviv

ment z_i based on the word's prevalence across topics, and the topic's prevalence in that document, following the distribution outlined below, until the chain converges. For z_i:

$$P(z_i = j | z_{-i}, w) \sim \frac{n_{-i,j}^{(w_i)} + \eta}{n_{-i,j}^{(\cdot)} + W\eta} \frac{n^{(d_i)} + \alpha}{n_{-i}^{(\cdot)} + K\alpha} \tag{4.2}$$

where $n_{-i,j}^{(w_i)}$ is the number of instances of word w in topic j excluding the current; $n_{-i,j}^{(\cdot)}$ is the total number of words in topic j exluding the current; $n^{(d_i)}$ is the number of times topic j is assigned to some word token in document d, excluding the current; and $n_{-i}^{(\cdot)}$ is the total number of words in document j excluding the current word.

Further details of both the generative process and the sampling algorithm are found in several papers (Blei and Lafferty, 2009; Blei et al., 2003; Steyvers and Griffiths, 2007). An example of the sample output is shown in Table 4.2.

A faster inference algorithm than Collapsed Gibbs Sampling is Variational Inference, also called Variational Bayes (VB) when used in a Bayesian hierarchical model. While MCMC methods seek to generate independent samples from the posterior, Variational Inference optimizes a simplified parametric distribution close in

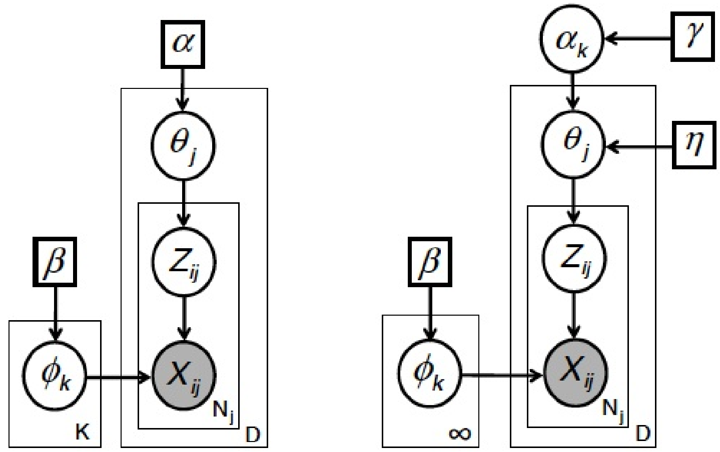

FIGURE 4.1: Graphical models for LDA (left) and HDP (right). Figure is taken from Newman et al (2009)

Kullback-Leibler (KL) divergence to the posterior. While MCMC methods can be made arbitrarily accurate by increasing the sampling sizes, in empirical tests, Variational Inference has been shown to be as accurate as MCMC (Hoffman, 2010).

Instead of Collapsed Gibbs Sampling, we use Variational Inference to increase efficiency.

The LDA output, when compared with the LSI output above, is easier to understand and separate into topics. Most words in each topic are distinct, though there is some overlap, as we would expect. Some documents that contain Topic 1 (HIV infected, sepsis, level...) are: *Mild Dementia Related to Unsafe Street-Crossing Decisions?*, *Optimal restricted estimation for more efficient longitudinal causal inference*, and *The JAK2 V617F mutational status and allele burden may be related with the risk of venous thromboembolic events in patients with Philadelophi-negative myeloproliferative neoplasms.*

Collected data from limited empirical tests (perplexity scores, based on a bag-of-words assumptions), also shows the LDA consistently out-performing pLSI (Blei et al., 2003).

The Hierarchical Dirichlet Process Mixture Model (HDPMM) is a nonparametric extension of LDA. As with LDA, each document is still viewed as a group of observed words, whose document-specific mixture components, or topics, are distributions over the words. But now, the number of possible mixture components is assumed to be unknown a priori, to be inferred from the data (Teh et al., 2006). There is an initial base measure H, from which a random measure G_0, which represents the set of all topics (mixture components) that can be used in a given corpus, is drawn. G_0 is a countably infinite collection of multinomial probability vectors. For each document j, G_j is sampled from the base measure G_0, representing the specific subset of topics used. The allows for information sharing between documents, while retaining the ability of each document to have its own topic mixing proportion. Each word x_{ji} is represented as a draw from a multinomial probability vector θ_{ji}, drawn from G_j. Mathematically,

$$G_0 \sim DP(\gamma H) \tag{4.3}$$

$$G_j \sim DP(\alpha_{j0} G_{j0}) \tag{4.4}$$

$$\theta_{jn} \sim G_j \tag{4.5}$$

$$w_{jn} | \theta_{jn} \sim F_{\theta_{jn}} \tag{4.6}$$

where the α and γ are concentration parameters.

Further information about the Dirichlet Process can be found in the Appendix. Details of the sampling algorithm for HDP can be found in (Teh et al., 2006) and (Newman et al., 2009).

The HDP topics contain mostly distinct words, just like the LDA topics. However, there does seem to be more specificity to the words, indicating that it is extracting more of the unique features of each topic. While running the LDA model, I arbitrarily set K to 20. In the HDP output, the discovered K was around 30.

Table 4.3: Four distinct topics (along with their associated word loadings) that result from the HDP transformation.

T1 Words	T2 Words	T3 Words	T4 Words
hemorrhagic	poor	producing	gastroenteritis
marker	pheotypically	hormone	resonance
advanced	dog	attenuates	preventing
required	platelet	ccr6	regulating
auto antigen	endothelial	morpheme	population
neuropathy	technique	aged	multicenter
a-induced	subset	controlled	endothelial
snake	igg	encephalopathy	case-control
relevance	node	adipocyte	ability
traditional	ketogenic	tlr2	drive
disseminated	pemphigiodes	alters	rheumatoid
middle	mitigates	extracellular	aging
memory	carrier	phosphorylation	glioma
non-segmental	communication	encephalitis	ana
quinolinic	long-term	hyaluronan	senthesis
tertiary	regulates	clinic	low-dose
phosphorylation	complication	lung	ingredient
failure	combination	survival	superoxide
ccr6	arteritis		effector
minimal			experience

There are two major drawbacks of using the HDP algorithm. The first is that its discovered number of topics do not follow a power law distribution. This distribution is ideal for cases where feedback from a set of events influences future events; that is, for investigating things such as network effects. This is the case with topic modeling, since the number and makeup of current topics influences that of future topics (i.e., 'hot' topics seem to get disproportionately more papers focused on them, etc). For this reason, a more statistically accurate topic modeling algorithm would be the Pitman-Yor Process (Teh et al., 2006; Pitman and Yor, 1997), a generalization of the HDP that ensures the model produces power-law distributions.

The second is that it is extremely computationally expensive, and the Pitman-

Yor Process even more so. Even when we use the more efficient variational inference algorithm for the HDP, shown to quite closely approximate the results of the more traditional Gibbs sampling method (Wang et al., 2011), it proves infeasible for a corpus of 10,000 citations and above.

4.3 Topic Labels

We use Chavalarias and Ioannidis's (2010) methodology to label topics. We create a specificity index I_s and a genericity index I_g. The specificity index is the extent to which each word or word group w is specific relative to others in the same topic. The genericity index indicates the extent to which it is generic:

$$I_s(w) = \frac{1}{card(C)} \sum_{w' \in C} P_{\max(\alpha, 1/\alpha)}(w, w') \tag{4.7}$$

$$I_g(w) = \frac{1}{card(C)} \sum_{w' \in C} P_{\min(\alpha, 1/\alpha)}(w, w') \tag{4.8}$$

$$P_\alpha(i,j) = \left(\left(\frac{n_{ij}}{n_i}\right)^\alpha \left(\frac{n_{ij}}{n_j}\right)^{\frac{1}{\alpha}}\right)^{\min(\alpha, 1/\alpha)} \tag{4.9}$$

We then choose the most specific word, subject to some constraints, to act as the topic label.

4.4 Discussion

While the Pitman-Yor Process would be ideal, out of the algorithms discussed, for this project, its computational complexity makes it intractable. A straightforward extension of the HDP, using the online VB algorithm developed by Wang et al. (2011) , would make it feasible if used in a cluster computing framework. Since we currently lack the pipelines and computational infrastructure necessary to support this, we use LDA as our clustering algorithm, with MeSH terms used to estimate the number of topics.

The LDA seems more theororetically sound and empiracally accurate than its predecessor pLSI. However, as discussed earlier in this section, there isn't a straightforward way to determine whether it and other probabilistic topic models give the best results, when compared to other clustering algorithms. However, we do hope that, by collecting enough user data on our site, we might be able to make headway on this question.

5

Visualization

Networks are useful for seeing relationships in large data sets such as citations information. The three major ways of representing networks are as lists of relationships, nodes and edges, and as adjacency tables or matrices.

The simplest and least revealing network representation is the list of relationships. Nodes and edges is the standard representation, and is also used in this paper. The last representation, using adjacency matrices, is often in the form of a heat map, with the intensity of the colors representing the strength of the relationships. This representation is ideal for highly connected networks, whose underlying structure is difficult to perceive via nodes-and-edges representations. Nodes-and-edges representations are ideal for networks with sparser connections.

In the nodes-and-edges representation we utilize, circles are nodes and lines are edges. Edges have weights, while nodes have properties such as degree (the number of connections) and clustering coefficient (amount of connections in the neighborhood of a node divided by the number of connections it could have), representing the density of the network around the node.

A requirement of this visualization is an automatic network layout. There are numerous algorithms for this, all of which attempt to reduce overlap between nodes and edges, since overlapping nodes and crossed edges create distracting visual complexity. This is usually portrayed as an optimization problem, with an objective function defined over the position of nodes based on aesthetic criteria such as length of edges, number of crossings, etc.

The most popular layout, and the one we use, is a force-directed network. It represents the network as a simulated system where nodes are repelling each other, and the edges are pulling the nodes together. Our specific implementation of this, using the open-source d3 Javascript library,[1], is the d3 force-directed network layout.[2] It uses a position Verlet integration to allow simple constraints, a quadtree utilizing the Barnes-Hut approximation to accelerate charge interaction, and a pseudo-gravity force to keep nodes centered in the visible area (Dwyer, 2009).[3]

Most networks in bibliometrics are built using citation paths, co-authorships, common meta keywords (such as MeSH terms), or word co-occurrances. Our network is built by exploiting a feature of the LDA algorithm, which portrays documents as mixtures of topics. Each node/circle represents an underlying topic associated with the query. Each edge represents a link between topics, determined by the weighted number of documents that share the two topics.

The node/circle size represents how important and influential that topic is in the list of outputted citations, and the procedure for determining this involves multiple steps. First, we find the weighted number of citations associated with each topic. The next step is the PageRank algorithm, which uncovers a topic's real influence by finding the stationary distribution of the probability of arriving at that topic after following links from other topics (Page et al., 1999). We then use an automatic scale

[1] http://d3js.org/

[2] http://bl.ocks.org/mbostock/4062045

[3] https://github.com/mbostock/d3/wiki/Force-Layout

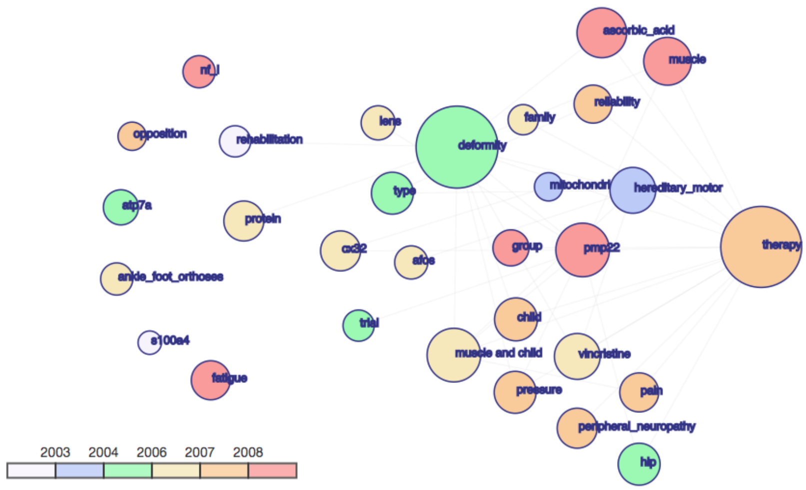

FIGURE 5.1: VizMaps' nodes and edges visualization

to normalize the sizes, ensuring that all node sizes lie between certain parameters.

Finally, the node colors represent just how topical a certain topic is. The higher the percentage of recent publications, the more intense the color, allowing researchers to identify current research and publication trends.

6

Conclusion

The metadata associated with scientific articles, such as keywords, title, and citation count, contains a significant amount of information. But when there are a high volume of articles, as discussed above, the amount of metadata is hard for researchers to assimilate, resulting in information overload. Traditional search engines, such as PubMed's, cannot effectively abstract the underlying relationships between the articles and cannot reveal the underlying structure of ideas contained in the citations.

For this reason, we've built a pipeline which automatically abstracts, organizes, and represents the information contained in citations in a way that allows for more efficient knowledge extraction.

Our main innovation over existing PubMed interfaces is twofold. The first is an initial visualization component, which allows for interactive knowledge discovery and is in line with the information-seeking mantra of overview first, then zoom and filter. The second is our use of LDA as a clustering algorithm, as well as the more straightforward MeSH terms and author citation networks. Using LDA allows us to cluster the newer PubMed articles, which don't yet contain MeSH information, and gives us a more thorough overview of the corpus, though it also adds the complexity of a high-dimensional feature space, requiring much more cleaning and preprocessing of the data.

Immediate next steps would be to use online distributed computing and variational inference techniques and apply HDP and Pitman-Yor Processes, rather than LDA, to extract clusters. There are other clustering methods that could also be applied; however, because of the nature of the problem, it is difficult to empirically determine which are the best. But once we begin collecting user data, by analyzing the number of corrections each user makes, we will be able to compare different clustering algorithms.

There are also other aspects of this problem that impact user experience, such as the use of caching to more quickly retreive common searches and developing an online distributed framework, that also lie beyond the scope of this paper.

Appendix A

Appendix

A.0.1 Dirichlet Process

A probability measure is a function from subsets of a space X to [0,1] that is always greater than or equal to zero and that sums to one. The Dirichlet Process (DP) is a distribution over probability measures. It has two parameters:

- base distribution H, which can be thought of as the mean of the DP
- strength parameter α, which can be thought of as the inverse-variance of the DP

Mathematically,

$$G \sim DP(\alpha, H) \tag{A.1}$$

$$\theta_i | G \sim G \tag{A.2}$$

Marginalizing out G, the conditional distributions are

$$\theta_n | \theta_{1:n-1} \sim \frac{\sum_{i=1}^{n-1} \delta_{\theta_i} + \alpha H}{n - 1 + \alpha} \tag{A.3}$$

This is otherwise known as Polya's urn scheme. Since the θs are i.i.d. given G, they are exchangeable.

Bibliography

Backer, E. and Jain, A. K. (1981), "A clustering performance measure based on fuzzy set decomposition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 66–75.

Blei, D. M. and Lafferty, J. D. (2009), "Topic models," *Text mining: classification, clustering, and applications*, 10, 34.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003), "Latent dirichlet allocation," *the Journal of machine Learning research*, 3, 993–1022.

Börner, K., Chen, C., and Boyack, K. W. (2003), "Visualizing knowledge domains," *Annual review of information science and technology*, 37, 179–255.

Chavalarias, D. and Ioannidis, J. P. (2010), "Science mapping analysis characterizes 235 biases in biomedical research," *Journal of clinical epidemiology*, 63, 1205–1215.

Chen, C. (2010), "Information visualization," *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 387–403.

Chen, C. (2014), "The CiteSpace manual," .

Dogan, R. I., Murray, G. C., Névéol, A., and Lu, Z. (2009), "Understanding PubMed® user search behavior through log analysis," *Database*, 2009, bap018.

Dwyer, T. (2009), "Scalable, versatile and simple constrained graph layout," in *Computer Graphics Forum*, vol. 28, pp. 991–998, Wiley Online Library.

Gove, R., Dunne, C., Shneiderman, B., Klavans, J., and Dorr, B. (2011), "Understanding scientific literature networks: an evaluation of action science explorer," *University of Maryland Technical Report (March 2011)*.

Harter, S. P., Nisonger, T. E., and Weng, A. (1993), "Semantic relationships between cited and citing articles in library and information science journals," *Journal of the American Society for Information Science*, 44, 543–552.

Hearst, M. (2009), *Search user interfaces*, Cambridge University Press.

Herskovic, J. R., Tanaka, L. Y., Hersh, W., and Bernstam, E. V. (2007), "A day in the life of PubMed: analysis of a typical day's query log," *Journal of the American Medical Informatics Association*, 14, 212–220.

Hoffman, Matthew, F. R. B. D. M. B. (2010), "Online learning for latent dirichlet allocation," *Advances in neural information processing systems*, pp. 856–864.

Hook, P. A. and Börner, K. (2005), "Educational knowledge domain visualizations: tools to navigate, understand, and internalize the structure of scholarly knowledge and expertise," in *New directions in cognitive information retrieval*, pp. 187–208, Springer.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999), "Data clustering: a review," *ACM computing surveys (CSUR)*, 31, 264–323.

Kotsiantis, S. and Pintelas, P. (2004), "Recent advances in clustering: A brief survey," *WSEAS Transactions on Information Science and Applications*, 1, 73–81.

Lacroix, E.-M. and Mehnert, R. (2002), "The US National Library of Medicine in the 21st century: expanding collections, nontraditional formats, new audiences," *Health Information & Libraries Journal*, 19, 126–132.

Lu, Z. (2011), "PubMed and beyond: a survey of web tools for searching biomedical literature," *Database*, 2011, baq036.

Lu, Z., Wilbur, W. J., McEntyre, J. R., Iskhakov, A., and Szilagyi, L. (2009), "Finding query suggestions for PubMed," in *AMIA Annual Symposium Proceedings*, vol. 2009, p. 396, American Medical Informatics Association.

Manning, C. D. (2011), "Part-of-speech tagging from 97% to 100%: is it time for some linguistics?" in *Computational Linguistics and Intelligent Text Processing*, pp. 171–189, Springer.

Mosa, A. S. M. and Yoo, I. (2013), "A study on PubMed search tag usage pattern: association rule mining of a full-day PubMed query log," *BMC medical informatics and decision making*, 13, 8.

NCBI (2013), "Database resources of the National Center for Biotechnology Information," *Nucleic Acids Research*, 41, D8–D20.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2009), "Distributed algorithms for topic models," *The Journal of Machine Learning Research*, 10, 1801–1828.

Nourbakhsh, E., Nugent, R., Wang, H., Cevik, C., and Nugent, K. (2012), "Medical literature searches: a comparison of PubMed and Google Scholar," *Health Information & Libraries Journal*, 29, 214–222.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999), "The PageRank citation ranking: bringing order to the Web." .

Pedersen, T., Banerjee, S., McInnes, B. T., Kohli, S., Joshi, M., and Liu, Y. (2011), "The Ngram statistics package (text:: nsp): A flexible tool for identifying ngrams, collocations, and word associations," in *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pp. 131–133, Association for Computational Linguistics.

Pitman, J. and Yor, M. (1997), "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator," *The Annals of Probability*, pp. 855–900.

Rogers, F. B. (1964), "The development of MEDLARS," *Bulletin of the Medical Library Association*, 52, 150.

Salton, G. and Michael, J. (1983), "McGill," *Introduction to modern information retrieval*, pp. 24–51.

Shariff, S. Z., Bejaimal, S. A., Sontrop, J. M., Iansavichus, A. V., Haynes, R. B., Weir, M. A., and Garg, A. X. (2013), "Retrieving clinical evidence: a comparison of PubMed and Google Scholar for quick clinical searches," *Journal of medical Internet research*, 15.

Shneiderman, B. (1996), "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 336–343, IEEE.

Steyvers, M. and Griffiths, T. (2007), "Probabilistic topic models," *Handbook of latent semantic analysis*, 427, 424–440.

Team, N. (2006), "Network Workbench Tool. Indiana University, Northeastern University, and University of Michigan," .

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006), "Hierarchical dirichlet processes," *Journal of the american statistical association*, 101.

Thiele, R. H., Poiro, N. C., Scalzo, D. C., and Nemergut, E. C. (2010), "Speed, accuracy, and confidence in Google, Ovid, PubMed, and UpToDate: results of a randomised trial," *Postgraduate medical journal*, 86, 459–465.

Tryon, R. and Bailey, D. E. (1970), "Cluster Analysis," .

Wang, C., Paisley, J. W., and Blei, D. M. (2011), "Online variational inference for the hierarchical Dirichlet process," in *International conference on artificial intelligence and statistics*, pp. 752–760.

Wang, X., McCallum, A., and Wei, X. (2007), "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pp. 697–702, IEEE.