Dissertation

UNSUPERVISED INDEXING OF MEDLINE ARTICLES THROUGH GRAPH-BASED RANKING

By

Jorge R. Herskovic, MD, MS

December 17, 2008

APPROVED:

Elmer V. Bernstam, MD, MSE, MS (chair)

Jack W. Smith, MD, PhD

M. Sriram Iyengar, PhD

Devika Subramanian, PhD

UNSUPERVISED INDEXING OF MEDLINE ARTICLES THROUGH GRAPH-BASED RANKING

A

DISSERTATION

Presented to the Faculty of
The University of Texas
School of Health Information Sciences
at Houston
in Partial Fulfillment
of the Requirements

for the Degree of

Doctor of Philosophy

by

Jorge R. Herskovic, MD, MS

Committee Members:

Elmer V. Bernstam[1], MD, MS, MSE (Chair)
Jack W. Smith[1], MD, PhD
M. Sriram Iyengar[1], PhD
Devika Subramanian[2], PhD

[1]The University of Texas School of Health Information Sciences at Houston
[2]Rice University Department of Computer Science

DEDICATION

I dedicate this work to those who waited patiently

while I researched, wrote, and created it: my family.

Pilar and Emma, I love you. Thank you for your support,

patience, humor, and for coming to this adventure with me.

ACKNOWLEDGMENTS

Many people went above and beyond their duties to make this work possible. I am deeply and forever grateful to all of them. The staff at the NLM, most of whom I never met but who I pestered with questions, asked for help and technical support when servers died or MetaMap or SEMREP failed, were all professional and extremely kind. Jim Mork and Olivier Bodenreider both went the extra mile to help me, giving me insight, data (notably a copy of Restrict To MeSH), support, suggestions, and kind encouragement. I had the pleasure of meeting Dr. Bodenreider at AMIA this year, and I was extremely happy to finally thank him in person.

I couldn't have asked for a better doctoral committee. Each and every member took time out of their hectic schedules to hear me out, offer encouragement, and when the situation required it, sharp questions that made me doubt myself and think twice. Dr. Sriram always offered a sympathetic ear, and encouraged me to think mathematically. For a 30+ MD, this wasn't always easy, but he was very patient. Dr. Jack Smith questioned incisively, and with an extraordinary ability to untangle complex issues and get to the heart of the problem. Dr. Devika Subramanian helped me through thorny performance issues and offered fantastic insight on graph-based ranking, while maintaining a positive attitude despite my sometimes gloomy mood. And finally, my committee chair, mentor, and friend Dr. Elmer Bernstam, who kept his calm despite my

mistakes, always had his door open for me, heard me out whenever I wanted, and went out of his way to support and help me every single time I asked him to. He will be proud to read that he was tough. Thank you for always pushing me to do better. I couldn't have done this without you.

The impact and drain of my PhD training on my family was tremendous. Pilar, thank you for moving to another country for me. Thank you for our wonderful Emma. Thank you for keeping me on track, and thank you for your excellent advice as a colleague and friend. There is no way I would have finished my PhD without your help. My PhD thesis is dedicated to you and Emma.

Table of Contents

Chapter 1

Indexing and the biomedical literature

index

noun (pl. -dexes or esp. in technical use -dices)

• *an alphabetical list of names, subjects, etc., with references to the places where they occur, typically found at the end of a book.*

• *an alphabetical list by title, subject, author, or other category of a collection of books or documents, e.g., in a library.*

• *Computing a set of items each of which specifies one of the records of a file and contains information about its address.*

verb [trans.]

• *record (names, subjects, etc.) in an index: the list indexes theses under regional headings.*

• *provide an index to.*

(Excerpted from the *New Oxford American Dictionary*)

Indexing

Indexing is a human activity whose origins are lost in time. The earliest linguistic precursors of the term "index" mean "point to," or "call attention to." The figurative meaning of the term itself is very old, dating at least to ancient Rome. Small slips of parchment called "index" were routinely attached to scrolls, noting the title and author so that the scroll itself would not need to be pulled off the shelf and opened for inspection.

The use of these indexes to hold the title of a scroll led to the use of "index" to refer to the title of a book or scroll. In approximately the first century A.D. the word "index" (and, probably, the physical "index" attached to the scroll as well) started to refer to a short list of chapters. Sometimes these tables of contents included brief abstracts of the chapters, which in turn led to "index" being adopted as a term for a bibliographic list or catalog. Hans Wellisch narrates, "Seneca (Epistulae, 39) tells a certain Lucilius, who had asked him to suggest suitable sources for an introductory course in philosophy: 'Sume in manus indicem philosophorum' (Pick up the list of philosophers), referring to a list of authors' names and the topics on which they had written." (Wellisch, 1991)

Tables of contents and back-of-the-book indexes as we know them today appeared much later. They had to wait for the appearance of the printing press in the fifteenth century, because by their nature they required consistent page numbers. Indexes on reference books appeared shortly after the Gutenberg printing press made its debut (Wellisch, 1991).

Despite its age and tradition, indexing still is a difficult task that requires experienced and specialized personnel. It is slow, laborious, and expensive. Current computerized

tools help, but are not yet able to approach human performance. In this thesis, I propose that a new understanding of the structure of scientific documents together with current advances in graph theory and text processing can be used to improve automated, unsupervised indexing. I applied this understanding to the field of biomedicine, where fast, accurate, and consistent automated indexing may help translational research efforts, help develop better information retrieval tools, and make high-quality literature more accessible to clinicians.

Creation of the National Library of Medicine, the Index Medicus, and MEDLINE

The National Library of Medicine

The National Library of Medicine (NLM) started out as the bookshelf of Surgeon General of the United States Army Dr. Joseph Lovell, appointed in 1818. In 1836 the budget request included, for the first time, a request for "medical books for [the] office." This prompted the NLM to choose, retroactively, 1836 as the year of its own birth. The Surgeon General's bookshelf grew slowly at first, and was catalogued for the first time in 1840. It contained 134 titles.

At the end of 1864 the library had grown to 2,100 volumes. Dr. John Shaw Billings, a young army surgeon, was appointed in 1864 to the Surgeon General's office. It was, apparently, a fortuitous appointment. Dr. Billings was an avid bibliophile who, by 1867, took over acquisitions for the burgeoning library and longed to make it as complete as possible. He acquired, traded, and begged for books at an astonishing pace. Barely a year later, the collection had more than tripled to 7,000 volumes.

Dr. Billings started collecting periodicals, and went to great lengths to find older issues

to complete his collection. He also issued catalogs regularly, and by the mid-1870s had also started a card file to serve both as a source of catalog information and as a repository of the latest bibliographic information. The first catalogs were inspired by catalogs in other medical libraries, and listed books by subject and author. He was, however, unsatisfied with the bibliographic usefulness of the catalogs. Probably inspired by European abstracting and indexing periodicals, in 1874 he started incorporating the journal articles in the library by subject to his card file. He recruited the help of bored army doctors in the frontier to do so.

Dr. Billings thought that a unified catalog was necessary to achieve the library's potential. Knowing that it would be expensive, he bound a part of his card file. He included all cards from "Aabec" to "Air," and called the volume *Specimen Fasciculus of a Catalogue of the National Medical Library*. The *Specimen Fasiculus* was the first precursor of what was to become MEDLINE. With this *Specimen Fasciculus* in hand as a proof of concept, Dr. Billings went to Congress and secured funding to catalog the rest of the Library's collection. The complete catalog may have cost as much as a post office building.

Even Dr. Billings underestimated the size of the task he had undertaken. The first volume of the *Index-Catalogue of the Library of the Surgeon General's Office, United States Army* (A-Berlinski) appeared in 1880. The entire *Index-Catalogue* was expected to fill eight volumes, but wasn't completed until 1895. By the time the task was done, the National Library of Medicine had published 16 volumes. The second series of volumes were published between 1896 and 1916.

The Index Medicus and MEDLARS

Dr. Billings quickly realized that the catalog was incapable of dealing with current output quickly. There was an almost 20-year gap before a volume was reissued and new entries could be included in the *Index-Catalogue*. Dr. Billings instituted, together with publisher F. Leypoldt, the *Index Medicus* in 1879. The *Index Medicus* was a monthly publication that classified journal articles and books by subject. It also included an author index.

The *Index Medicus* was published between 1879 and 1899. It was prepared after hours at the library through the same work that produced the *Index-Catalogue*. Its publication ceased because it was not commercially viable. It did, however, establish itself as the most comprehensive guide to the medical literature available anywhere.

Dr. Billings retired from the army and the Library, but kept a post on the board of the Carnegie Institution of Washington. With the Carnegie Institution's support, the *Index Medicus* resumed publication in 1903. In 1930 it merged with a similar publication from the American Medical Association (AMA) and became the *Quarterly Cumulative Index Medicus*. The AMA published it until 1959, when responsibility passed backed to the Library. The Library restored the name *Index Medicus* and started publishing it monthly. The NLM ceased publishing the *Index Medicus* in 2004.

In 1960 Dr. Frank Rogers, Director of the Library, installed a computerized system with a state-of-the-art photocomposition machine to produce the *Index Medicus*. This system permitted the Library to substantially increase the number of journals the library could index, produce custom one-time or recurring bibliographies on specific subjects, or even produce bibliographies to answer specific researchers' needs. The search service was

called the Medical Literature Analysis and Retrieval System (MEDLARS) and was a success. At the time of its introduction, it was the largest information retrieval system in existence.

MEDLARS' success came with its own share of problems. Demand for its services was so great that it threatened to overwhelm the system. To alleviate the demand, the NLM decentralized MEDLARS by sending tapes with copies of its databases to other libraries, and led indirectly to the Medical Library Assistance Act of 1964, which enabled the construction of a regional medical library system (Blake, 1986; U.S. National Library of Medicine, 2004d).

In 1971, MEDLARS was brought online, with data from the *Index Medicus* starting in 1966. The online version of MEDLARS was called MEDLINE (U.S. National Library of Medicine, 2003).

MEDLINE and its growth

MEDLINE is now the premier index of biomedical articles. It currently contains more than 16,000,000 references from more than 5,000 biomedical journals, and grows continuously, at an ever-increasing rate. Over 670,000 new references were added in 2007 alone (U.S. National Library of Medicine, 1999). Finding the correct references among this mass requires good search tools and high-quality indexing that describes the references precisely.

All MEDLINE entries corresponding to journal articles are indexed by hand using a purpose-built and continually maintained vocabulary called the Medical Subject Headings (MeSH). The use of a controlled vocabulary like MeSH allows users to retrieve documents more efficiently, thanks to several advantages over using free text.

In MeSH there are unambiguous single correct ways of describing a concept. The hierarchical IS-A structure of MeSH also allows broadening or restricting retrieval in intuitive ways. For example, a search for "myocardial ischemia" will also retrieve references indexed under "myocardial infarction," since the former is a parent of the latter in the MeSH tree (U.S. National Library of Medicine, 1999). In other words, the MeSH hierarchy embodies the knowledge that a "myocardial infarction" is a "myocardial ischemia."

To achieve high-quality indexing, the National Library of Medicine (NLM) maintains a highly trained professional staff. The NLM staff includes indexers who perform the actual indexing process, and MeSH staff that maintain and update the vocabulary.

Indexing of MEDLINE articles by hand can be traced back more than 100 years. It is, by any measure, a success. MEDLINE is an established part of the biomedical literature ecosystem, and its importance cannot be overstated. MEDLINE searches form the basis of meta-analyses that are the backbone of Evidence-Based Medicine (EBM). MEDLINE is a regular part of biomedical research, the start of and final repository of almost every research project.

Despite MEDLINE's importance, and its' clear success, it is not perfect. While human indexing is flexible, adaptable, and the current gold standard, it suffers from several important deficiencies. Perhaps the most obvious one is its cost. Maintaining an in-house staff to perform indexing is expensive. Actual figures are hard to obtain, but already in 1990 the U. S. National Library of Medicine (NLM) spent more than $2,000,000 on 44 full time equivalent employees to index MEDLINE (Hersh, Hickam, Haynes, & McKibbon, 1994). While not directly comparable, the budget request for 2009

for all health information services (including MEDLINE, among other NLM databases) is $107,382,000 (National Institutes of Health, 2008).

Indexing is also slow; processing a reference may take weeks, during which the reference is only available with arbitrary publisher-supplied terms. Due to the time constraints and large amount of material to process, reading the title and abstract of an article constitutes the bulk of the indexing process. The entire article is not used regularly, although certain sections (like figure captions) are regularly skimmed. Unfortunately, abstracts and titles may not represent the contents of articles accurately (Dupuy, Khosrotehrani, Lebbe, Rybojad, & Morel, 2003; Pitkin, Branagan, & Burmeister, 1999). The effect of this discrepancy on the quality of NLM indexing is unknown, but at least some NLM indexers do not trust indexing software that does not use the full text of the article (Ruiz & Aronson, 2007). There is an inherent tension between the need for indexing productivity and accuracy.

Finally, human indexers are inconsistent. There is one large study on MEDLINE indexing consistency, published in 1983 after an involuntary error set up a natural experiment when several hundred articles were indexed twice. In this study, Funk and Reid found that indexing consistency varied from 74.7% at a very high conceptual level (when comparing checktags) to 33.8% when comparing detailed concepts (MeSH heading/subheading combinations) (Funk, Reid, & McGoogan, 1983). Unfortunately, Funk and Reid's study does not account for semantic similarities between indexing terms. Under their model, "myocardial infarction" is just as different from "myocardial ischemia" as it is from "colon cancer," which is an evaluation weakness this thesis addresses.

The structure of scientific writing

Despite the enormous importance of scientific writing to our civilization, few efforts have been made to study or analyze it consistently. Some authors turn to measure writing and scientific output, a discipline called bibliometrics (or scientometrics when applied strictly to science). One of the foremost experts in scientometrics is Eugene Garfield, creator of the Science Citation Index and the Journal Impact Factor. Yet Garfield knew that quantifying scientific writing and citation was very different from analyzing or understanding it (Garfield, 1972, 2007).

Frederick Suppe is a philosopher who analyzes the structure of scientific writing. In a seminal 1998 paper he argued that scientific articles are rigidly structured. Scientific articles present an argument that tries to advance one or more claims. Since journal space is a scarce resource, there are constraints on the amount of text articles can consume. Scientific articles must therefore use every paragraph and sentence to advance their claims (Suppe, 1998).

An intuitive consequence of this theory is that scientific papers essentially build a network of interrelated concepts. They advance their claims by stating facts about them, about concepts related to the facts, or about relationships between these. The most important of these networks' concepts will be the papers' most important concepts, surrounded by the ones most important to the argument. These will, in turn, be connected to the concepts that support them, and so on. Thus, by virtue of Suppe's theory, a network of concepts can be derived from any scientific paper, and it will represent its contents.

Semantic abstraction graphs and graph-based ranking algorithms

A tremendous amount of research into the structure of all kinds of networks has been done in the last two decades. Networks are generally analyzed through a branch of mathematics called graph theory. A graph is "a diagram consisting of a set of points together with lines joining certain pairs of these points" (Bondy & Murty, 1976). The points and lines in graph theory are commonly called "nodes" and "edges" respectively. This thesis will adopt the terms "nodes" and "edges" to refer to graph components.

A particular kind of graph that has been the subject of study at the NLM is the Semantic Abstraction Graph (SAG). Semantic Graphs represent words in a piece of text as nodes, and relationships between these words as edges connecting the nodes. A SAG is a specialized kind of Semantic Graph. A SAG represents, instead of words, concepts from a piece of text as nodes and relationships between these concepts as edges in the graph, and is built around a user-supplied central query concept (Fiszman, Rindflesch, & Kilicoglu, 2004).

Graph theory and graph analysis have been extremely useful in dealing with many kinds of human knowledge networks. Perhaps the best-known example of graph analysis is Google (http://www.google.com), an Internet search engine company. Google indexes the World Wide Web (WWW) and provides search results to user-entered queries. Google internally models the WWW as a graph: web pages are represented as nodes, and the hyperlinks that connect web pages are the edges. The graph is analyzed using an algorithm called PageRank.

PageRank is an iterative algorithm that computes a formula (Figure 1) over the entire

graph. The PageRank value for a web page converges on the probability that a person clicking hyperlinks at random will end up on that page. This is called the random surfer model (Page, Brin, Motwani, & Winograd, 1998). PageRank has been successfully applied to graph models of other networks beside the WWW, including the citation network of biomedical literature (Bernstam et al., 2006), analysis of social networks (Pujol, Sanguesa, & Delgado, 2002), and text summarization through selection of important sentences (Mihalcea, 2004). PageRank can be considered a general algorithm that will rank nodes in a graph based on their relative importance as established by the set of edges.

$$R'(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} + cE(u)$$

Figure 1 - The PageRank formula

R is the PageRank, R' is the new PageRank, N is the number of outgoing links, v is the recommender and u is the recommendee. B is the set of incoming links. c is a decay factor, and E is a baseline PageRank for "rank sinks" like closed loops.

There are other graph analysis algorithms besides PageRank. For example, HITS models a graph as a set of interconnected hubs and authorities, which it can discover iteratively (Kleinberg, 1999). TextRank is a PageRank derivative that is tailored specifically to work on undirected graphs (Mihalcea, 2004).

In this thesis, I apply Suppe's theory of the structure of scientific papers to build semantic abstraction graphs based on the concepts in biomedical articles. I then apply graph-based ranking algorithms to rank the concepts in the SAG. From this ranked list of concepts I obtain a set of indexing terms.

Research hypothesis

I propose that ranking the concepts in a semantic abstraction graph using graph-based ranking

algorithms will yield the most important concepts of a biomedical scientific article.

In support of this hypothesis, I make three major claims: (1) that it is possible to build, in an unsupervised way, semantic abstraction graphs from scientific articles, (2) that ranking the concept nodes in these SAGs yields the most important concepts in an article, and (3) that this approach, being grounded in a theory of the structure of scientific writing, performs better than the current state of the art in biomedical indexing (MTI). I will validate these claims by building a prototype system that is able to construct these graphs and rank the nodes in them. I will demonstrate experimentally that the ranked concepts are meaningful by comparing them to concepts selected by human indexers, and will compare the performance of the prototype system to MTI.

Building graphs from scientific articles

I claim that it is possible to build unsupervised Semantic Abstraction Graphs from the contents of a scientific article. Useful SAGs that properly represent Suppe's theory of scientific writing should have the following properties (see Evaluation below, Chapter 3, and (Bondy & Murty, 1976; Dhyani, Ng, & Bhowmick, 2002)):

1. Be highly connected.
2. Have identifiable important nodes.

The important nodes in a graph correspond to the most important concepts in an article

I claim that the most important nodes in these SAGs will correspond to the most important concepts in scientific articles. Since identifying the most important concepts in articles is analogous to the indexing task, the important concepts should correspond

to the indexing terms for the article. In particular, they should agree with the indexing as much as human indexers agree with one another.

Such a system should outperform MTI

Indexing terms based on the full text of the article should be, both intuitively and according to the NLM (Ruiz & Aronson, 2007), more accurate than those generated by MTI. When MTI has been experimentally applied to full text articles, it performed worse than when applying an optimal selection strategy, i.e. hand-selecting the parts of articles that yield the best results, and not better than when using just the title and abstract (Gay, Kayaalp, & Aronson, 2005). This is unsurprising; MTI was developed over years to leverage the title and abstract of an article, and is not designed to work on full text. A system built from the ground up to leverage full text intelligently should perform better, since MEDLINE indexers also have access to the full text of the article when choosing indexing terms.

MEDRank

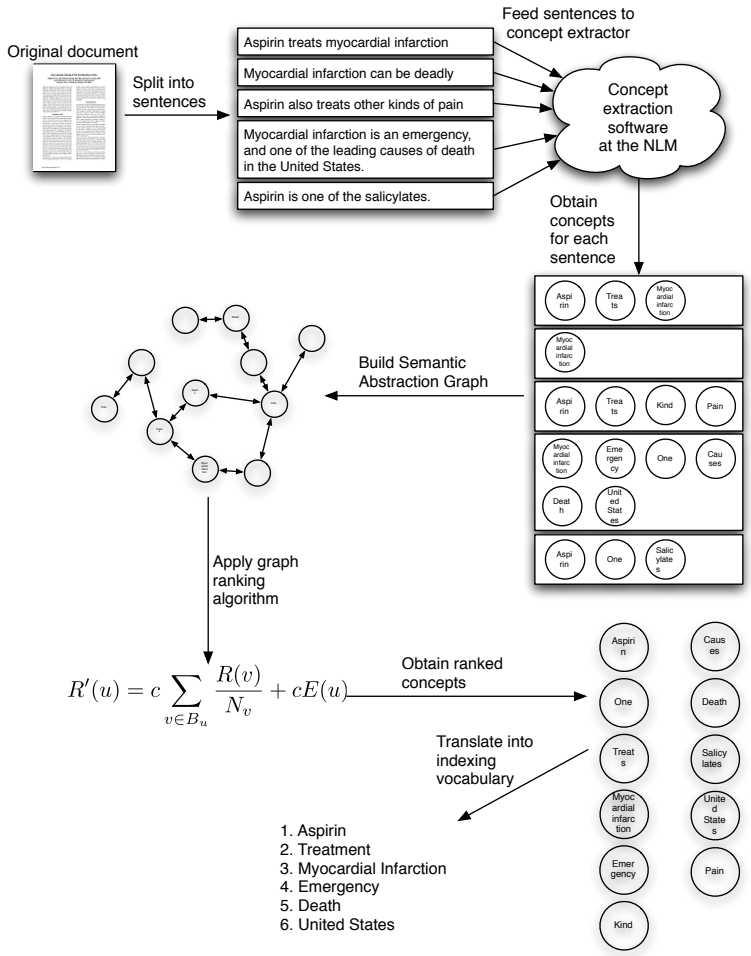

Figure 2 - Basic architecture of MEDRank. The character sets and concept representations highlight that the MEDRank process is independent of the ontology, language, and vocabularies.

To study and prove my claims, I designed and built MEDRank. MEDRank leverages Suppe's theory of the structure of scientific papers by considering the concepts in a paper as part of a logically ordered collection of statements designed to advance a central claim or claims. As Suppe's theory requires, no concept is discarded, and no assumptions about the role or article segment (i.e. introduction, methods, results, etc.) are used. Instead, the article author's choice of concepts and the relations between them is considered meaningful, and used to determine the indexing terms. MEDRank also fulfills the three desirable traits of an automated indexer: it produces results of comparable quality to human indexers, it uses the full text of the articles, and it is potentially generalizable to other domains.

MEDRank generates SAGs based on full text biomedical articles. It then ranks the

concepts in these SAGs using graph analysis algorithms, and compares the resulting indexing terms to MEDLINE records.

The basic architecture of MEDRank is shown in Figure 2. MEDRank processes documents by splitting them into individual sentences. Each sentence is fed separately to an external concept extraction stage that returns an ordered list of concepts for each sentence. MEDRank can use a list of concepts and infer relationships between them, or can accept a list of relationships between concepts. MEDRank uses the concepts and relationships to generate SAGs. It then ranks the concepts in the SAG with a graph-based ranking algorithm. MEDRank then translates the ranked list of concepts into the destination indexing vocabulary.

For my research I used the NLM's Unified Medical Language System (UMLS) concepts, its Semantic Representation (SEMREP) as a concept extractor and the destination vocabulary was always MeSH. These choices are not fundamental to the design of MEDRank, and the concept ontology, extractor, and indexing vocabulary could be easily replaced with others and the system repurposed for other uses.

Evaluation

To evaluate each of my three claims, I constructed graphs and conducted experiments comparing MEDRank output both to human indexer output and to MTI's output.

Evaluating the possibility of constructing SAGs from scientific articles

Even though it is always possible to run an algorithm like the one outlined above, the results will not necessarily be meaningful. For the results to be meaningful, and in concordance with Suppe's theory of the structure of scientific papers, graphs produced

by such an algorithm will have to fulfill the requirements outlined above:

1. Be highly connected.
2. Have identifiable important nodes.

Highly connected graphs are graphs in which most nodes are connected to most other nodes, either directly or indirectly. In other words, graphs with many islands are not highly connected (Figure 3). Graph connectivity can be measured directly with a graph metric known as compactness. Graph compactness varies between 0 (no node can reach another through an edge) and 1 (all nodes can reach other nodes through edges either directly or indirectly) (Dhyani et al., 2002).

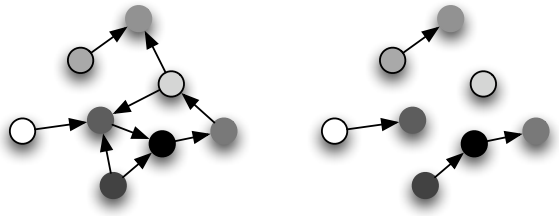

Figure 3 - Highly connected (left) and disconnected (right) versions of a graph with the same nodes

I will characterize the SAGs generated by MEDRank on their compactness. The true test, however, of the quality of the graphs will be their suitability to the task: if graphs do not have identifiable central nodes (because, for example, they are extremely dispersed) they will be unsuitable to the task. I will therefore study the distribution of ranking scores (i.e. the output of the graph-based ranking algorithm) to ascertain whether it contains easily identified central nodes or, for example, all nodes achieve the same ranking score.

Evaluating whether the output is comparable to human indexer output

MEDLINE is the clear gold standard against which to compare the output of any

biomedical indexer. In particular, comparing to MEDLINE is desirable because:

1. MEDLINE uses professional human indexers
2. The records are freely available in an electronic format
3. It is widely used
4. There is a comparison of inter-indexer concordance that provides a measurement baseline (Funk et al., 1983)

Unfortunately, and as desirable as MEDLINE appears initially, it may be less than compelling as a gold standard. Indexers working on the same article can legitimately disagree on what an article is about. Actual agreement rates vary between approximately 30 and 70%, with the highest rates corresponding to the most general MeSH terms, called checktags. Checktags are used to index very general concepts like "humans."

The MEDLINE indexing process is also more complex than reading the articles and selecting the best terms. Checktags, for example, are used for some extremely common search filters that are indispensable to MEDLINE users. The indexers are therefore instructed to assign checktags regardless of whether the article is about the topic, or merely mentions it. This enables MEDLINE users to fetch, for example, all articles that involve humans.

MEDLINE indexers are also required to select the "major topics" in a paper, and index them as major headings.

```
MH  - Algorithms
MH  - Information Storage and Retrieval/*statistics & numerical data
MH  - Internet
MH  - Medical Subject Headings/utilization
MH  - PubMed/*statistics & numerical data
```

Figure 4 - Part of a MEDLINE record showing the MeSH headings for an article. Asterisks are major headings.

Major headings are highlighted with an asterisk in MEDLINE records (Figure 4). There are some restrictions on what MeSH terms can be designated major headings; among

them, checktags can never be selected as major headings. Legitimate disagreement of what constitutes an article's major topic is also possible. For example, I am the first author of the article whose MEDLINE record is displayed in Figure 4, and I do not consider "statistics & numerical data" the only major topic in it.

In summary, the MEDLINE indexing process deviates from an ideal in which only the most important concepts in an article are selected as indexing terms. Despite these limitations, MEDLINE is still the gold standard for evaluating automated indexers in the biomedical domain (Aronson, 2001; Aronson, Mork, Névéol, Shooshan, & Demner-Fushman, 2008; Hersh et al., 1994; Kim, Aronson, & Wilbur, 2001).

Semantically aware comparisons

Traditional information retrieval evaluation techniques model the semantically rich indexing task poorly. In particular, they do not distinguish between terms that match partially and terms that do not match at all. For example, "myocardial ischemia" is a parent of "myocardial infarction" in the MeSH hierarchy, and both terms are closer in their everyday meaning than "myocardial infarction" and "conjunctivitis," yet substituting "myocardial ischemia" for "myocardial infarction" is as much a mismatch as substituting "conjunctivitis." Quantifying matches between terms allows me to evaluate indexing quality better (Olivier Bodenreider, 2008). I describe and use a semantically aware version of the vector cosine comparison based on (Medelyan & Witten, 2006) to measure indexing performance.

This semantically aware vector cosine comparison (SAVCC) is based on the idea that terms in a vocabulary can be semantically related to one another. In particular, for a vocabulary V with terms numbered sequentially from 0 to n, there is an n x n matrix M

that describes the relationships between all terms i and j by having M_{ij}=1 if the terms are related, and 0 otherwise. M is then applied to one of the vectors, which produces a vector that takes the relationship between terms into account. A VCC calculation on the modified vector is therefore a SAVCC (Medelyan & Witten, 2006). The original SAVCC takes into account both uni- and bi-directional relations between concepts, but in MeSH all relations are bidirectional. The SAVCC presented here is a derivative of Medelyan's that uses only a single matrix, and takes into account not only the existence of relationships between terms but also the strength of each relationship.

Evaluating performance against MTI

MTI is the tool of choice at the National Library of Medicine for automated indexing. It currently indexes conference proceedings unattended, and assists MEDLINE indexers with the regular indexing process. Under some experimental circumstances, MTI can achieve the same level of consistency with MEDLINE indexers as the inter-indexer consistency described in (Funk et al., 1983) (Névéol, Shooshan, Mork, & Aronson, 2007).

MTI is currently used as an interactive aid to the MEDLINE indexing process. Not all indexers use it, but most of them do. The number of articles in MEDLINE indexed with the help of MTI is unknown. It is therefore impossible to quantify the influence MTI suggestions actually have on the chosen indexing terms. However, even if unquantifiable, the indexing process strongly suggests that the MEDLINE indexers' terms have a built-in bias for MTI suggestions (Ruiz & Aronson, 2007). In other words, if two equally plausible but mutually exclusive indexing terms exist for an article, but only one of them is suggested by MTI, I believe that it is more likely for the one suggested by MTI to be in the MEDLINE record. From now on, I will therefore refer to

the contents of the MEDLINE record as "MEDLINE (with MTI)."

The practical upper bound of algorithm performance is the level of human-human agreement. Since MTI is already close to the level of the same level of agreement with human indexers as inter-indexer agreement, any further gains will be between MTI's performance and inter-indexer agreement, are therefore marginal. In fact, MTI performance has not improved substantially in the last years, and falls when considering full text articles. Surpassing MTI's performance when compared to MEDLINE (with MTI) records will potentially be even harder due to the inherent bias in the indexing process.

Despite this, I will demonstrate that an indexing solution that applies Suppe's theory and is implemented through graph-based ranking algorithms outperforms MTI, even when compared to MEDLINE (with MTI) records.

Chapter 2

Historical and current approaches to automated indexing

Automated indexing

Given the expense and difficulty of indexing manually, its poor consistency (Funk et al., 1983; Hersh et al., 1994), and the potential volume of indexing work (over 670,000 articles yearly at the NLM, for example (U.S. National Library of Medicine, 1999)), automated indexing solutions are extremely attractive. Automated indexing is an integral part of many computerized information retrieval systems. Like its manual counterpart, automated indexing assigns metadata to documents (Hersh, 2003), and attempts to assign metadata that will facilitate retrieving documents. Automated indexing can be divided into two large categories: statistical or semantic automated indexing.

Statistical automated indexing

The simplest automated indexes are based on statistics like word frequency counts. Words, in this context, are strings of alphanumeric characters separated by "whitespace" (spaces, tabulation marks, and punctuation). Frequency counts collect all instances of a single word in a document, and consider the most frequent words as indexing terms. Since most English text contains many repetitions of words that convey little or no content, like "the," "of,"

$$TF(i,j) = \log(\text{frequency of i in j}) + 1$$

$$IDF(i) = \log\left(\frac{\text{number of documents in corpus}}{\text{number of documents that contain i}}\right) + 1$$

$$WEIGHT(i,j) = TF(i,j) * IDF(i)$$

Figure 5 - TF*IDF weight computation for a term i in a document j.

"to," among others, these stop words are removed before computing frequency counts (Hersh, 2003).

Another simple yet surprisingly effective statistical technique is using weighted term frequencies relative to frequencies in a corpus. This technique is called Term Frequency*Inverse Document Frequency and abbreviated TF*IDF. TF*IDF weights the frequency of a term in a document by the inverse of the number of documents that contain the term (Figure 5) (Hersh et al., 1994). In other words, terms with low presence in the corpus but high frequency in one particular document will have high TF*IDF weights, and will be chosen as indexing terms for that document.

SAPHIRE

SAPHIRE is a good example of a statistical system. It was created by William Hersh, a pioneer in biomedical information retrieval. SAPHIRE detected Unified Medical Language System (UMLS) concepts in each document and then ranked the concepts using TF*IDF (Figure 5). These ranked concepts were used as indexing terms.

Concepts detected in users' queries were compared to indexing concepts in order to retrieve articles. SAPHIRE was evaluated in user studies, and it performed "slightly better than novice physicians using MEDLINE, but somewhat worse than expert physicians, although none of the differences were statistically significant." Hersh judged SAPHIRE's performance lackluster, and attributed this to the construction of the sample, lack of full-text indexing but, over all, to the poor coverage of the UMLS. At the time, the UMLS could not code for approximately 25% of the medically significant noun phrases in the study. He also blamed the inability of the system to act like human indexers and "choose indexing terms focused on the main topics" (Hersh et al., 1994).

Latent semantic indexing

Semantic algorithms do not truly understand the text they process, but they attempt to use knowledge about meaning to improve indexing quality. A statistical technique that discovers "concepts" through word co-occurrence in a corpus is called Latent Semantic Indexing (LSI). While LSI is a statistical indexing technique, it can approximate meaning and provide extremely good results. LSI performs very well in situations where high recall is desirable (Manning & Schütze, 1999).

LSI uses Single Value Decomposition (SVD) to reduce the dimensionality of a corpus to common features, and can detect common, implicit patterns in text (Manning & Schütze, 1999). For example, if "myocardial infarction" and "aspirin" co-occur frequently in a set of documents, LSI will index them under a single feature. Retrieval systems looking for "aspirin" or for "myocardial infarction" will then return the same documents, exposing the original relationship in the corpus. This ability of the algorithm to expose previously unknown relationships in text earned it the name Latent Semantic Indexing.

The ability of LSI to uncover hidden connections makes it very useful for indexing free text collections, and some of its proponents believe that it is superior to the use of controlled vocabularies. However, since the main contribution of LSI to indexing is the discovery of these related concepts, it is incompatible with the use of controlled vocabularies. The historical nature of MEDLINE, the hundreds of thousands of people who know how to use it, and its continued use of MeSH make LSI unsuitable for indexing the biomedical literature in a way that is compatible with MEDLINE.

Semantic indexing techniques

Semantic indexers use knowledge to improve the indexing process. The addition of semantic content to the indexing process can be classified into two different subtypes: conceptual indexing and semantic indexing. Conceptual indexing captures concepts from a source vocabulary instead of terms from the documents, and has been used successfully in the legal field. Semantic indexing uses ontologies like WordNet (Fellbaum, 1998) as knowledge sources to perform word sense disambiguation (Mihalcea & Moldovan, 2000).

MeSH and the UMLS

MeSH is a controlled, hierarchical vocabulary developed and maintained by the NLM. Its first edition was in 1954, when it was called "Subject Heading Authority List" (U.S. National Library of Medicine, 2006a). Its 2008 edition, the latest for which this data is available, contains 24,767 unique descriptors and more than 172,000 supplemental records. It also has thousands of cross-references that point indexers and users to the actual MeSH term (U.S. National Library of Medicine, 1999). For example, "Acetylsalicylic Acid" is a cross-reference for Aspirin.

MeSH is continually revised, expanded, and corrected. Qualifiers and terms may merge, disappear, or be added. Whenever terms are removed or merged, the NLM updates existing MEDLINE records to reflect the changes in MeSH (U.S. National Library of Medicine, 2006a). Therefore every MEDLINE record is indexed using the latest edition of MeSH at all times.

The UMLS is a complex data and knowledgebase distributed by the NLM. It consists of three knowledge sources and software to manipulate them. The first knowledge source

is the Metathesaurus, which contains dozens of biomedical vocabularies like SNOMED, LOINC, and MeSH, among others, and information on the relationships between them. It accomplishes this by linking every vocabulary entry to a single "concept." Each unique UMLS concept has an identifier called a Concept Unique Identifier (CUI). The second knowledge source is the Semantic Network, which contains categories to classify every CUI and every possible relationship between CUIs into a consistent set of types. The third knowledge source is the SPECIALIST Lexicon, which is a dictionary of the English language supplemented with spelling variations and biomedical terms. Its main purpose is to make developing Natural Language Processing (NLP) software that uses the UMLS easier (U.S. National Library of Medicine, 2006b).

The Medical Text Indexer (MTI)

The scope, expense, and continuous growth of the indexing task led the NLM to look for automated alternatives or, at least, systems that could facilitate the indexers' job. One result of this ongoing effort is the Medical Text Indexer (MTI), an in-house system that embodies many of the heuristics that NLM indexers use. It has been developed over the past 10 years. It relies, among other inputs, on the past behavior of NLM indexers to assign MeSH terms to MEDLINE references. MTI is used to index conference proceedings without human intervention. It also suggests terms to human indexers that process journal articles.

MTI has its own set of limitations. It currently processes only the titles and abstracts of articles. This is a problem for the NLM because of two reasons. One is that full text has been electronically available to the NLM only recently, and is still not available for all articles. The second reason is that indexers do not trust MTI's output. In fact, studies of

indexer preferences showed that they would be more likely to use MTI if it incorporated full text indexing (Ruiz & Aronson, 2007). Unfortunately, MTI's experimental performance on full text is worse than its performance on titles and abstracts only (Gay et al., 2005)

MTI may be the best example of a semantic indexer. MTI combines both senses of semantic indexing described above: it is a conceptual indexer that produces MeSH terms as its output, and it performs word sense disambiguation using external knowledge. Part of MTI uses the SPECIALIST lexicon from the UMLS to generate candidate indexing terms based on the detected concepts. These candidate indexing terms are evaluated individually against the original phrase to decide whether they should be included in the final mappings. In other words, the knowledge embedded in the SPECIALIST lexicon enables MTI to improve concept recall by considering alternative concepts as potential mappings for phrases (Aronson, 2001).

MTI has three separate modules to generate indexing terms. These modules are MetaMap Indexing (MMI), PubMed Related Citations (REL), and Trigram Phrase Matching (Gay et al., 2005; Kim et al., 2001). Each module produces independent output, and the output of the three is combined to obtain a final set of ranked terms.

MetaMap Indexing (MMI)

MMI is part of the NLM MetaMap software. It generates indexing terms by parsing text into simple noun phrases. The SPECIALIST lexicon is then used to generate variants based on each noun phrase. All Metathesaurus strings that contain at least one of the variants are retrieved, and compared to the original text using a combination of evaluation metrics. The candidate strings are combined to form mappings, which are

"MetaMap's best interpretation of the original phrase" (Aronson, 2001). The mappings are ranked using a function called MMI. The top ranked mappings are used as indexing terms. The user must specify the number of indexing terms MTI outputs. The default is 25 indexing terms (U.S. National Library of Medicine, 2004b).

PubMed Related Citations (REL)

The PubMed Related Citations (REL) algorithm pulls indexer-assigned MeSH terms from documents related to the one being indexed. Document similarity is determined with the algorithm used for PubMed's Related Citations feature. In PubMed's Related Citations, similar documents are clustered according to common words in titles and abstracts, weighed using an IDF scheme. For details, see (Kim et al., 2001).

Trigram Phrase Matching

Trigram Phase Matching is "a method of identifying phrases that have a high probability of being synonyms" (Kim et al., 2001). It breaks down phrases of one to six words into sets of three character tokens. These tokens are then compared against all possible phrases in the UMLS. The highest scoring matches are paired to the original phrases. These pairs are then ranked by the frequency with which they appear in the original article (U.S. National Library of Medicine, 2004e). Trigram Phrase Matching is not currently used in production at the NLM.

Combining the output of the three modules

Two of the three modules, MMI and Trigram Phrase Matching, produce UMLS concepts as output, and REL produces MeSH terms. In order to combine the output of the three, the output from MMI and Trigram Phrase Matching is converted to MeSH

using an algorithm called Restrict to MeSH. Restrict to MeSH uses synonymy and hierarchical relations between concepts to obtain MeSH descriptions of UMLS concepts (U.S. National Library of Medicine, 2004c). These MeSH terms are considered mapping candidates and are further processed during ranking.

The output of the each module is weighted differently. By default, MMI, REL, and Trigram Phrase Matching weight 7, 2, and 0 respectively in production use. The combined terms are ranked using a ranking function that takes into account the confidence in the UMLS to MeSH translation, presence of the concept in the title of the article, and known co-occurrence with other MeSH terms. For further details, see (U.S. National Library of Medicine, 2004a).

MTI has two different modes: "DCMS" and "Gateway" processing. DCMS processing is used on articles that will go into MEDLINE. The MEDLINE indexers see the output of MTI in DCMS mode as term suggestions. Its authors do not consider MTI's ranking functions to have any intrinsic value and, as such, do not use them to limit the length of MTI's output. Despite the fact that indexers typically assign 10 to 12 MeSH headings to an article (Gay et al., 2005), MTI in DCMS mode always returns 25 terms.

MTI in Gateway mode is use for the unattended indexing of a collection of documents called the "Gateway abstracts collection." The Gateway abstracts collection is not part of MEDLINE.

Indexing evaluation

Precision and recall

Precision and recall are the most traditional information retrieval measurements. It is also possible to evaluate indexing quality using recall and precision by considering the indexing task a retrieval task. These measures require a gold standard. Recall measures how much of the gold standard the system retrieves, while precision measures how many results in the result set are part of the gold standard. There are other measures that combine these two into a single number. For example, the F measure is the harmonic mean of both. Precision and recall can also be given different weights depending on the task; F0.5, for example, weighs precision twice as much as recall (Baeza-Yates & Ribeiro-Neto, 1999) (Figure 6). To evaluate indexing quality using these measures, the terms chosen by one indexer are considered the gold standard and the other, the result set. I used the indexing terms from MEDLINE records as the gold standard.

Let N be the size of the gold standard
Let S be the total number of results
Let S_N be the number of results that were in the gold standard

$$R = \frac{S_N}{N}$$

$$P = \frac{S_N}{S}$$

$$F = \frac{PR}{P + R}$$

$$F_\alpha = \frac{(\alpha^2 + 1)PR}{\alpha^2 P + R}$$

Figure 6 - Traditional information retrieval measures

Hooper's Indexing Consistency

Hooper's Indexing Consistency (HIC) is a measure of agreement between two indexers that was used in the landmark

For two indexers, M and N,

$$IC = \frac{A}{A + M + N}$$

A = number of terms in common
M = number of terms used by indexer M but not N
N = number of terms used by indexer N but not M

Figure 7 - Hooper's Indexing Consistency Measure

study of inter-indexer agreement at the NLM (Funk et al., 1983). It is computed by dividing the terms in common between two indexers by the total number of unique terms assigned by both indexers (Figure 7).

$$Vocabulary = \{\text{dog,cat,rat,bird}\}$$
$$Indexer_1 = \{\text{dog,rat}\}$$
$$Indexer_2 = \{\text{cat,rat,bird}\}$$
$$I_1 = [1,0,1,0]$$
$$I_2 = [0,1,1,1]$$
$$VCC = \frac{I_1 \cdot I_2}{|I_1||I_2|} = \frac{1\cdot 0 + 0\cdot 1 + 1\cdot 1 + 0\cdot 1}{\sqrt{2}\sqrt{3}} = 0.408$$

Figure 8 - Example of a vector cosine comparison calculation

Semantically Aware Vector Cosine Comparison (SAVCC)

Another widely used measure is the Vector Cosine Comparison (VCC). The VCC computation for indexing measures is performed by creating vectors of the length of the vocabulary (i.e. if the vocabulary is 20,000 terms then each vector has 20,000 elements) for each indexer (Salton, 1963). Each vector is filled in with ones for the terms the indexer used, and zeros for the ones he or she did not (Figure 8). The normalized dot product of both vectors is the VCC. The VCC actually measures the proximity of both sets of terms in the semantic space defined by the vocabulary.

$$V = \text{MeSH}$$
$$s(m_i,m_j) = \text{the length of the shortest walk between } m_i \text{ and } m_j$$
$$\forall m_i \in V \text{ we define closeness, C such that :}$$
$$C(m_1,m_2) = \begin{cases} 0 \text{ if there is no walk between } m_i \text{ and } m_j \\ \dfrac{1}{e^{s(m_1,m_2)}} \text{ otherwise} \end{cases}$$
$$\text{Thus we define M as :}$$
$$M_{ij} = C(m_i,m_j)$$

Figure 9 – Computing the matrix M for a MeSH SAVCC. The C function is the inverse of the smallest distance between each pair of MeSH terms.

A major problem with the previous evaluation techniques is that they treat indexing terms as opaque strings of characters. Using traditional measures, if for the same article

one indexer selects "Myocardial infarction" but another selects "Coronary artery disease," they do not agree. This situation, however, is very different than if one chose "Myocardial infarction" and the other "Osteosarcoma." In the first case both indexers partially agree on the contents, while in the second they do not. Semantically aware indexing quality measures overcome this problem. These metrics use the relationships between indexing terms to calculate the degree of agreement between two terms (Medelyan & Witten, 2006) and then compute a modified VCC measure based on these.

$V = \{\text{dog},\text{cat},\text{rat},\text{bird}\}$

$\text{Hunts}(\text{dog},\text{cat}) = 1$

$\text{Hunts}(\text{cat},\text{rat}) = 1$

$\text{Hunts}(\text{cat},\text{bird}) = 1$

$\text{Hunts}(\text{dog},\text{rat}) = 1$

$M_{ij} = 1$ if both terms are related in any direction

$$M_{Hunts} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$SAVCC = \frac{I_1 \cdot (\alpha I_2 + (1-\alpha) M \times I_2)}{|I_1 \| \alpha I_2 + (1-\alpha) M \times I_2|}$$

Figure 10 - Semantically-Aware Vector Cosine Comparison on the same vectors as Figure 8

In particular, for a vocabulary V with terms v_i numbered sequentially from 0 to N, there is an N x N matrix M that describes the relationships between all terms i and j by having $M_{ij} = 1$ if the terms are related, and 0 otherwise (Figure 9). M is then applied to one of the vectors, which produces a vector that takes the relationship between terms into account. A VCC calculation on the modified vector is therefore a Semantically-Aware Vector Cosine Comparison (SAVCC) (Figure 10). The original SAVCC takes into account both uni- and bi-directional relations between concepts (Medelyan & Witten, 2006), but in MeSH all relations are bidirectional. The SAVCC presented here is thus a simplified version that uses only a single matrix. The coefficient α represents the weight of the traditional VCC computation. α=1 transforms SAVCC into traditional VCC, while

α=0 gives no weight to VCC. Following (Medelyan & Witten, 2006), I use α=0.65.

MeSH is hierarchical by nature. I use this organization to evaluate the quality of indexing taking into account semantic similarities. Further, a measure of semantic distance (J. R. Herskovic, Tanaka, Hersh, & Bernstam, 2007; Richardson & Smeaton, 1995) can provide information about the degree to which two terms are related. I therefore modified SAVCC further: to compute SAVCC, I create vectors with one element for each element in MeSH. Each element in these vectors is 0 or 1, 0 representing absence and 1, presence of the term in the indexer's output. I consider MeSH as a collection of independent trees, so terms that are classified under different categories do not match. Instead of a simplistic binary related/not related determination, I use the closeness between terms to create the relatedness matrix for the SAVCC computation (Figure 9).

Chapter 3

Graph-based ranking algorithms

Overall, our experiments with PageRank suggest that the structure of the Web graph is very useful for a variety of information retrieval tasks.

From *The PageRank Citation Ranking: Bringing Order to the Web* (Page et al., 1998)

Graph theory

Graphs are a convenient general model for any situation that consists of a set of entities, some of which are paired. Graph theory does not address what the entities or the pairings represent. A graph may model a computer network, a set of social relationships, the citation pattern of scientific publications, or any other situation that can be described as a set of entities, some of which are paired.

Formally, a graph G can be defined as a nonempty set of vertices V(G), a set of edges E(G) and an incidence function ψ_G that associates each edge in E(G) with an unordered pair of vertices in V(G). If e is an edge, x and y are vertices, and $\psi_G(e)=xy$, e connects x and y (Bondy & Murty, 1976). While (Bondy & Murty, 1976) use the term "vertex," the graph-based ranking algorithm literature prefers the term "node," which I will use from now on.

A convenient feature of graphs is that they can be easily represented through diagrams. In a typical graph diagram, a node is represented by a circle. An edge is represented by

a line joining the circles that correspond to the nodes identified by the incidence function for that edge.

For example,

$$
\begin{aligned}
G &= (V(G), E(G), \psi_G) \\
V(G) &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\} \\
E(G) &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\} \\
\psi_G &= \begin{cases}
\psi_G(e_1) = & v_1v_2 \\
\psi_G(e_2) = & v_2v_4 \\
\psi_G(e_3) = & v_4v_5 \\
\psi_G(e_4) = & v_3v_5 \\
\psi_G(e_5) = & v_5v_6 \\
\psi_G(e_6) = & v_5v_7 \\
\psi_G(e_7) = & v_6v_8 \\
\psi_G(e_8) = & v_8v_9 \\
\psi_G(e_9) = & v_7v_9 \\
\psi_G(e_{10}) = & v_9v_{10} \\
\psi_G(e_{11}) = & v_{10}v_{11}
\end{cases}
\end{aligned}
$$

fully describes a graph that corresponds to the diagram shown in Figure 11. The same graph may have several different correct diagrammatic representations.

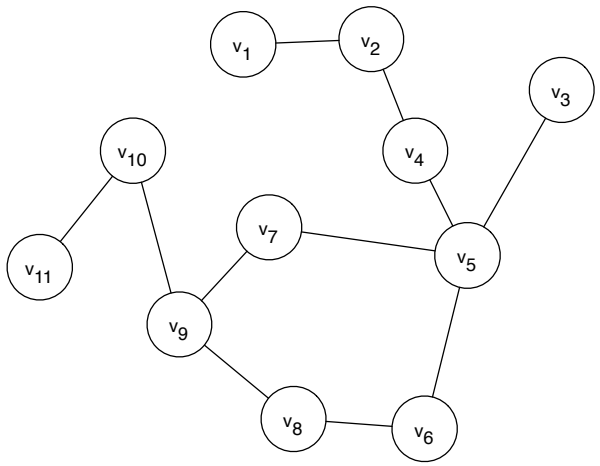

Figure 11 - Diagram illustrating graph G

The terminology and many concepts in graph theory come from the diagrammatic

representation. For an in-depth look at the basics of graph theory, please consult (Bondy & Murty, 1976).

Adjacency matrix

The adjacency matrix A of a graph G is $A(G) = [a_{ij}]$, where a_{ij} is the number of edges joining nodes v_i and v_j. For example, the adjacency matrix for the graph G in Figure 11 is:

$$A(G) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The adjacency matrix is the preferred format for storing graphs in many computer applications (Bondy & Murty, 1976).

Walks

A walk is an ordered sequence of alternating nodes and edges that starts with a node, ends with a node, and is not empty. If the walk contains no repeating edges, it is called a trail. If a trail contains no repeating nodes, it is called a path.

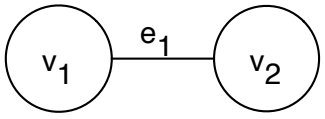

Figure 12 - Simple graph with two nodes and one edge joining them

If a walk begins and ends with the same node it is a closed walk. For example, a two-node graph with a single edge joining both nodes (Figure 12) contains infinite closed walks ($v_1e_1v_2e_1v_1$, $v_1e_1v_2e_1v_1e_1v_2e_1v_1$, etc.). A closed trail that contains at least one node different from the origin node is a cycle. Since all walks in the previous example contain edge e_1, the graph in Figure 12 contains no trails, and therefore cannot contain cycles. A graph is called acyclic if it does not contain cycles. (Bondy & Murty, 1976)

Directed and undirected graphs

A useful way of dividing graphs classifies them as directed or undirected. Undirected graphs are graphs in which the incidence function returns unordered pairs, i.e., the edges have no inherent direction. In other words, in such a graph $\psi_G(e_i) = v_j v_k = v_k v_j$. The adjacency matrix for an undirected graph is symmetrical.

A directed graph, in contrast, is one in which the incidence function returns an ordered pair. In directed graphs $\psi_G(e_i) = v_j v_k \neq v_k v_j$. Edges therefore have an inherent direction, usually represented with an arrow in a diagram (Bondy & Murty, 1976). The adjacency matrix for a directed graph is not guaranteed to be symmetrical and, in fact, most will not be symmetrical.

Directed graphs can be used to model some problems more accurately than undirected graphs. For example, citations in scientific papers are not commutative: if paper A cites paper B, it does not imply that paper B cites paper A. Directed graphs are useful to model asymmetric relationships.

Graph metrics

Several important graph metrics are commonly used to describe graphs, edges, and

nodes. The simplest node measure is the degree. The degree d_G of a node is the number of edges connected to it (Bondy & Murty, 1976), and is equivalent to the sum of the corresponding row in the adjacency matrix. In other words, if a_{ij} is an element of the adjacency matrix A(G) for a graph G, $d_G(v_i) = \sum_j a_{ij}$.

Distance and centrality

The distance matrix C of a graph G is a matrix describing the number of edges that must be traversed to reach one node from another, such that (if c_{ij} is an element of C), c_{ij} is the smallest number of edges that must be traversed from v_i to v_j. If there is no path between both nodes, c_{ij} contains a predefined constant K (usually the number of vertices in the graph). For example, for the graph in Figure 11,

$C =$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	*OD*	*ROC*
v_1	0	1	4	2	3	4	4	5	5	6	7	41	7.66
v_2	1	0	3	1	2	3	3	4	4	5	6	32	9.81
v_3	4	3	0	2	1	2	2	3	3	4	5	29	10.83
v_4	2	1	2	0	1	2	2	3	3	4	5	25	12.56
v_5	3	2	1	1	0	1	1	2	2	3	4	20	15.70
v_6	4	3	2	2	1	0	2	1	2	3	4	24	13.08
v_7	4	3	2	2	1	2	0	2	1	2	3	22	14.27
v_8	5	4	3	3	2	1	2	0	1	2	3	26	12.08
v_9	5	4	3	3	2	2	1	1	0	1	2	24	13.08
v_{10}	6	5	4	4	3	3	2	2	1	0	1	31	10.13
v_{11}	7	6	5	5	4	4	3	3	2	1	0	40	7.85
ID	41	32	29	25	20	24	22	26	24	31	40	314	
RIC	7.66	9.81	10.83	12.56	15.70	13.08	14.27	12.08	13.08	10.13	7.85		

Note that, for a directed graph, the distance matrix will not be symmetrical. There are four measures that are computed directly from the centrality matrix for each node: Out

Distance (OD), In Distance (ID), Relative Out Centrality (ROC) and Relative In Centrality (RIC). For undirected graphs, ID=OD and ROC=RIC. These measures are shown above, and are computed as follows:

$$OD_i = \sum_j c_{ij}$$

$$ID_i = \sum_j c_{ji}$$

$$ROC_i = \frac{\sum_i \sum_j c_{ij}}{\sum_j c_{ij}}$$

$$RIC_i = \frac{\sum_i \sum_j c_{ij}}{\sum_j c_{ji}}$$

The higher the RIC and ROC, the closer the node is to other nodes on the graph (Dhyani et al., 2002).

Compactness

The compactness of a graph measures its global connectedness, or cross-referencing. Compactness varies between 0 and 1. The higher the compactness, the easier is for nodes to reach each other by walking along the edges of the graph. A graph with a compactness of 0 is completely disconnected, while a completely connected graph (like the one in Figure 11) has a compactness of 1.

The following definitions are necessary to compute compactness:

Max is the highest possible value for $\sum_i \sum_j c_{ij}$ for a given distance matrix C. If N is the size of C, and K is the constant used to represent no path between vertices in the graph, Max=$(N^2-N)K$. Conversely, Min is the lowest possible value for a distance matrix of size

N and a certain K, and is $\text{Min}=N^2-N$.

The compactness C_p is then defined as $C_p = \frac{\text{Max} - \sum_i \sum_j c_{ij}}{\text{Max} - \text{Min}}$ and, as mentioned, is 0 for a disconnected graph and 1 for a fully connected graph (Dhyani et al., 2002).

Graph-based ranking algorithms

Graph-based ranking algorithms rank nodes according to their relative importance. In a basic graph-based ranking model, edges in a directed graph connecting two nodes represent links and are considered "votes," and these votes are weighted by the reputation of the voter. The number of votes each node receives and emits determines its reputation. These algorithms are therefore computed iteratively until the computation converges (Hersh, 2003; Jorge R. Herskovic & Bernstam, 2005; Kleinberg, 1999; Page et al., 1998).

HITS

HITS, developed by Jon Kleinberg, was one of the first published algorithms to exploit the link structure of graphs on the World Wide Web (WWW). It models hyperlinked environments such as the WWW as collections of authorities and hubs. Authorities are documents that provide authoritative information on a particular subject or to answer a particular query. Authorities can be identified by a large number of incoming links. For example, http://www.shis.uth.tmc.edu is an authority on the WWW on The University of Texas School of Health Information Sciences at Houston, and will have a large number of links pointing to it. Kleinberg argues that the problem with this approach is that many documents in hyperlinked environments have both large numbers of

incoming links and mention relevant terms repeatedly, but omit other relevant terms (like, in this example, "biomedical informatics"). Therefore, ranking pages solely by in-degree and retrieving them purely by relevance is suboptimal (Kleinberg, 1999).

Kleinberg's answer to this problem was to exploit human judgment information encoded in links. He proceeds to define "hubs" as pages that have large numbers of outgoing links to authorities. For example, a web page on "biomedical informatics graduate degrees" might list Harvard Medical Informatics, Stanford Biomedical Informatics, and The University of Texas School of Health Information Sciences at Houston. Thus, according to Kleinberg, hubs are documents that link to large numbers of authorities, and authorities are documents that have large numbers of incoming links from hubs.

Applying the HITS computation to a graph results in two scores for each node. One is the authority score, and the second is the hub score. The nodes of a graph with the highest authority score will be its authorities and, analogously, the highest hub scores will belong to the graph's hubs. For details on the computation of HITS, please see (Kleinberg, 1999).

PageRank

Perhaps the best-known example of graph-based ranking is Google (http://www.google.com), an Internet search engine company. Google indexes the WWW and provides search results to user-entered queries. Google internally models the WWW as a directed graph like HITS, but does not assume that web pages have any particular role. PageRank instead ranks nodes in a graph by their relative importance, as determined by the information encoded in the graph's edges.

PageRank computes the formula in Figure 13 over the entire graph iteratively. The PageRank value for a web page converges on the probability that a person clicking hyperlinks at random will end up on that page. This is called the random surfer model (Page et al., 1998). PageRank has been successfully applied to graph models of other networks beside the WWW, including the citation network of biomedical literature (Bernstam et al., 2006), analysis of social networks (Pujol et al., 2002), and text summarization through selection of important sentences (Mihalcea, 2004). PageRank can be considered a general algorithm that will rank nodes in a graph based on their relative importance as established by the set of edges.

$$R'(u) = c \sum_{v \in Bu} \frac{R(v)}{N_v} + cE(u)$$

Figure 13 - The PageRank formula

R is the PageRank, R' is the new PageRank, N is the number of outgoing links, v is the recommender and u is the recommendee. B is the set of incoming links. c is a decay factor, and E is a baseline PageRank for "rank sinks" like closed loops.

TextRank

TextRank is a PageRank derivative created by Rada Mihalcea to work on undirected graphs. TextRank exploits the network of related sentences in a piece of text for summarization. Mihalcea's work relied on determining relations between sentences based on word co-occurrence. Word co-occurrence has no inherent directionality but may have a weight, unlike hyperlinks on the WWW. Although algorithms designed for directed graphs can be used on undirected graphs by considering every link bidirectional, PageRank does not account for different link weights. Mihalcea therefore created a new ranking algorithm based on PageRank that took into account the lack of directionality and could use the information in weighted

$$T(V_i) = (1-d) + d \sum_{v_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} T(V_j)$$

Figure 14 - The TextRank formula. d is a decay factor, V is the set of vertices, w is an element in the weight matrix, and T is the TextRank score for a node

links (Mihalcea, 2004; Mihalcea & Tarau, 2004). TextRank can perform text summarization (Mihalcea, 2004; Mihalcea & Tarau, 2004) and keyword extraction (Mihalcea & Tarau, 2004) successfully.

TextRank is defined by the formula in Figure 14.

Relevant previous work

Semantic graphs

Semantic or lexical graphs represent relationships between words in a piece of text or a corpus. Semantic graphs can discover relationships between terms by clustering. These graphs are built by computing various metrics for word association, such as co-occurrence. Associated terms are represented as linked nodes in the graph. Frequency cutoffs, among other techniques, keep the number of nodes from increasing exponentially. Applications include word sense disambiguation (detecting which of several possible meanings of a word a text is using in a particular instance) and automated lexical acquisition (unsupervised discovery of vocabularies) (Widdows & Dorow, 2002).

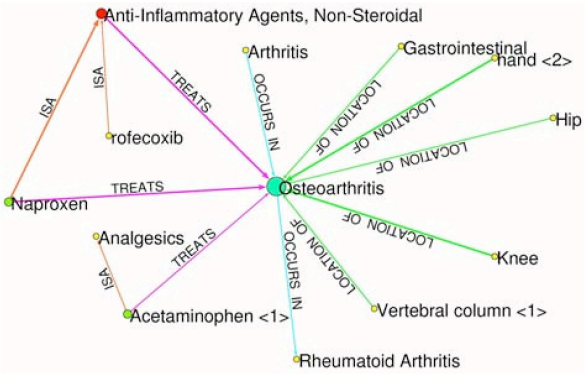

Figure 15 - Example of a semantic abstraction graph from Fiszman et al., 2004

Semantic abstraction graphs

Semantic abstraction graphs are semantic graphs that represent relationships between concepts instead of words. Computerized dictionaries or ontologies can deduce generic concepts from the actual

terms on a piece of text, and this enables the user to study the relationships between those concepts.

A team at the NLM uses SemRep to create graphical representations that summarize an article by drawing semantic abstraction graphs. These semantic abstraction graphs show concepts related to a user-selected topic (Osteoarthritis, in Figure 15) (Fiszman et al., 2004). However, since they require the user to select the core concept explicitly, they are unsuitable for fully unsupervised summarization.

MEDRank is inspired, in part, by this work by Fiszman and collaborators at the U. S. National Library of Medicine (NLM). Fiszman's work could be applied to generate topical summaries of large MEDLINE result sets, but it is not directly applicable to automatic processing of articles. I extended Fiszman's original work to leverage all concepts in an article, and use unsupervised graph-based ranking algorithms to determine the most important concepts in the article.

Graph-based ranking in semantic graphs

MEDRank is also based in part on Mihalcea's work using graph-based ranking algorithms to generate automated summaries. Mihalcea's TextRank algorithm computed networks of sentences in an article by using the repetition of terms. If the same term appears in two sentences, the two sentences must be somewhat related; if they have more terms in common, they are more related. She then represented these networks as undirected graphs, with the sentences being nodes and the relationships between sentences as edges. Each edge's weight was set to the strength of the relationship between the pair of sentences it joined. Mihalcea applied different graph-based ranking algorithms to rank these nodes and chose the highest-ranking ones, using

the sentences they represented to build document summaries (Mihalcea, 2004). Mihalcea also applied TextRank to keyword extraction in abstracts, and found that it performed better than other unsupervised keyword extraction methods, but not as well as supervised methods (Mihalcea & Tarau, 2004). Mihalcea's work was done on semantic graphs (using words from the text directly), and she did not study the application of TextRank to semantic abstraction graphs. MEDRank extends Mihalcea's work by indexing full text and by using semantic abstraction graphs to sidestep the problem of word sense disambiguation, which is prevalent in the biomedical literature (Savova et al., 2008; Schuemie, Kors, & Mons, 2005).

Theoretical basis for using semantic abstraction graphs

Suppe argued that concepts found in scientific papers and the relationships between them are structured in such a way as to support a set of claims (Suppe, 1998). These claims, being the most important part of the paper's argument, will be about the most important concepts in the paper. Each concept in the paper can be represented as a node in a semantic abstraction graph, and each relationship between concepts as an edge in the same graph. Therefore, finding the most important nodes in this graph will produce the most important concepts in the paper.

Graph-based ranking using incomplete graphs

The use of graph-based ranking algorithms to determine summaries, keywords, and search results assumes that the graphs are adequate for the task. However, graph construction may depend on Natural Language Processing (NLP) software, database mappings, or uncertain data. In previous work, I explored how much gradually removing links from a graph of article citations impacted PageRank's ability to detect

the most important nodes in a graph. I discovered that removing 99% of the original links was necessary before the ranking produced by PageRank changed substantially. Graphs with suboptimal link structures are still useful for determining relative importance (Jorge R. Herskovic & Bernstam, 2005).

Chapter 4

The MEDRank system

Determining the most important concepts in an article

Mihalcea's work suggests that an appropriate way to obtain the most important concepts in a piece of text is to apply graph-based ranking algorithms to a graph representation of its contents. To obtain the most important concepts in a full-text biomedical article MEDRank:

1. Obtains all of the detectable concepts in an article
2. Determines the relationships between concepts
3. Builds a Semantic Abstraction Graph with these concepts and relationships
4. Applies a graph-based ranking algorithm to the graph

MEDRank's processing is similar to MTI's pipeline, with the addition of the graph creation and ranking steps (Figure 16).

Obtaining all detectable concepts in an article

The NLM produces Natural Language Processing (NLP) software that takes text as input and returns the UMLS concepts that it finds, along with their positions in the text. The NLM's software is developed as part of the Indexing Initiative that also created MTI and, in fact, is part of MTI itself.

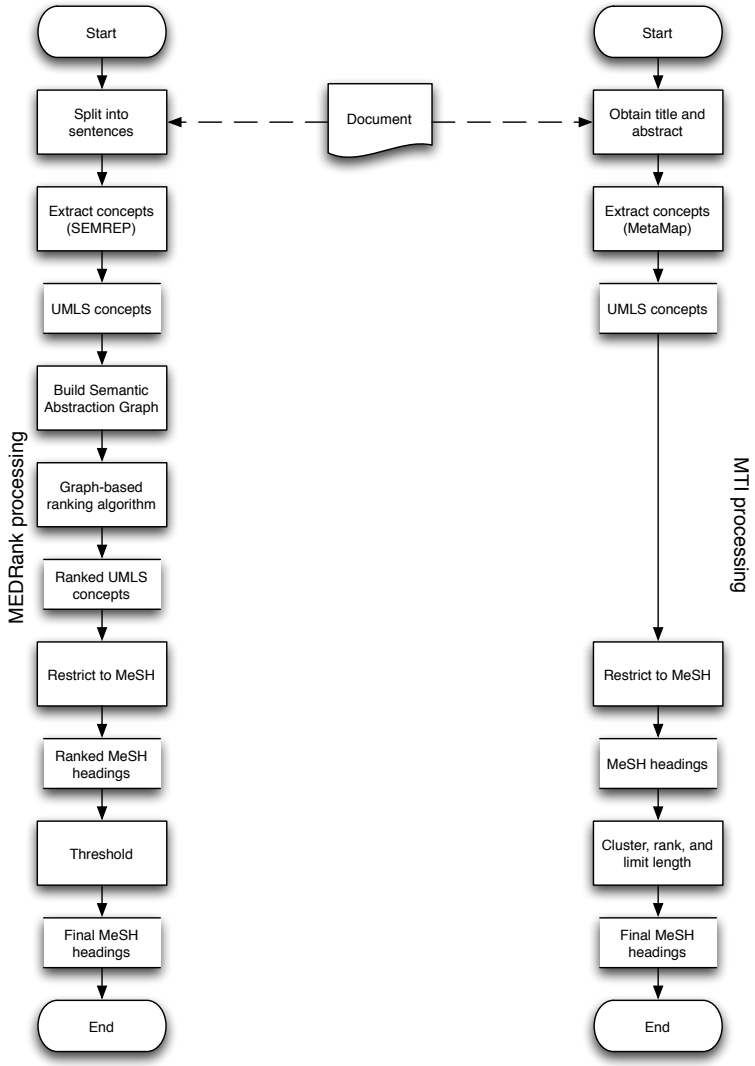

Figure 16 - MEDRank and MTI processing

MEDRank can currently use two different NLM NLP products (available at http://skr.nlm.nih.gov): MetaMap (described in Chapter 2) and SEMREP. SEMREP leverages semantic knowledge from the UMLS to improve the accuracy of concept and relationship detection (Rindflesch, Bean, & Sneiderman, 2000). I used SEMREP because

it produced better results on the training sample than MetaMap.

MEDRank uses a simple model in which adjacent concepts in the same sentence are considered related. Before submitting text to the NLM NLP servers, MEDRank divides the text into single sentences for individual processing. Unfortunately, biomedical text is difficult to split. The traditional sentence separators is the period (“.”). I therefore chose a naïve algorithm that split sentences at every period, but performance was unacceptable. The period is legitimately embedded in abbreviations (“Fig.”, “vs.”) and species names (“E. coli”). I therefore developed a sentence splitting process that can split most sentences in the training sample well. It splits the text using a regular expression. The sentence separator ignores periods that are not followed by a space and an uppercase letter, ignores periods that are not followed by a newline, and ignores periods that come after the strings “eg”, “e.g”, “eq”, “fig”, “vs”, “exp”, “al”, or “r.m.s”, or before the strings “coli” and “typhimurium” (the two most common bacteria names in the training sample).

Removing the influence of the NLP software

The quality of the NLP software is a critical factor in the performance of systems like MEDRank. Other indexing systems like MTI make heavy use of historical data and heuristics, and can thus outperform their underlying NLP by correcting for known deficiencies. MEDRank, in contrast, does not incorporate heuristics that can compensate for poor NLP output. Since this problem is beyond this thesis’ claims and scope, I operationalize the quality of the NLP for each article by mapping all detected concepts to MeSH and I obtain the recall (see Precision and recall above) of this set of terms, which I call the total recall. Although only articles with a total recall of 1.0 give

MEDRank all of the data it needs to potentially capture all MEDLINE (with MTI) terms, these articles are scarce. Only four articles (0.77%) in the training sample have a total recall of 1.0. I therefore chose a lower total recall threshold, 0.85, to obtain a higher number of articles (11, 2.1%). Only these articles with a total recall of 0.85 or more (high-quality NLP) are used to compute information retrieval measures.

Noise

The output of the NLP software can be noisy. Like many other NLP systems, the NLM software finds many high frequency concepts that contain little information. For example, the three most frequent UMLS concepts that SEMREP finds in the training sample are (in order) Negation, One, and Two. Some indexing systems use stop words to eliminate low-content concepts such as these. MEDRank takes a more general approach. It performs a first pass through the entire sample counting all concepts. It then uses these frequency counts to weight concepts when processing each article using the TF*IDF (Salton, 1963) formula as described in (Hersh et al., 1994) and shown in Figure 17.

$$TF(i,j) = \log(\text{frequency of i in j}) + 1$$

$$IDF(i) = \log\left(\frac{\text{number of documents in corpus}}{\text{number of documents that contain i}}\right) + 1$$

$$WEIGHT(i,j) = TF(i,j) * IDF(i)$$

Figure 17 - TF*IDF weight computation for a term i in a document j.

Obtaining all relationships between concepts

The UMLS contains a set of relationships that the NLM NLP software can detect. Unfortunately, this set is incomplete, even for highly technical domains like genetics. The UMLS can describe between 60% and 83% of the genetics relationships found in other ontologies (O. Bodenreider, Mitchell, & McCray, 2002).

SEMREP returns UMLS relationships between concepts as part of its output. I explored using these to generate SAGs. Unfortunately, SEMREP is very conservative and produces few relationships. The graphs produced using SEMREP's relationships were sparse, and were not able to achieve meaningful results. In practical terms, SEMREP fell short of the possible relationships between concepts in a scientific article.

I therefore turned to co-occurrence within sentences to obtain more relationships between concepts. Concepts that co-occur in a sentence are likely to be related (Matsuo & Ishizuka, 2004) and, more specifically, concepts that co-occur within a certain sliding window are likely to be related (Mihalcea & Tarau, 2004; Pedersen, 2000). In the English language, concepts that occur close to one another in the same sentence are more likely to be related, and the closer they are to each other, the stronger their relationship (Eisner & Smith, 2005; Gamon, 2006). Word co-occurrence has been successfully used with TextRank to extract keywords from abstracts (Mihalcea & Tarau, 2004) so, by analogy, concept co-occurrence is likely to generate useful relationships.

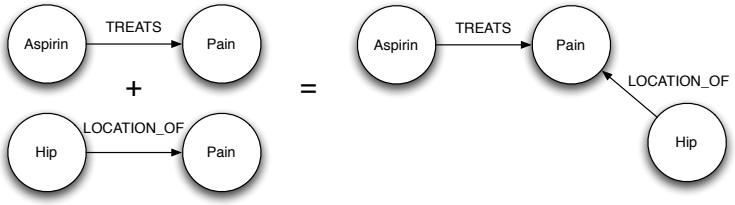

Figure 18 - Adding a new relationship to a graph where one of the nodes already exists (using UMLS relationships)

I determined the optimal size of the sliding window by running several experiments with increasing window sizes (from one to six), and comparing the results.

Building a Semantic Abstraction Graph

Fiszman describes a simple algorithm for building SAGs based on the output from SEMREP. Fiszman's algorithm requires a seed concept, and is focused on SAGs for

disorders (Fiszman et al., 2004). I simplified Fiszman's algorithm to build SAGs in an unsupervised way.

Since MEDRank does not use directional relationships, it creates weighted undirected SAGs. To build SAGs, I take the list of relationships produced by the previous step and iterate through them in order, adding each one to the SAG. Concepts are treated as nodes, and the relationship itself is treated as the edge connecting the nodes.

SEMREP returns an estimate of the quality of its mapping (called "confidence" and ranging 0-1000) for each concept. If a node representing a specific concept is already present in the SAG, the confidence of each instance of each specific concept is recorded but no new nodes are added to the graph. Instead, a new edge from the same node is added. In other words, there will only be a single node for each unique concept in the graph (Figure 18). Starting node weights are the average confidence divided by 1,000. Each edge's weight is the average confidence of the nodes it connects.

Graph-based ranking algorithms

MEDRank can apply one of three different graph-based ranking algorithms to SAGs generated by the previous steps:

$$T(V_i) = (1-d) + d \sum_{v_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} T(V_j)$$

Figure 19 - The TextRank formula. d is a decay factor, V is the set of vertices, w is an element in the weight matrix, and T is the TextRank score for a node

HITS (Kleinberg, 1999), PageRank (Page et al., 1998), or TextRank (Mihalcea, 2004). The implementation of each algorithm is based on its original published description. Since HITS produces two different scores (the "hubs" and "authorities" scores), MEDRank's HITS implementation accepts a function that can combine them to produce a single score. MEDRank applies the selected ranking algorithm iteratively, until the total

difference in scores between two consecutive iterations is less than 0.0001. The computations converge in 20 to 30 iterations.

TextRank performed better than PageRank and HITS on preliminary evaluations on the training data. HITS also had a large disadvantage: creating an adequate function to combine hubs and authorities was impossible, and MEDRank performed very poorly using HITS. I thus performed all experiments using TextRank. I used 0.85 for the value of the decay factor d (Figure 19), as suggested in the literature (Mihalcea & Tarau, 2004; Page et al., 1998).

Further processing

After ranking, MEDRank holds a list of nodes ranked by score. To obtain a final list of MeSH terms for an article, the list must be cut off at a threshold, and the UMLS concepts must be mapped to MeSH terms. MEDRank normalizes the scores to values between 0 and 1 for every article. In other words, the top-ranked concept of every article has a score of 1.

MEDRank uses the same Restrict to MeSH algorithm (U.S. National Library of Medicine, 2004c) that MTI uses. Restrict to MeSH takes UMLS concepts as input, and returns candidate MeSH terms. MTI post processes the candidate MeSH terms during its ranking step. It disambiguates between mappings according to several factors including, among others, its position in the MeSH hierarchy, and known co-occurrences with other MeSH terms (U.S. National Library of Medicine, 2004a). MEDRank simplifies MTI's post processing into a single heuristic: if there is more than one potential mapping available for a UMLS concept, MEDRank chooses the one that is deeper in the MeSH hierarchy (the most specific one). After mapping all UMLS concepts above the

threshold to MeSH, duplicate MeSH terms are removed.

The length of MEDRank's output list is limited to a maximum of 25 terms, regardless of the score of the terms. In other words, if 30 terms have a ranking score greater than the threshold, only the first 25 will be used.

Implementation

I built MEDRank in Python (http://www.python.org), an interpreted scripting language. I used Python 2.6, the latest version available at the time of this writing. Computationally intensive components (notably, matrix distance calculations) were implemented in C++ for speed. The adjacency and distance matrices that MEDRank uses are large and relatively dense, and have known access patterns. MEDRank therefore has its own matrix and vector classes that implement just the functionality that the software requires. MEDRank uses the BioPython library (http://www.biopython.org) 1.49 to access PubMed records and obtain MEDLINE (with MTI) data.

I ran all software on an Apple Macintosh computer running Mac OS X 10.5.5 (**Apple Inc.**, Cupertino, CA). I preprocessed data using Microsoft Excel 2008 for Mac 12.1.4 (Microsoft Inc., Redmond, WA) and processed it, computed statistics, and built plots using R 2.8.0 (http://www.r-project.org).

To test MEDRank during development I used a small sample of full-text articles from PubMed Central that I called a "training sample." PubMed Central is a free repository of scientific articles maintained by the NLM and available at http://www.pubmedcentral.nih.gov/. I chose 544 articles at random (using a computer program) from PubMed Central's entire catalog. To create the sample, I first

downloaded the PubMed Central catalog from ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/file_list.txt. In the catalog, each line represents an article. I then selected articles randomly from the catalog. I downloaded the chosen articles, and excluded articles if they had no XML representation, or did not have a labeled title and abstract. I also excluded articles with no PubMed ID, and articles that were in PubMed but did not have indexing terms yet. I also downloaded the corresponding MEDLINE records. I submitted the abstracts and titles to MTI for processing, and eliminated articles that MTI could not process. This left a training sample of 521 articles that I used to evaluate the MEDRank system while building it.

I performed all work for this thesis using the 2008 editions of MeSH and the UMLS.

Threshold determination

I evaluated all articles in the Training sample with a total recall of 0.85 or more against the MEDLINE (with MTI) gold standard to determine an optimal TextRank threshold to cut off the generated lists. I evaluated several TextRank thresholds by computing a mean SAVCC score (see Semantically Aware Vector Cosine Comparison (SAVCC) below) and a mean length of the output list and comparing them to the MEDLINE (with MTI) gold standard. I looked for the threshold that returned the highest mean SAVCC score, did not have low outliers (single articles that performed very poorly), and also had an output length comparable to the MEDLINE indexers (i.e. 12 to 15 terms, to account for checktags). I selected 0.20 as the best combination of these features.

Test sample

I built a test sample in exactly the same way as the Training sample. I requested 4,999 full text articles at random from PubMed Central and excluded articles without tagged

titles or abstracts. I also excluded articles with no PubMed ID, and articles that were in PubMed but did not yet have indexing terms. I excluded articles from the training sample in the test sample. The final test sample consisted of 4,690 articles. I downloaded it and did not inspect or use it until I was satisfied with the results of running MEDRank on the training sample.

Evaluation

I evaluated MEDRank using five different metrics: Precision, Recall, F_2 (a weighted harmonic mean of recall and precision that favors recall and is used in most NLM papers evaluating MTI), Hooper's Indexing Consistency, and a Semantically Aware Vector Cosine Comparison. I conducted all evaluations on articles from the test sample for which the total recall of all possible MeSH terms was 0.85 or more.

Analysis

I evaluated MEDRank's performance by comparing it to the entire output of SEMREP (no ranking) and three types of ranking, from ineffective (ranking concepts in alphabetical order) to frequency ranking and TF*IDF.

Alphabetical ranking does not have a score that can be used as a threshold to limit the list length. For alphabetical ranking, I assumed that the optimal term list length is the one chosen by the MEDLINE indexers and present in the MEDLINE (with MTI) gold standard. I thus limited the list length for MeSH terms ranked alphabetically to the same length as the gold standard.

To avoid performing a slow and laborious manual search, I obtained thresholds for the frequency-ranked and TF*IDF-ranked data computationally. I computed mean SAVCC

and output length for 100 different thresholds between 0 and 1 using TF*IDF ranking on the training sample. I repeated the procedure for frequency ranking (using cutoffs from 0 to 100 occurrences). I plotted the results and inspected the graphs to choose the threshold with the highest mean SAVCC and output length closest to 12. I expected MEDRank to perform significantly better than these other ranking strategies (None, Alphabetical, and frequency ranking, and TF*IDF). I obtained the top two ranking strategies by mean SAVCC score. I compared these two top strategies using a paired, two-tailed Student's T test looking for a positive difference (α=0.05).

Comparison to MTI

I submitted the titles and abstracts of articles in the test sample to MTI through its WWW interface at http://skr.nlm.nih.gov. I set MTI to its default settings, which are the ones used to preprocess citations for MEDLINE, with one exception. I set the weight of the PubMed Related Citations (REL) path (see Chapter 2) to 0, since REL leverages human indexer input directly by pulling terms from related articles. I kept MTI's default setting of returning 25 terms per article to emulate MTI's original task of providing terms for the indexers.

I compared MEDRank's performance against MTI using articles from the test sample with a total recall of 0.85 or more. I compared all terms generated by MTI by computing precision, recall, Hooper's IC and SAVCC measures using MEDLINE (with MTI) as a gold standard. I expected MEDRank to perform significantly better than MTI.

I performed all comparisons only on the SAVCC variable using paired, two-tailed T tests with α=0.05.

Chapter 5

MEDRank evaluation

This chapter describes the results of my evaluation of MEDRank, as described in Chapter 4.

Training sample

The training sample was 544 articles chosen and retrieved at random from PubMed central. The 544 articles were split into 76,941 individual sentences and processed by SEMREP. Titles and abstracts were extracted from their corresponding MEDLINE records and processed by MTI. Of these 544 articles, 23 had no titles or abstracts, or caused MTI processing errors and were eliminated. The final training sample therefore contained 521 articles.

Quality of the NLP software

I compared all MeSH terms obtained by converting SEMREP output into MeSH to the terms in the MEDLINE (with MTI) gold standard. The mean recall was 0.523 (range 0-1, SD=0.157) (Figure 20).

Of the articles in the training sample, 11 had a total recall of 0.85 or more. Those 11 articles were used to compute the thresholds below.

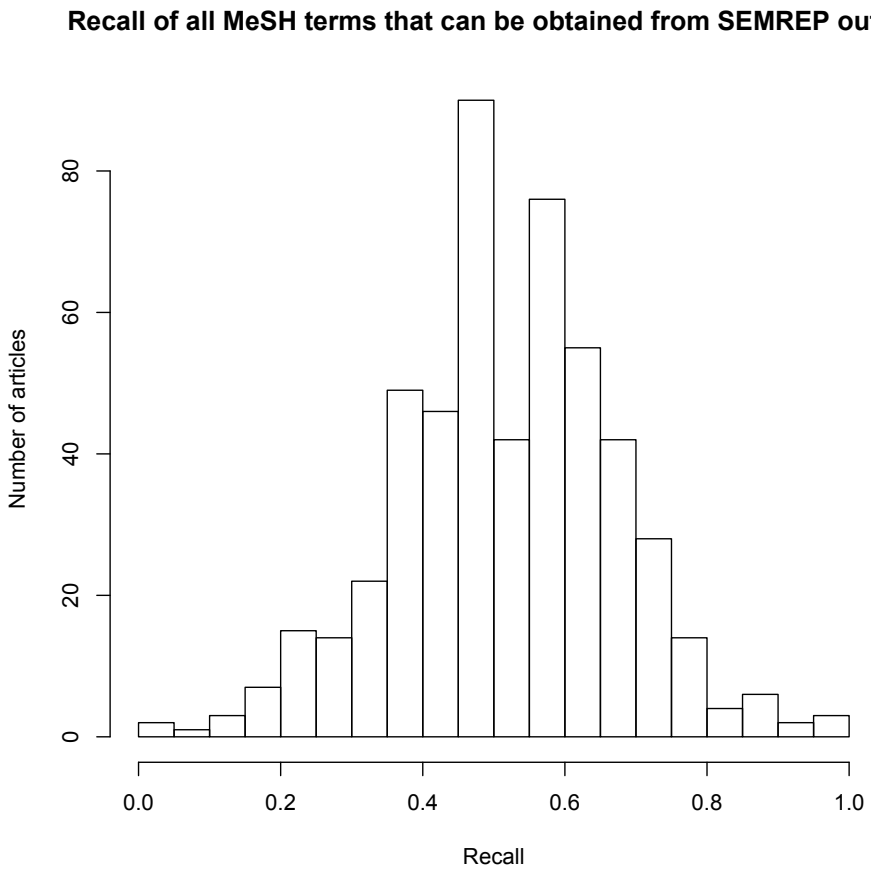

Figure 20 - Recall for all generated MeSH terms for articles in the training sample

Generated graphs

The Semantic Abstraction Graphs (SAG) built from the 521 articles in the training sample had a mean of 350.7 nodes (range: 105- 906, SD=104.4) (Figure 21).

The SAG had a median of 723.8 edges (range: 157- 2,156, Figure 22). The graphs were also highly connected: mean compactness was 0.966 (range: 0.623-0.997, SD=0.036).

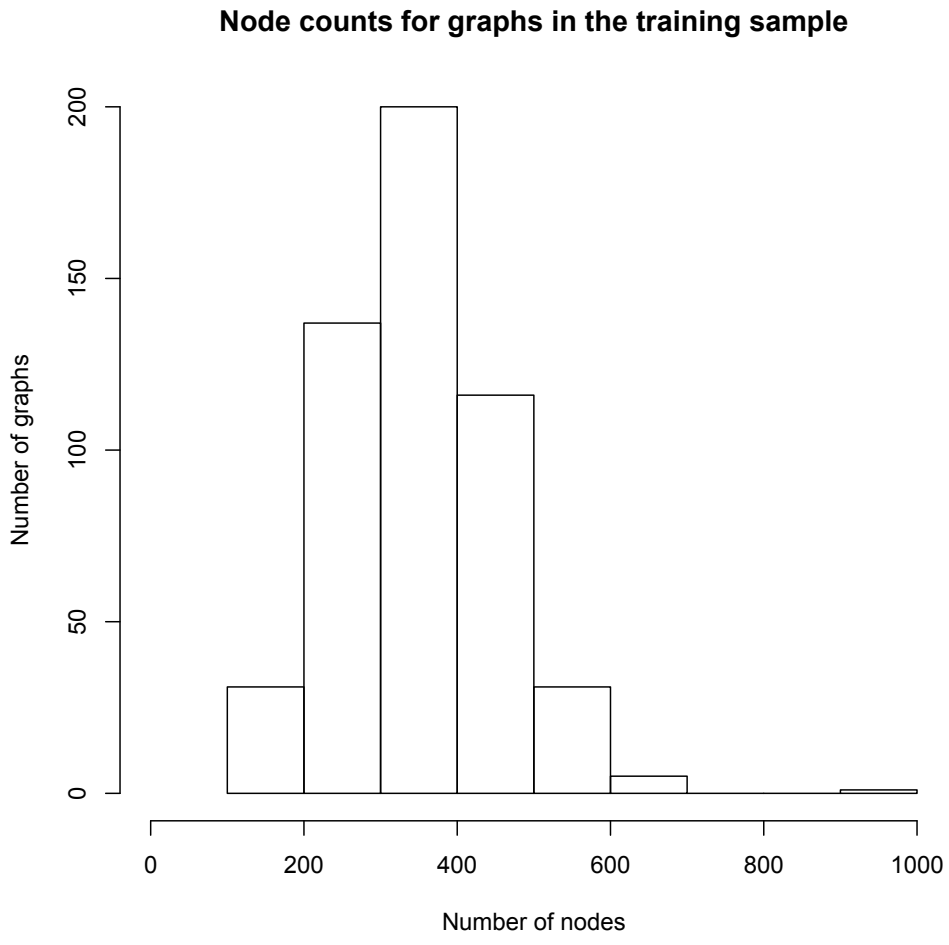

Figure 21 - Histogram of node counts for graphs in the training sample

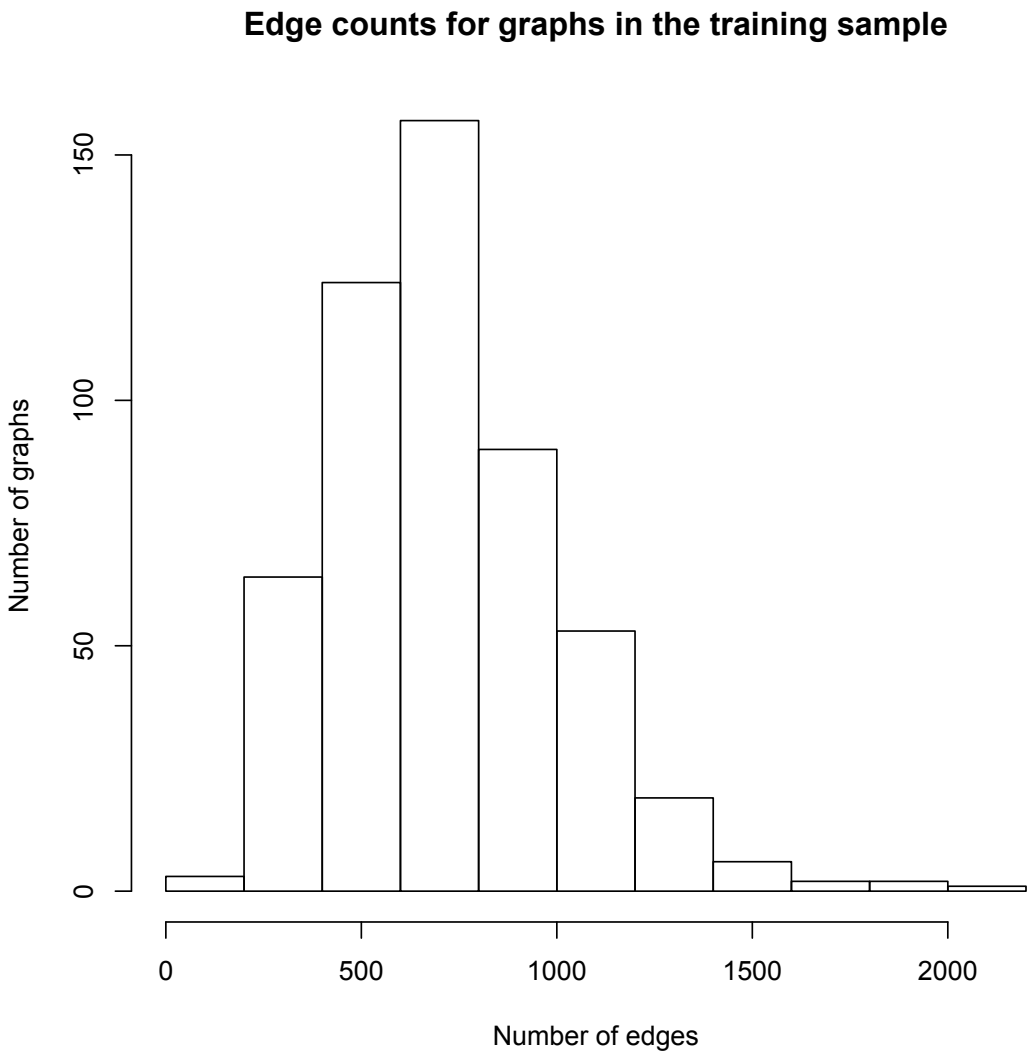

Figure 22 - Distribution of edge counts for graphs in the training sample

Concept co-occurrence window size

Varying sliding window size between one and six did not affect MEDRank performance on the training data (Figure 23). Since Mihalcea found that a window of two words was optimal when using TextRank (Mihalcea & Tarau, 2004), and a single UMLS concept can describe more than two words, I performed the rest of the experiments using a sliding window size of one concept. In other words, I only built relationships using concepts that were adjacent.

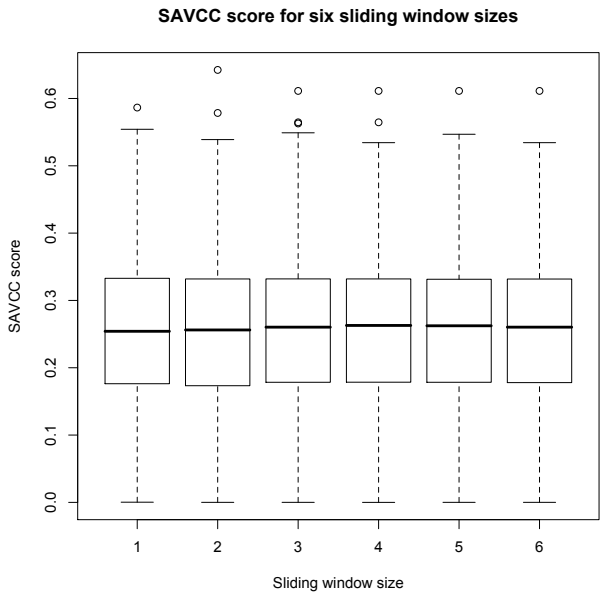

Figure 23 - Box plots of the SAVCC scores for six different window sizes over the entire training sample

TextRank scores

All TextRank scores MEDRank produces for each article are normalized to a maximum of 1.0. TextRank scores for all terms generated from the 521 articles in the training sample had a median of 0.040 (range: 0.004 to 1.0000). The lower scores are more prevalent. A histogram of the probability densities of the logarithm of these TextRank scores is shown in Figure 24.

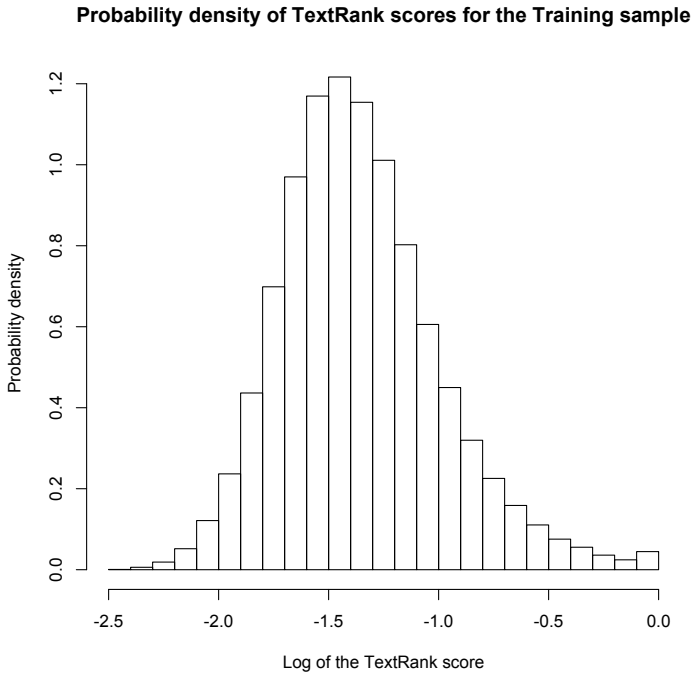

Figure 24 - Probability density for the logarithm of a TextRank score for each node in the training sample

Threshold determination

I determined the optimal threshold to stop processing MEDRank output using frequency ranking and TF*IDF by processing the training sample 100 times with different thresholds. I measured the mean output length (Figure 25) and SAVCC score (Figure 26) for each of these thresholds.

The optimal threshold for TF*IDF ranking was 0.60, and the optimal threshold for frequency ranking was 9 (see Figure 27).

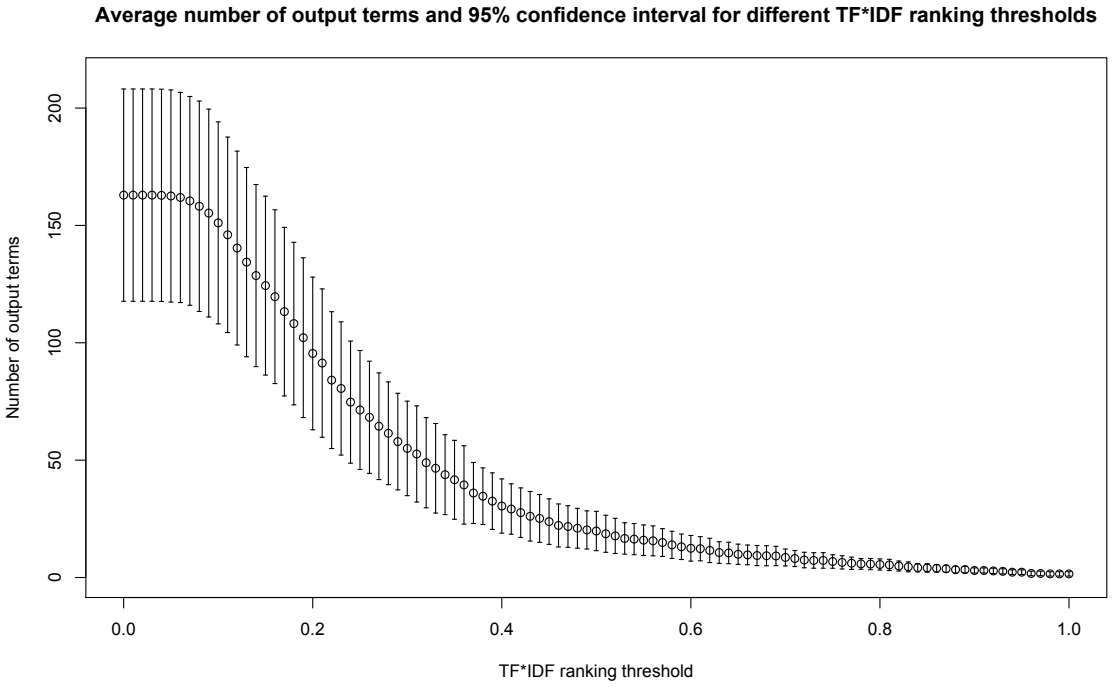

Figure 25 - Average output length and 95% confidence intervals for different TextRank ranking thresholds on the training data.

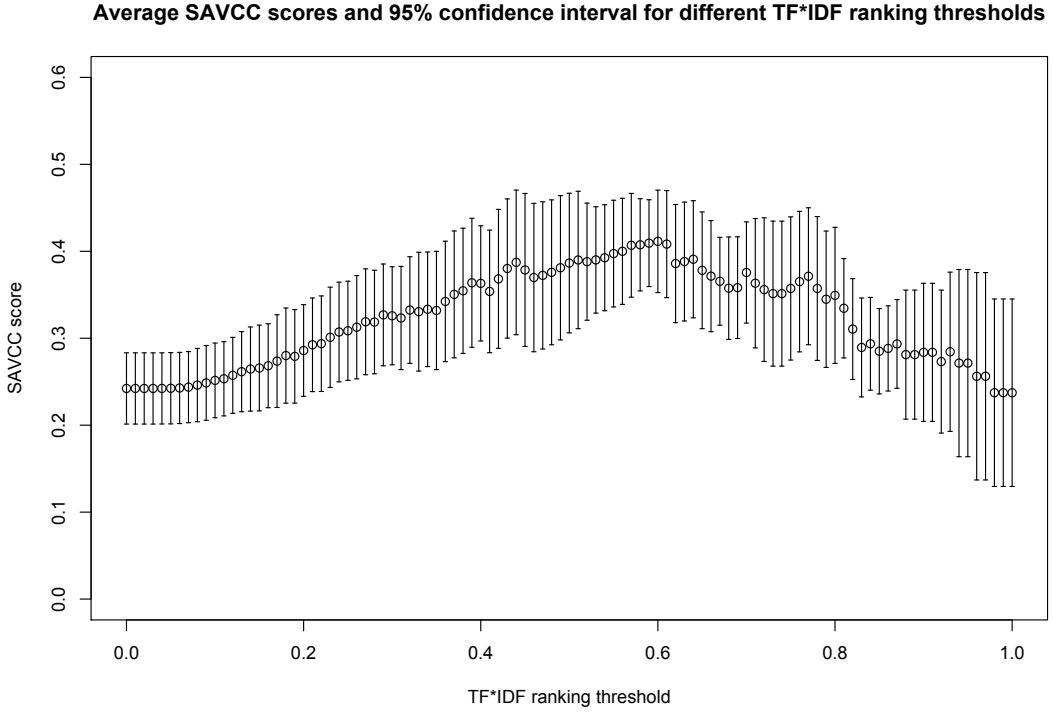

Figure 26 - Average SAVCC scores and 95% confidence intervals for different TF*IDF ranking thresholds on the training data

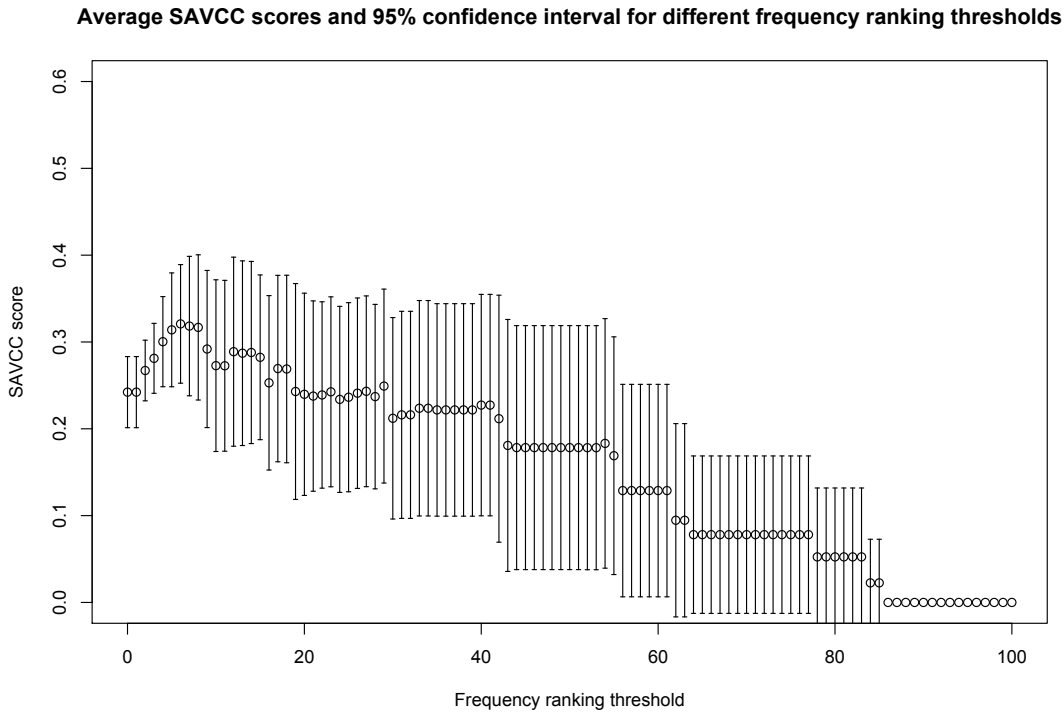

Figure 27 - Average SAVCC scores and 95% confidence intervals for different frequency ranking thresholds on the training sample

Test sample

To create the test sample, I downloaded 4,999 full text articles from PubMed Central at random. Of the 4,999 downloaded articles, 308 (6.2%) had no PubMed ID, abstract, or title, or their titles and abstracts caused errors when processed by MTI and were excluded. The Test sample therefore had 4,691 articles. Of these 4,691 articles, 88 (1.9%) had a total recall of 0.85 or more. I used those 88 articles for the rest of the evaluation.

MEDRank evaluation

I compared MEDRank's output on the Test sample to the MEDLINE (with MTI) gold standard.

Information retrieval measures

MEDRank achieved a 0.391 mean precision (SD= 0.204) and 0.351 mean recall (SD=0.155) on the 88 high-quality NLP articles in the test sample. Its mean F_2 measure was 0.339 (SD= 0.136). Hooper's mean indexing consistency was 0.212 (SD= 0.103). Mean SAVCC was 0.359 (SD= 0.136).

Comparison to other ranking strategies

I evaluated no ranking, alphabetical ranking, frequency ranking, TF*IDF, and MTI to compare against MEDRank. The results for the 88 articles with high quality in the test set are presented in Table 1. Bold type denotes the highest score for each measure.

Ranking	Precision (mean ± SD)	Recall (mean ± SD)	F_2 (mean ± SD)	Hooper's (mean ± SD)	SAVCC (mean ± SD)
None	0.056±0.023	0.905±0.053	0.219±0.071	0.056±0.023	0.223±0.046
Alphabetical	0.066±0.080	0.066±0.080	0.066±0.080	0.036±0.045	0.066±0.080
Frequency	0.244±0.156	0.324±0.157	0.285±0.125	0.153±0.082	0.274±0.123
TF*IDF	0.177±0.086	0.448±0.159	0.331±0.118	0.144±0.067	0.278±0.103
MTI	0.207±0.088	**0.525±0.164**	**0.388±0.119**	0.173±0.071	0.324±0.101
MEDRank	**0.391±0.204**	0.351±0.155	0.339±0.136	**0.212±0.103**	**0.359±0.136**

Table 1 - Information retrieval measures for different ranking strategies.

The top two ranking strategies were, as expected, MTI and MEDRank. I compared both samples with a two-tailed Student's T test. The test showed that the mean SAVCC for MEDRank was significantly higher than the mean for MTI ($p<0.05$). MEDRank's mean SAVCC was also significantly higher than TF*IDF's ($p<0.001$).

Chapter 6

Discussion

Discussion of the experimental results

MEDRank outperforms no ranking, ineffective (alphabetical) alphabetical ranking, and simple algorithms like frequency ranking and TF*IDF. Its mean recall (0.351) is lower than MTI's (0.525), which is also expected. MTI's output length is approximately twice as long as MEDRank's. If both were equally accurate, MTI's recall could potentially be twice as high as MEDRank's.

MTI was built to provide indexing suggestions to indexers, and its goal is high recall rather than precision. The NLM uses F_2, a measure biased for recall, precisely because it is MTI's goal. MTI's high recall and F_2 scores and comparatively low precision are thus artifacts of its development history. MEDRank has lower recall than MTI but higher precision, a reasonable tradeoff for a general indexing system. When performance is measured using general measures (Hooper's IC and SAVCC) MEDRank significantly outperforms MTI.

Further, MEDRank's performance as measured by mean F_2 (0.339) is superior to MTI's performance using titles and abstracts in a comparable study in 2005 (mean F_2=0.324) (Gay et al., 2005). MTI improved since 2005, achieving a mean F_2 of 0.388 in my comparable experiments. Considering the time and effort expended on MTI, a product of years of work by a team of dedicated computer scientists at the NLM, I believe that MEDRank is capable of surpassing it given comparable effort and resources.

Original hypothesis and claims

In Chapter 1 I stated the following hypothesis: "I propose that ranking the concepts in a semantic abstraction graph using graph-based ranking algorithms will yield the most important concepts of a biomedical scientific article." I believe that the work I present here shows that building SAGs and ranking the nodes in them using TextRank yields the most important concepts in a sample of the biomedical articles in PubMed Central.

I show that it is possible to construct these graphs in an unsupervised way, improving on Fiszman's original construction technique, which requires a seed node and pruning certain concepts (Fiszman et al., 2004). The graphs produced by my construction algorithm are compact (see Chapters 1, 3, and 5) and therefore highly connected, and have few important nodes (see Chapter 5). These features satisfy the criteria for my first claim: "that it is possible to build, in an unsupervised way, semantic abstraction graphs from scientific articles."

The most important nodes in the generated SAGs match the indexers' intent as much as possible. Although the measures in (Funk et al., 1983) are not directly comparable to the ones in this thesis, other authors claim that MTI is already close to inter-human indexing agreement (Névéol et al., 2007). Since MEDRank outperforms MTI, it is even closer to inter-human agreement, and thus satisfies my second and third claims: "that ranking the concept nodes in these SAGs yields the most important concepts in an article," and "that this approach, being grounded in a theory of the structure of scientific writing, performs better than the current state of the art in biomedical indexing (MTI)" respectively.

Why MEDRank works

Mihalcea's previous attempts to rank the nodes in semantic graphs using TextRank were successful. By analogy, an approach based on Semantic Abstraction Graphs (SAG) was therefore interesting to explore. Mihalcea attributed the success of TextRank, when using words, to the phenomenon that "co-occurring words recommend each other as important, and it is the common context that enables the identification of connections between words in text" (Mihalcea & Tarau, 2004). Suppe's theory of scientific writing gives Mihalcea's insight a stronger foundation. It is not, at least for scientific texts, merely that words recommend each other as important, but rather that the concepts those words describe are purposefully related to each other to weave an argument throughout a piece of writing.

MEDRank's advantages

MEDRank has several advantages over other indexing systems like the Medical Text Indexer (MTI). MEDRank separates Natural Language Processing (NLP) from indexing processing, it was created and evaluated using task-oriented measures, and it is consistent.

Consistency

Perhaps the most important advantage of automated indexers, including MEDRank, is that they are deterministic and, therefore, consistent with themselves. Unlike human indexers (Funk et al., 1983), whose performance is not reproducible, given the same input text MEDRank will always produce the same set of indexing terms. Consistent indexing may improve the usability of biomedical information retrieval systems like

MEDLINE by providing a more consistent experience than the current one.

Semantically aware vector cosine comparisons (SAVCC)

The strictness of Hooper's Indexing Consistency and traditional information retrieval measures like recall and precision, in which terms match completely or not at all, is not an adequate evaluation for the MEDLINE indexing task. Olivier Bodenreider, a research scientist at the NLM who is one of the authors of Restrict to MeSH and an authority on biomedical ontologies wrote, "I have argued for a long time that evaluating the quality of indexing in direct reference to the manual indexing in MEDLINE is too harsh. The idea is that, if your system comes close to, but not right on the MeSH descriptor assigned by the indexer, you don't get any credit for it, which is probably not fair, as "your" descriptor would likely do reasonably well in a retrieval task." (Olivier Bodenreider, 2008) Further, the correctness of any MEDLINE indexing is uncertain, since disagreement among experts producing the gold standard is large (Funk et al., 1983).

My use of SAVCC to evaluate the indexing process addresses this inherent ambiguity in the indexing task. The SAVCC computation accounts for the use of similar but not identical indexing terms. It therefore fulfills the need to evaluate the indexing task using a model that is closer to the retrieval task, which is the ultimate goal of every information retrieval system (Hersh, 2003).

Surpassing MTI's performance

MTI-indexer agreement is close to inter-human indexing agreement. It is also used to assist MEDLINE indexers. Since MTI is used to produce the gold standard itself, the gold standard is inherently biased. Although this bias is not quantifiable, it must make

it difficult to outperform MTI. MEDRank's design as a general-purpose biomedical indexer allows it to achieve higher precision at the expected expense of some recall. Higher general summary measures like Hooper's indexing consistency and SAVCC illustrate MEDRank's overall improvement in performance over MTI.

Limitations and future work

Practicality

MEDRank, and this thesis, have many limitations that need to be addressed in the future. The largest limitation to the practical application of MEDRank is its dependency on external NLP software. The number of articles that can be indexed successfully when compared to the MEDLINE (with MTI) gold standard is small. The lack of heuristics to compensate for poor NLP and edge cases make MEDRank strictly a research project at the moment. One way this limitation may be addressed in future work is by leveraging human knowledge by integrating the Related Citations (REL) indexer from MTI (see Chapter 2). Another potential way of addressing this limitation is implementing a voting scheme. Since MEDRank and MTI have different designs and implementations, accepting MeSH terms only if they are part of the output of both systems may improve precision significantly over using only one.

MEDRank could also be added as an alternate path to MTI. Since MTI implements a voting scheme to select final terms, MEDRank's output after ranking could be integrated to MTI's. This would allow MTI to support full text while adding a new, different source of data that would enrich its output.

Separation of NLP performance from indexing performance

Indexing performance depends greatly on NLP performance yet this is, to my knowledge, the first study to decouple NLP from indexing. Although decoupling the NLP performance from the ranking algorithm used to index is not a good representation of real-world performance, it allows me to isolate the performance of MEDRank from that of the NLP. It also illustrates an important fact: as NLP technology improves, MEDRank's performance is likely to improve.

Sentence splitting

The sentence splitter used in MEDRank is simplistic. It only recognizes two common bacteria, which means that many other bacteria are not recognized correctly. This, in turn, probably lowers MEDRank's precision and recall. An obvious future improvement is to use the Approved List of Bacterial Names (Skerman, McGowan, Sneath, Moore, & Moore, 1989) to recognize bacteria and avoid them during sentence splitting. This modification is already implemented and undergoing preliminary testing.

UMLS to MeSH mapping

MEDRank uses the same Restrict to MeSH algorithm that MTI uses. MTI post-processes data differently, and its ranking algorithm is linked to its post-processing. MEDRank's post-processing is simpler. It is unlikely that the changes in the mapping algorithm are the source of MEDRank's performance advantage over MTI. The other ranking algorithms explored in this thesis (TF*IDF, frequency ranking) share MEDRank's version of Restrict to MeSH and cannot outperform MTI.

I believe that using a different mapping algorithm constitutes a limitation of this study, and may introduce a confounding variable.

Evaluation limitations

The most significant limitation of this study is the quality of the gold standard. Current conditions at the NLM are different than when Funk and Reid studied inter-indexer consistency (Funk et al., 1983). Currently, trained experts index MEDLINE using MTI (among other tools). The impact of these tools, i.e., how the indexers use them, and how much the indexers rely on them affects results, but how much is unknown. It is possible, for example, that due to the use of automated tools, inter-indexer consistency has improved.

The ultimate goal of every information retrieval system is to allow its users to satisfy their information needs. My study, by adopting MEDLINE (with MTI) as its gold standard, currently assumes that the users' information needs are met optimally by MEDLINE (with MTI) indexing. This may be a false assumption. It is possible that a consistent automated indexing algorithm would better serve users' information needs than manual indexing. A user study is necessary, and will be performed in the future, to determine whether MEDRank's indexing is adequate, useful and, perhaps, better for users than MEDLINE (with MTI).

Another evaluation limitation is my use of a "bag of terms" model to compare the output of both systems. MeSH headings can be classified into different categories (checktags, major headings, qualifiers, etc.) according to their position in the MeSH hierarchy and annotations on the MEDLINE record itself. Inter-indexer consistency is different for MeSH headings in each category. Analyzing these separately would allow

me to compare my results directly to Funk and Reid's 1983 study on inter-indexer agreement (Funk et al., 1983).

Evaluating entire systems

Although I designed MEDRank to emulate MTI's processing pipeline as much as possible (Figure 16), the two systems are different in several ways. My current evaluation only studies the output of the entire system. For example, MTI relies on MetaMap instead of SEMREP. I tested SEMREP's output using other ranking algorithms, all of which performed worse than MEDRank. I thus can claim that SEMREP by itself is not solely responsible for MEDRank's performance. However, I did not isolate SEMREP's contribution to MEDRank's performance. Similarly, every parameter and design decision in which MEDRank diverges from MTI may have contributed (or impaired) MEDRank's performance relative to MTI. I believe that this is an intrinsic limitation of the study of competing indexing systems.

Other potential applications

Large-scale full text biomedical information retrieval systems that use MEDRank could potentially cluster similar documents together by using the generated terms and applying clustering algorithms. It could enable, for example, the identification of subjects for clinical studies based on analysis of the entire medical record. Alternatively, being able to index arbitrary full-text documents into MeSH could be used to create an integrated biomedical search engine that, unlike current offerings like Google Scholar, (http://scholar.google.com) accepts the PubMed queries that physicians and biomedical researchers already know how to compose. Conversely, it could also be used to expand PubMed by adding documents from the World Wide Web to its

collections seamlessly.

Suppe's theory assumes that writing is structured in such a way as to advance claims to construct an argument. It is therefore possible that the quality and clarity of the writing are related to the ability of MEDRank to detect the most important concepts in the text. This could be used to measure how well a piece of writing conforms to Suppe's ideal; in other words, it would be an objective measure of text quality. If the author supplies, for example, the most important topics in the text and those are compared to MEDRank's output, the degree of concordance could be used to judge how well the author has presented his or her views. This could be tested by comparing author-supplied keywords to MEDRank's output and correlating the difference between both to the number of citations a paper receives. Assuming that higher-quality papers receive more citations than lower-quality papers, there should be a correlation between these measures.

If the previous hypothesis is correct, graph-based ranking algorithms like MEDRank could be used as automated submission filters. These filters would allow authors to get a neutral second opinion on the quality of their writing. This in turn would allow journal editors and reviewers to spend their time working on articles that have already been vetted for clarity.

Conclusion

MEDRank is a new, innovative biomedical indexer that is based on an understanding of the structure of scientific papers and advances in text processing, graph theory, and graph-based ranking algorithms. It can outperform MTI even when compared to MEDLINE records indexed with MTI assistance.

References

Aronson, A. R. (2001, November 3-7). *Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program.* Paper presented at the AMIA Symposium, Washington, DC.

Aronson, A. R., Mork, J. G., Névéol, A., Shooshan, S. E., & Demner-Fushman, D. (2008, November 9-12). *Methodology for Creating UMLS Content Views Appropriate for Biomedical Natural Language Processing.* Paper presented at the AMIA Annual Symposium, Washington, DC.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval* (Vol. 1). New York, NY: Addison Wesley and ACM Press.

Bernstam, E. V., Herskovic, J. R., Aphinyanaphongs, Y., Aliferis, C. F., Sriram, M. G., & Hersh, W. R. (2006). Using Citation Data to Improve Retrieval from MEDLINE. *J Am Med Inform Assoc, 13*(1), 96-105.

Blake, J. B. (1986). From Surgeon General's bookshelf to National Library of Medicine: a brief history. *Bull Med Libr Assoc, 74*(4), 318-324.

Bodenreider, O. (2008) (personal communication). Thoughts on UMLS indexing performance. Houston, TX.

Bodenreider, O., Mitchell, J. A., & McCray, A. T. (2002). Evaluation of the UMLS as a terminology and knowledge resource for biomedical informatics. *Proc AMIA Symp*, 61-65.

Bondy, J. A., & Murty, U. S. R. (1976). *Graph Theory with Applications*. New York, NY:

Elsevier Science Publishing Co.

Dhyani, D., Ng, W. K., & Bhowmick, S. S. (2002). A Survey of Web Metrics. *ACM Computing Surveys, 34*(4), 469-503.

Dupuy, A., Khosrotehrani, K., Lebbe, C., Rybojad, M., & Morel, P. (2003). Quality of abstracts in 3 clinical dermatology journals. *Arch Dermatol, 139*(5), 589-593.

Eisner, J., & Smith, N. A. (2005, October 9-10). *Parsing with Soft and Hard Constraints on Dependency Length.* Paper presented at the Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT), Vancouver, BC, Canada.

Fellbaum, C. e. (1998). *Wordnet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Fiszman, M., Rindflesch, T. C., & Kilicoglu, H. (2004). *Abstraction Summarization for Managing the Biomedical Research Literature.* Paper presented at the Proc HLT NAACL Workshop on Computational Lexical Semantics.

Funk, M. E., Reid, C. A., & McGoogan, L. S. (1983). Indexing Consistency in MEDLINE. *Bull Med Libr Assoc, 71*(2), 176-183.

Gamon, M. (2006, June 9). *Graph-Based Text Representations for Novelty Detection.* Paper presented at the Workshop on TextGraphs, at HLT-NAACL, New York, NY.

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science, 178*(4060), 471-479.

Garfield, E. (2007). *Panel on Evaluative Measures for Resource Quality: Beyond the Impact Factor*. Paper presented at the Medical Library Association Meeting. Retrieved September 25, 2008, from

http://garfield.library.upenn.edu/papers/oakridge0907.pdf

Gay, C. W., Kayaalp, M., & Aronson, A. R. (2005). *Semi-automatic indexing of full text biomedical articles.* Paper presented at the AMIA Annu Symp Proc.

Hersh, W. R. (2003). *Information Retrieval: A Health and Biomedical Perspective* (Second ed.). New York, NY: Springer-Verlag.

Hersh, W. R., Hickam, D. H., Haynes, R. B., & McKibbon, K. A. (1994). A performance and failure analysis of SAPHIRE with a MEDLINE test collection. *J Am Med Inform Assoc, 1*(1), 51-60.

Herskovic, J. R., & Bernstam, E. V. (2005). *Using incomplete citation data for MEDLINE results ranking*. Paper presented at the American Medical Informatics Association Fall Symposium, Washington, DC.

Herskovic, J. R., Tanaka, L. Y., Hersh, W., & Bernstam, E. V. (2007). A Day in the Life of PubMed: Analysis of a Typical Day's Query Log. *J Am Med Inform Assoc, 14*(2), 212-220.

Kim, W., Aronson, A. R., & Wilbur, W. J. (2001). Automatic MeSH term assignment and quality assessment. *Proc AMIA Symp*, 319-323.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM, 46*(5), 604-632.

Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Matsuo, Y., & Ishizuka, M. (2004). Keyword Extraction from a Single Document using Word Co-ocurrence Statistical Information. *International Journal on Artificial*

Intelligence Tools, 13(1), 157-169.

Medelyan, O., & Witten, I. H. (2006). *Measuring inter-indexer consistency using a thesaurus*. Paper presented at the Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries.

Mihalcea, R. (2004). *Graph-based ranking algorithms for sentence extraction, applied to text summarization*. Paper presented at the Proceedings of the ACL 2004 (Interactive poster and demonstration sessions).

Mihalcea, R., & Moldovan, D. (2000). *Semantic Indexing using WordNet Senses.* Paper presented at the Proceedings of ACL Workshop on IR & NLP, Hong Kong.

Mihalcea, R., & Tarau, P. (2004, July). *TextRank: Bringing Order into Texts.* Paper presented at the Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain.

National Institutes of Health. (2008). *FY 2009 President's Budget Request: Congressional Justifications for Institutes and Centers: National Library of Medicine*. Retrieved October 29, 2008. from http://officeofbudget.od.nih.gov/ui/2008/NLM.pdf.

Névéol, A., Shooshan, S. E., Mork, J. G., & Aronson, A. R. (2007, November 10-14). *Fine-Grained Indexing of the Biomedical Literature: MeSH Subheading Attachment for a MEDLINE Indexing Tool* Paper presented at the AMIA Annual Symposium, Chicago, IL.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. Retrieved 03/15/2005, from http://dbpubs.stanford.edu:8090/pub/1999-66

Pedersen, T. (2000). *A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation*. Paper presented at the Proceedings of the first conference on North American chapter of the Association for Computational Linguistics.

Pitkin, R. M., Branagan, M. A., & Burmeister, L. F. (1999). Accuracy of Data in Abstracts of Published Research Articles. *JAMA, 281*(12), 1110-1111.

Pujol, J. M., Sanguesa, R., & Delgado, J. (2002). *Extracting reputation in multi agent systems by means of social network topology*. Paper presented at the Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Bologna, Italy.

Richardson, R., & Smeaton, A. F. (1995). Using WordNet in a Knowledge-Based Approach to Information Retrieval. *Proceedings of the BCS-IRSG Colloquium, Crewe*.

Rindflesch, T. C., Bean, C. A., & Sneiderman, C. A. (2000). Argument identification for arterial branching predications asserted in cardiac catheterization reports. *Proc AMIA Symp*, 704-708.

Ruiz, M. E., & Aronson, A. R. (2007). *User-centered Evaluation of the Medical Text Indexing (MTI) system*: U.S. National Library of Medicine. (U. S. N. L. o. Medicine). Retrieved 09/30/08 from http://ii.nlm.nih.gov/resources/MTIEvaluation-Final.pdf

Salton, G. (1963). Associative Document Retrieval Techniques Using Bibliographic Information. *J. ACM, 10*(4), 440-457.

Savova, G. K., Coden, A. R., Sominsky, I. L., Johnson, R., Ogren, P. V., Groen, P. C., et al.

(2008). Word sense disambiguation across two domains: Biomedical literature and clinical notes. *Journal of Biomedical Informatics, 41*(6), 1088-1100.

Schuemie, M. J., Kors, J. A., & Mons, B. (2005). Word sense disambiguation in the biomedical domain: an overview. *J Comput Biol, 12*(5), 554-565.

Skerman, V. B. D., McGowan, V., Sneath, P. H. A., Moore, W. E. C., & Moore, L. V. H. (1989). *Approved Lists of Bacterial Names* Washington, D.C.: ASM Press.

Suppe, F. (1998). The Structure of a Scientific Paper. *Philos Sci, 65*(3), 381-405.

U.S. National Library of Medicine. (1999, October 30, 2007). Medical Subject Heading (MESH) Fact Sheet. *Fact Sheets* Retrieved September 25, 2008, from http://www.nlm.nih.gov/pubs/factsheets/mesh.html

U.S. National Library of Medicine. (2003, January 01, 2008). OLDMEDLINE Data. Retrieved September 24, 2008, from http://www.nlm.nih.gov/databases/databases_oldmedline.html

U.S. National Library of Medicine. (2004a, March 16). Clustering and Ranking process. Retrieved November 6, 2008, from http://ii.nlm.nih.gov/MTI/cluster.shtml

U.S. National Library of Medicine. (2004b, March 16). MetaMap Indexing Algorithm. Retrieved November 6, 2008, from http://ii.nlm.nih.gov/MTI/mmi.shtml

U.S. National Library of Medicine. (2004c, March 16). Restrict to MeSH algorithm. Retrieved November 6, 2008, from http://ii.nlm.nih.gov/MTI/RTM.shtml

U.S. National Library of Medicine. (2004d, February 19, 2005). The Story of NLM Historical Collections. *History of Medicine* Retrieved September 23, 2008, from http://www.nlm.nih.gov/hmd/about/collectionhistory.html

U.S. National Library of Medicine. (2004e, March 16). Trigram Algorithm. Retrieved November 6, 2008, from http://ii.nlm.nih.gov/MTI/trigram.shtml

U.S. National Library of Medicine. (2006a, November 27). MeSH history. Retrieved March 30, 2007, from http://www.nlm.nih.gov/mesh/intro_preface2007.html#pref_hist

U.S. National Library of Medicine. (2006b, 03/23/2006). Unified Medical Language System Fact Sheet. *Fact Sheets* Retrieved 11/08/2006, from http://www.nlm.nih.gov/pubs/factsheets/umls.html

Wellisch, H. H. (1991). *Indexing from A to Z* Bronx, NY: The H. W. Wilson Company.

Widdows, D., & Dorow, B. (2002, August). *A Graph Model for Unsupervised Lexical Acquisition.* Paper presented at the 19th International Conference on Computational Linguistics, Taipei.

Index of figures