Technical Report

UCAM-CL-TR-848
ISSN 1476-2986

Number 848

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Automatically generating reading lists

James G. Jardine

February 2014

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted August 2013 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Robinson College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

This thesis addresses the task of automatically generating reading lists for novices in a scientific field. Reading lists help novices to get up to speed in a new field by providing an expert-directed list of papers to read. Without reading lists, novices must resort to ad-hoc exploratory scientific search, which is an inefficient use of time and poses a danger that they might use biased or incorrect material as the foundation for their early learning.

The contributions of this thesis are fourfold. The first contribution is the ThemedPageRank (TPR) algorithm for automatically generating reading lists. It combines Latent Topic Models with Personalised PageRank and Age Adjustment in a novel way to generate reading lists that are of better quality than those generated by state-of-the-art search engines. TPR is also used in this thesis to reconstruct the bibliography for scientific papers. Although not designed specifically for this task, TPR significantly outperforms a state-of-the-art system purpose-built for the task. The second contribution is a gold-standard collection of reading lists against which TPR is evaluated, and against which future algorithms can be evaluated. The eight reading lists in the gold-standard were produced by experts recruited from two universities in the United Kingdom. The third contribution is the Citation Substitution Coefficient (CSC), an evaluation metric for evaluating the quality of reading lists. CSC is better suited to this task than standard IR metrics such as precision, recall, F-score and mean average precision because it gives partial credit to recommended papers that are close to gold-standard papers in the citation graph. This partial credit results in scores that have more granularity than those of the standard IR metrics, allowing the subtle differences in the performance of recommendation algorithms to be detected. The final contribution is a light-weight algorithm for Automatic Term Recognition (ATR). As will be seen, technical terms play an important role in the TPR algorithm. This light-weight algorithm extracts technical terms from the titles of documents without the need for the complex apparatus required by most state-of-the-art ATR algorithms. It is also capable of extracting very long technical terms, unlike many other ATR algorithms.

Four experiments are presented in this thesis. The first experiment evaluates TPR against state-of-the-art search engines in the task of automatically generating reading lists that are comparable to expert-generated gold-standards. The second experiment compares the performance of TPR against a purpose-built state-of-the-art system in the task of automatically reconstructing the reference lists of scientific papers. The third experiment involves a user study to explore the ability of novices to build their own reading lists using two fundamental components of TPR: automatic technical term recognition and topic modelling. A system exposing only these components is compared against a state-of-the-art scientific search engine. The final experiment is a user study that evaluates the technical terms discovered by the ATR algorithm and the latent topics generated by TPR. The study enlists thousands of users of Qiqqa, research management software independently written by the author of this thesis.

Acknowledgements

I would like to thank my supervisor, Dr. Simone Teufel, for allowing me the room to develop my ideas independently from germination to conclusion, and for dedicating so much time to guiding me through the writing-up process. I thank her for the many interesting and thought-provoking discussions we had throughout my graduate studies, both in Cambridge and in Edinburgh.

I am grateful to the Computer Laboratory at the University of Cambridge for their generous Premium Research Studentship Scholarship. Many thanks are due to Stephen Clark and Ted Briscoe for their continued and inspiring work at the Computer Laboratory. I am also grateful to Nicholas Smit, my accomplice back in London, and the hard-working committee members of Cambridge University Entrepreneurs and the Cambridge University Technology and Enterprise Club for their inspiration and support in turning Qiqqa into world-class research software.

I will never forget my fellow Robinsonians who made the journey back to university so memorable, especially James Phillips, Ross Tokola, Andre Schwagmann, Ji-yoon An, Viktoria Moltz, Michael Freeman and Marcin Geniusz. Reaching further afield of College, University would not have been the same without the amazing presences of Stuart Barton, Anthony Knobel, Spike Jackson, Stuart Moulder and Wenduan Xu.

I am eternally grateful to Maïa Renchon for her loving companionship and support through some remarkably awesome and trying times, to my mother, Marilyn Jardine, for inspiring me to study forever, and to my father, Frank Jardine, for introducing me to my first “thinking machine”.

Table of Contents

Table of Figures

Table of Tables

Chapter 1.

Introduction

This thesis addresses the task of automatically generating reading lists for novices in a scientific field. The goal of a reading list is to quickly familiarise a novice with the important concepts in their field. A novice might be a first-year research student or an experienced researcher transitioning into a new discipline. Currently, if such a novice receives a reading list, it has usually been manually created by an expert.

Reading lists are a commonly used educational tool in science (Ekstrand et al. 2010). A student will encounter a variety of reading lists during their career: a list of text books that are required for a course, as prescribed by a professor; a list of recommended reading at the end of a book chapter; or the list of papers in the references section of a journal paper. Each of these reading lists has a different purpose and a different level of specificity towards the student, but in general, each list is generated by an expert.

A list of course textbooks details the material that a student must read to follow the lectures and learn the foundations of the field. This reading list is quite general in that it is applicable to a variety of students. The list of reading at the end of a textbook chapter might introduce more specialised reading. It is intended to guide students who wish to explore a field more deeply. The references section of a journal paper is more specific again: it suggests further reading for a particular research question, and is oriented towards readers with more detailed technical knowledge of a field. Tang (2008) describes how the learner-models of each individual learner are important when making paper recommendations. These learner-models are comprised of their competencies and interests, the landscape of their existing knowledge and their learning objectives.

The most specific reading list the student will come across is a personalised list of scientific papers generated by an expert, perhaps a research supervisor, spanning their specialised field of research. Many experts have ready-prepared reading lists they use for teaching, or can produce one on the fly from their domain knowledge should the need arise. After reading and understanding this list, the student should be in a good position to begin independent novel scientific research in that field.

Despite their potential usefulness, access to structured reading lists of scientific papers is generally only available to novices who have access to guidance of an expert. What can a novice do if an expert is not available to direct their reading?

Experts in a field are accustomed to strategic reading (Renear & Palmer 2009), which involves searching, filtering, scanning, linking, annotating and analysing fragments of content from a variety of sources. To do this proficiently, experts rely on their familiarity

with advanced search tools, their prior knowledge of their field, and their awareness of technical terms and ontologies that are relevant to their domain. Novices lack all three proficiencies.

While a novice will benefit from a reading list of core papers, they will benefit substantially more from a review of the core papers, where each paper in the list is annotated with a concise description of its content. In some respect, reading lists are similar to reviews in that they shorten the time it takes to get the novice up to speed to start their own research (Mohammad et al. 2009a), both by locating the seminal papers that initiated inquiry into the field and by giving them a sufficiently complete overview of the field. While automatically generating reading lists does not tackle the harder task of generating review summaries of papers, it can provide a good candidate list of papers to automatically review.

Without expert guidance, either in person or through the use of reading lists, novices must resort to exploratory scientific search – an impoverished imitation of strategic reading. It involves the use of electronic search engines to direct their reading, initially from a first guess for a search query, and later from references, technical terms and authors they have discovered as they progress in their reading. It is a cyclic process of searching for new material to read, reading and digesting this new material, and expanding awareness and knowledge so that the process can be repeated with better search criteria.

This interleaved process of searching, reading, expanding is laborious, undirected, and highly dependent on an arbitrary starting point, even when supported by online search tools (Wissner-Gross 2006). To compound matters, the order in which material is read is important. Novices do not have the experience in a new field to differentiate between good and bad papers (Wang et al. 2010). They therefore read and interpret new material in the context of previously assimilated information (Oddy et al. 1992). Without a reading list, or at least some guidance from an expert, there is a danger that the novice might use biased, flawed or incorrect material as the foundation for their early learning. This unsound foundation can lead to misjudgements of the relevance of later reading (Eales et al. 2008).

It is reasonable to advocate that reading lists are better than exploratory scientific search for cognitive reasons. Scientific literature contains opaque technical terms that are not obvious to a novice, both when formulating search queries and when interpreting search results (Kircz 1991; Justeson & Katz 1995). How should a novice approach exploratory scientific search when they are not yet familiar with a field, and in particular, when they are not yet familiar with the technical terms? Technical terms are opaque to novices because they have particular meaning when used in a scientific context (Kircz 1991) and because synonymous or related technical terms are not obvious or predictable to them. Keyword search is thus particularly difficult for them (Bazerman 1985). More importantly, novices – and scientists in general – are often more interested in the relationships between scientific facts than the isolated facts themselves (Shum 1998).

Without reading lists a novice has to repeatedly formulate search queries using unfamiliar technical terms and digest search results that give no indication of the relationships between papers. Reading lists are superior in that they present a set of relevant papers covering the most important areas of a field in a structured way. From a list of relevant papers, the novice has an opportunity to discover important technical terms and scientific facts early on in their learning process and to better grasp the relationships between them.

Reading lists are also better than exploratory scientific search for technical reasons. The volume of scientific literature is daunting, and is growing exponentially (Maron & Kuhns 1960; Larsen & von Ins 2009). While current electronic search tools strive to ensure that the novice does not miss any relevant literature by including in the search results as many matching papers as they can find, these thousands of matching papers returned can be overwhelming (Renear & Palmer 2009). Reading lists are of a reasonable and manageable length by construction. When trying to establish relationships between papers using exploratory scientific search, one obvious strategy is to follow the citations from one paper to the next. However, this strategy rapidly becomes intractable as it leads to an exponentially large set of candidate papers to consider. The search tools available for exploratory scientific search also do little to reduce the burden on the novice in deciding the authority or relevance of the search results. Many proxies for authority have been devised such as citation count, h-index score and impact factor, but so far, these have been broad measures and do not indicate authority at a level of granularity needed by a novice in a niche area of a scientific field. Reading lists present a concise, authoritative list of papers focussed on the scientific area that is relevant to the novice.

The first question this research addresses is whether or not experts can make reading lists when given instructions, and explores how they go about doing so. This question is answered with the assembly of a gold-standard set of reading lists created by experts, as described in Section 3.2.

While the primary focus of this research is the automatic generating of reading lists, the algorithms that I develop for automatically generating reading lists rely on both the technical terms in a scientific field and the relationships between these technical terms and the papers associated with them. These relationships are important for this thesis, and arise from my hypothesis that similar technical terms appear repeatedly in similar papers. These relationships make possible the useful extrapolation that a paper and a technical term can be strongly associated even if the term is not used in the paper. As a step towards automatically generating reading lists, this thesis will confirm that these technical terms and relationships are useful for the automatic generation of reading lists. In addition this thesis will explore the hypothesis that exploratory scientific search can be improved upon with the addition of features that allow novices to explore these technical terms and relationships.

The second question this research addresses is whether or not the exposition of relationships between papers and their technical terms improves the performance of a

novice in exploratory scientific search. This question is answered using the task-based evaluation described in Section 5.4.

The algorithms that I develop for automatically generating reading lists make use of two distinct sources of information: lexical description and social context. These sources of information are used to model scientific papers, to find relationships between them, and to determine their authority.

Lexical description deals with the textual information contained in each paper. It embodies information from inside a paper, i.e. the contributions of a paper from the perspective of its authors. In the context of this thesis, this information consists of the title, the full paper text, and the technical terms contained in that text. I use this information to decide which technical terms are relevant to each paper, to divide the corpus into topics, to measure the relevance of the papers to each topic, and to infer lexical similarities and relationships between the papers, technical terms and the topics.

Social context deals with the citation behaviour between papers. It embodies information from outside a paper, i.e. the contribution, relevance and authority of each paper from the perspective of other people. This information captures the fact that the authors of one paper chose to cite another paper for some reason, or that one group of authors exhibits similar citing behaviour to another group of authors. I use this information to measure the authority of papers and to infer social similarities and relationships between them.

These lexical and social sources of information offer different advantages when generating reading lists, and their strengths can be combined in a variety of ways. Some search systems use only the lexical information, e.g., TFIDF indexed search, topic modelling, and document clustering. Some use only social information, e.g. co-citation analysis, citation count and h-index, and collaborative filtering. More complex search systems combine the two in various ways, either as independent features in machine learning algorithms or combined more intricately to perform better inference. Much of Chapter 2 is dedicated to describing these types of search systems. The algorithms developed in this thesis fall into the last category, where lexical information is used to discover niches in scientific literature, and social information is used to find authority inside those niches.

The third question this research addresses is whether or not lexical and social information contributes towards the task of automatically generating reading lists, and if so, to measure the improvement of such algorithms over current state-of-the-art. It turns out that they contribute significantly, especially in combination, as will be shown in the experiments in Sections 5.2 and 5.3.

The task of automatically generating reading lists is a recent invention and so standardised methods of evaluation have not yet been established. Methods of evaluation fall into three major categories: offline methods, or “the Cranfield tradition” (Sanderson 2010); user-centred studies (Kelly 2009); and online methods (Kohavi et al. 2009). From these major categories, four specific evaluations are performed in this thesis: a gold-

standard-based evaluation (offline method); a dataset-based evaluation (offline method); a task-based evaluation (user-centred study); and a user satisfaction evaluation (online method).

Gold-standard-based evaluations test a system against a dataset specifically created for particular experiments. This allows a precise hypothesis to be tested. However, creation of a gold-standard is expensive so evaluations are generally small in scale. A gold-standard-based evaluation is used in Section 5.2 to compare the quality of the reading lists automatically generated by various algorithms against a gold-standard set of reading lists generated by experts in their field.

Because gold-standards tailored to a particular hypothesis are expensive to create, it is sometimes reasonable to transform an existing dataset (or perhaps a gold-standard from a different task) into a surrogate gold-standard. These are cheaper forms of evaluation as they leverage existing datasets to test a hypothesis. They operate at large scale, which facilitates drawing statistically significant conclusions, and generally have an experimental setup that is repeatable, which enables other researchers to compare systems independently. A disadvantage is that large datasets are generally not tailored to any particular experiment and so proxy experiments must be performed instead. Automated evaluation is used in Section 5.3 to measure the quality of automatically generated reading lists though the proxy test of reconstructing the references sections of 1,500 scientific papers.

Task-based evaluations are the most desirable at testing hypotheses because they elicit human feedback from experiments specifically designed for the task. However, this makes them expensive – both in the requirement of subjects to perform the task and experts to judge their results. They also require significant investment in time to coordinate the subjects during the experiment. A task-based evaluation is presented in Section 5.4. It explores whether the exposition of relationships between papers and their technical terms improves the performance of a novice in exploratory scientific search.

User satisfaction evaluations have the advantage of directly measuring human response to a hypothesis. Once deployed, they also can scale to large sample populations without additional effort. A user satisfaction evaluation is used in Section 5.5 to evaluate the quality of the technical terms and topic models produced by my algorithms.

In summary, this thesis addresses the task of automatically generating reading lists for novices in a scientific field. The exposition of this thesis is laid out as follows. Chapter 2 positions the task of automatically generating reading lists within a review of related research. The two most important concepts presented there are Latent Topic Models and Personalised PageRank, which are combined in a novel way to produce one of the major contributions of this thesis, ThemedPageRank. Chapter 3 develops ThemedPageRank in detail, along with the four other contributions of this thesis, while Chapter 4 describes their technical implementation. Chapter 5 presents two experiments that compare the performance of ThemedPageRank with state-of-the-art in the two tasks of automated

reading list construction and automated reference list reintroduction. Two additional experiments enlist human subjects to evaluate the performance of the artefacts that go into the construction of ThemedPageRank. Finally, Chapter 6 concludes with a summary of this thesis and discusses potential directions for future work.

Chapter 2.

Related work

The task of automatically generating reading lists falls broadly into the area of Information Retrieval, or IR (Mooers 1950; Manning et al. 2008). According to Fairthorne (2007), the purpose of an IR system is to structure a large volume of information in such a way as to allow a search user to efficiently retrieve the subset of this information that is most relevant to their information need. The information need is expressed in a way that is understandable by the searcher and interpretable by the IR system, and the retrieved result is a list of relevant items. When automatically generating reading lists, a novice's information need, approximated by a search query, must be satisfied by a relevant subset of papers found in a larger collection of papers (a document corpus).

2.1 Information Retrieval

Almost any type of information can be stored in an IR system, ranging from text and video, to medical or genomic data. In line with the task of automatically generating reading lists, this discussion describes IR systems that focus on textual data – specifically information retrieval against a repository of scientific papers.

An IR system is characterised by its retrieval model, which is comprised of an indexing and a matching component (Manning et al. 2008). The task of the indexing component is to transform each document into a document representation that can be efficiently stored and searched, while retaining much of the information of the original document. The task of the matching component is to translate a search query into a query representation that can be efficiently matched or scored against each document representation in the IR system. This produces a set of document representations that best match the query representation, which in turn are transformed back into their associated documents as the search results.

The exact specification of the retrieval model is crucial to the operation of the IR system: it decides the content and the space requirements of what is stored inside the IR system, the syntax of the search queries, the ability to determine relationships between documents inside the IR system, and the efficiency and nature of scoring and ranking of the search results. Increasingly complex retrieval models are the subject of continued and active research (Voorhees et al. 2005).

The Boolean retrieval model underlies one of the earliest successful information retrieval search systems (Taube & Wooster 1958). Documents are represented by an unordered multi-set of words (the bag-of-words model), while search queries are expressed as

individual words separated by Boolean operators (i.e. AND, OR and NOT) with well-known semantics (Boole 1848). A document matches a search query if the words in the document satisfy the set-theoretic Boolean expression of the query. Matching is binary: a document either matches or it does not. The Boolean retrieval model is useful at retrieving all occurrences of documents containing matching query keywords, but it has no scoring mechanism to determine the degree of relevance of individual search results. Moreover, in searchers' experience, the Boolean retrieval is generally too restrictive when using AND operators and too overwhelming when using OR operators (Lee & Fox 1988).

The TFIDF retrieval model (Sparck-Jones 1972) addresses the need for scoring the search results to indicate the degree of relevance to the search query of each search result. The intuitions behind TFIDF are twofold. Firstly, documents are more relevant if search terms appear frequently inside them. This phenomenon is modelled by "term frequency", or TF. Secondly, search terms are relatively more important or distinctive if they appear infrequently in the corpus as a whole. This phenomenon is modelled by the "inverse document frequency", or IDF.

TFIDF is usually implemented inside the vector-space model (Salton et al. 1975) where documents are represented by *T*-dimensional vectors. Each dimension of the vector corresponds to one of the *T* terms in the retrieval model and each dimension value is the TFIDF score for term *t* in document *d* in corpus *D*,

$$r_d = \begin{bmatrix} TFIDF_{1,d,D} \\ \vdots \\ TFIDF_{T,d,D} \end{bmatrix}$$

Salton & Buckley (1988) describe a variety of TFIDF-based term weighting schemes and their relative advantages and disadvantages, but commonly

$$TFIDF_{t,d,D} = TF_{t,d} \times IDF_{t,D}$$

$$IDF_{t,D} = log \frac{|D|}{|D_t|}$$

where $TF_{t,d}$ is the frequency of term *t* in document *d*, $|D|$ is the number of documents in the corpus, and $|D_t|$ is the number of documents in the corpus containing term *t*.

Similarly, a query vector representation is the TFIDF score for each term *t* in query *q*

$$r_q = \begin{bmatrix} TFIDF_{1,q,D} \\ \vdots \\ TFIDF_{T,q,D} \end{bmatrix}$$

The relevance score for a document is measured by the similarity between the query representation and the document representation. One such similarity measure is the normalised dot product of the two representations,

$$score_{d,q} = \frac{r_d \cdot r_q}{|r_d| \quad |r_q|}$$

This score, also called the cosine score, allows results to be ranked by relevance: retrieved items with larger scores are ranked higher in the search results.

The TFIDF vector-space model represents documents using a mathematical construct that does not retain much of the structure of the original documents. The use of a term-weighted bag-of-words loses much of the information in the original document such as word ordering and section formatting. However, this loss of information is traded off against the benefits of efficient storage and querying.

While traditional IR models like the Boolean and TFIDF retrieval models address the task of efficiently retrieving information relevant to a searcher's need, their designs take little advantage of the wide variety of relationships that exist among the documents they index.

Shum (1998) argues that scientists are often more interested in the relationships between scientific facts than the facts themselves. This observation might be applied to papers too because papers are conveyors of facts. The relationships between the papers are undoubtedly interesting to scientists because tracing these relationships is a major means for a scientist to learn new knowledge and discover new papers (Renear & Palmer 2009).

One place to look for relationships between papers is the paper content itself. By analysing and comparing the lexical content of papers we can derive lexical relationships and lexical similarities between the papers. The intuition is that papers are somehow related if they talk about similar things.

A straightforward measure of relationship between two papers calculates the percentage of words they have in common. This is known as the Jaccard similarity in set theory (Manning et al. 2008), and is calculated as

$$similarity_{d1,d2} = \frac{|w_{d1} \cap w_{d2}|}{|w_{d1} \cup w_{d2}|}$$

where w_{d1} and w_{d2} are the sets of words in documents *d1* and *d2*, respectively. It is intuitive that documents with most of their words in common are more likely to be similar than documents using completely different words, so a larger overlap implies a stronger relationship. All words in the document contribute equally towards this measure, which is not always desirable. Removing words such as articles, conjunctions and pronouns (often called stop-words) can improve the usefulness of this measure.

The TFIDF vector-space model (Salton et al. 1975), discussed previously in the context of information retrieval, can also be leveraged to provide a measure of the lexical similarity of two documents using the normalised dot product of the two paper representations. The TFIDF aspect takes into account the relative importance of words inside each document when computing similarity:

$$similarity_{d1,d2} = \frac{r_{d1} \cdot r_{d2}}{|r_{d1}| \quad |r_{d2}|}$$

In this thesis I focus on the technical terms that are contained by each document, and model documents as a bag-of-technical-terms rather than a bag-of-words. This is motivated from three perspectives.

Firstly, Kircz (1991) and Justeson & Katz (1995) describe the importance of technical terms in conveying the meaning of a scientific document, while at the same time highlighting the difficulty a novice faces in assimilating them. Shum (1998) argues that many information needs in scientific search entail solely the exposition of relationships between scientific facts. By using technical terms instead of words, there is an opportunity to find the relationships between these technical terms in the literature. Secondly, the distributions of words and technical terms in a document are both Zipfian (Ellis & Hitchcock 1986), so the distributional assumptions underlying many similarity measures are retained when switching from a bag-of-words to a bag-of-technical-terms model. Thirdly, many IR systems exhibit linear or super-linear speedup in a reduction in the size of the underlying vocabulary (Newman et al. 2006). Obviously, the vocabulary of technical terms in a corpus is smaller than the vocabulary of all words in a corpus, so using technical terms should also lead to a noticeable decrease in search time.

2.2 Latent Topic Models

While changing the representation of documents from bag-of-words to bag-of-technical-terms has all the advantages just described, it still suffers from the same two problems that plague the bag-of-words model: polysemy and synonymy. Two documents might refer to identical concepts with different terminology, or use identical terminology for different concepts. Naïve lexical techniques are unable to directly model these substitutions without enlisting external resources such as dictionaries, thesauri and ontologies (Christoffersen 2004). These resources might be manually produced, such as WordNet (Miller 1995), but they are expensive and brittle to domain shifts. This applies particularly to resources that cater towards technical terms, such as gene names (Ashburner et al. 2000). Alternatively, the resources might be automatically produced, which is non-trivial and amounts to shifting the burden from the naïve lexical techniques elsewhere (Christoffersen 2004).

Latent topic models consider the relationships between entire document groups and have inherent mechanisms that are robust to polysemy and synonymy (Steyvers & Griffiths 2007; Boyd-Graber et al. 2007). They automatically discover latent topics – latent groupings of concepts – within an entire corpus of papers, and latent relationships between technical terms in a corpus. Synonyms tend to be highly representative in the same topics, while words with multiple meanings tend to be represented by different topics (Boyd-Graber et al. 2007). Papers with similar topic distributions are likely to be related because they frequently mention similar technical terms. These same distributions over topics also expose relationships between papers and technical terms.

In addition to automatically coping with polysemy and synonymy, quantitatively useful latent relationships between papers emerge when various topic modelling approaches are applied to a corpus of papers. Latent topic models are able to automatically extract scientific topics that have the structure to form the basis for recommending citations (Daud 2008), and the stability over time to track the evolution of these scientific topics (He et al. 2009).

It is important to bear in mind that while these automated topic models are excellent candidates for dissecting the structure of a corpus of data, their direct outputs lack explicit structure. Topics are imprecise entities that emerge only through the strengths of association between documents and technical terms that comprise them, so it is difficult to interpret and differentiate between them. To help with interpretation, one might manually seed each topic with pre-identified descriptive terms, but this approach is not scalable and requires knowledge of the topics in a document corpus beforehand (Andrzejewski & Zhu 2009). This problem becomes even more difficult as the number of topics grows (Chang et al. 2009a; Chang et al. 2009b).

Sidestepping the issue about their interpretability, topics can be used internally as a processing stage for some larger algorithm. This has proved invaluable for variety of tasks, ranging from automatically generating image captions (Blei 2004) to automatic spam detection on the Internet (Wu et al. 2006).

Wsd	HMM	POS
WSD	markov	part-of-speech
word sense disambiguation	hidden markov	part of speech
Wordnet	DP	pos tagging
word senses	EM	part-of-speech tagging
senseval-3	markov models	ME
LDA	hidden markov model	pos tagger
computational linguistics	training data	rule-based
training data	hidden markov models	natural language
english lexical	dynamic programming	parts of speech
information retrieval	np	crf
Ir	NP	CRF
IR	VP	EM
TREC	PP	training data
text retrieval	nps	perceptron
query expansion	phrase structure	unlabeled data
IDF	parse trees	active learning
search engine	syntactic structure	semi-supervised
retrieval system	noun phrase	reranking
document retrieval	verb phrase	conditional random fields

Table 1. Examples of Topics Generated by LDA from a Corpus of NLP Papers.

To provide some idea of the nature of the topics produced by topic modelling, Table 1 shows an example of six topics generated by my implementation of a popular model for topic modelling, Latent Dirichlet Allocation (LDA) (Blei et al. 2003), over a typical NLP document collection, the ACL Anthology Network (Radev et al. 2009b). The top-10 most relevant technical terms are shown for six topics chosen arbitrarily from the 200 topics generated by LDA. Notice how synonymous technical terms congregate in the same topic. Also notice that while each topic is recognisable as an area of NLP, it is not straightforward to label each topic or to discern the boundaries of each topic. For example, the first topic might easily be labelled "word sense disambiguation" because most of the technical terms that comprise the topic are closely aligned with word sense disambiguation. However, the sixth topic contains a variety of technical terms that are loosely associated with machine learning, but are not similar enough to adequately label the entire topic.

Superficially, topic models collapse the sparse high-dimensional document-bag-of-technical-terms representation of a corpus of documents into a lower-dimensional representation where documents are represented by a mixture of topics and topics by a mixture of technical terms. Figure 1 shows how the sparse matrix Ω, containing the counts of V technical-terms (the columns) in each of the D documents (the rows), can be approximated by the combination of two smaller, but denser matrices Θ, containing the document-topic representation, and Φ, containing the topic-technical-term representation.

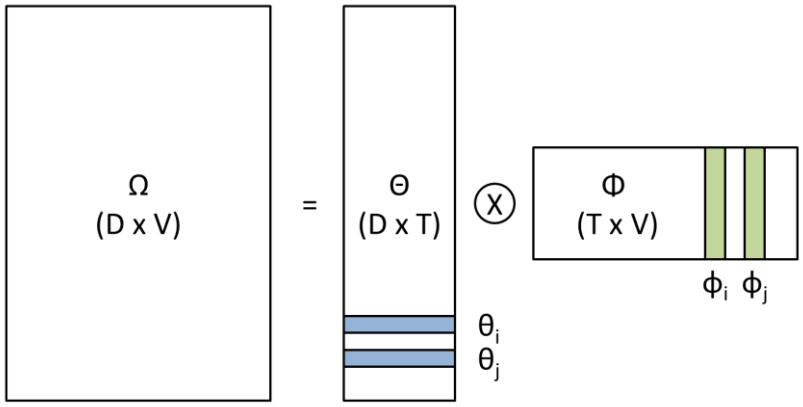


Figure 1. A High-Level Interpretation of Topic Modelling.

Matrix Θ contains the distributions of documents over the latent topics. Each row of the matrix corresponds to a document, and each column in that row specifies how strongly a latent topic applies to that document. Documents are related if they exhibit similar distributions (Steyvers & Griffiths 2007). One technique for measuring the similarity between two documents is the normalised dot-product of their distribution vectors θ_i and

θ_j: a larger normalised product indicates a stronger relationship between the documents. If matrix Θ contains probability distributions (i.e. each row sums to unity), another technique is to measure the Jensen-Shannon divergence (Lin 2002) between their probability distribution vectors: a smaller divergence indicates a stronger relationship.

2.2.1 Latent Semantic Analysis

One type of Latent Topic Model is Latent Semantic Analysis (LSA), which uses Singular Value Decomposition (SVD) to discover latent topics in a corpus (Deerwester et al. 1990).

SVD is performed on the sparse matrix Ω to produce three matrices:

$$\Omega_{\mathrm{D}\times\mathrm{V}} = \mathrm{U}_{\mathrm{D}\times\mathrm{D}}\Sigma_{\mathrm{D}\times\mathrm{V}}V^T_{V\times V}$$

where U and V^T are unitary matrices and $\sum$ is a $d\times v$ diagonal matrix. Truncating the columns of U and rows of V^T corresponding to the largest t singular values in $\sum$, U and V^T become $d\times t$ and $t\times v$ matrices and $\sum$ becomes a $t\times t$ diagonal matrix. Multiplying $\sum$ with either U or V^T produces the representation in Figure 1.

The number of rows and columns remaining after truncation corresponds to the desired number of latent topics, K. The values in Θ and Φ have no meaningful interpretation – they are merely positive and negative values whose product gives the best rank-K approximation (under the Frobenius norm measure) to Ω.

The lack of interpretability of matrices Θ and Φ is the main criticism against SVD (Hofmann 1999). Another criticism is based on the underlying distribution of words in language and whether SVD theoretically is the right tool to model such a distribution. SVD models joint Gaussian data best – particularly under the assumption that eliminating the smallest singular values is Frobenius-optimal (Hofmann 1999). However, word distribution in language is known to be Zipfian (Zipf 1949) and not Gaussian. Ellis & Hitchcock (1986) show that the adoption and use of technical terms in language is also Zipfian. This incorrect underlying theoretical assumption about the distribution of words in documents may limit the applicability of LSA in discovering latent topics in document corpora.

In an attempt to redress this criticism of LSA, Probabilistic Latent Semantic Analysis (PLSA) (Hofmann 1999) was developed upon a more principled statistical foundation than the generic algebraic technique of LSA. It is based on a mixture decomposition derived from a latent class aspect model to decompose documents d_i and words w_i into latent topics z_i.

Figure 2 shows two graphical model representations for PSLA. A document corpus can be modelled as shown in Figure 2(a) by

$$P(d,w) = P(d)P(w|d) = P(d)\sum_{z\in Z} P(w|z)P(z|d)$$

Or as shown in Figure 2(b) by

$$P(d,w) = \sum_{z \in Z} P(z)P(d|z)P(w|z)$$

P(d,w) can be inferred from a corpus using Expectation Maximisation (Dempster et al. 1977). Using the model in Figure 2(b), multiplying *P(z)* with either *P(d|z)* or *P(w|z)* produces the representation in Figure 1.

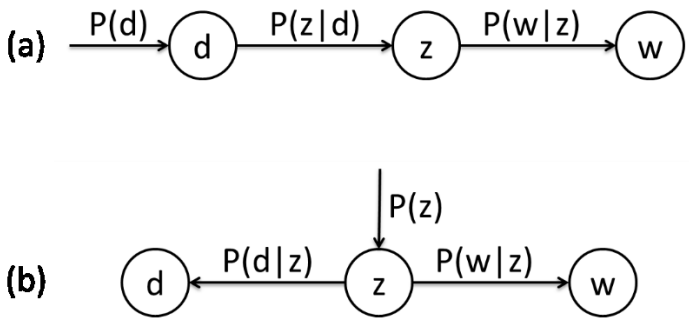


Figure 2. Graphical Model Representations of PLSA.

While the latent topics of PSLA resolve the joint Gaussian limitation of LSA, neither LSA nor PLSA implicitly supports a generative model for documents. After calculation of the initial LSA or PLSA model, later additions of documents to a corpus cannot be modelled without recalculating the model from scratch.

LSA and PLSA are also prone to over-fitting because there is no mechanism for specifying priors over any of the inferred distributions. Thus they do not adequately model under-represented documents (Blei et al. 2003). Latent Dirichlet Allocation was developed to address both these drawbacks.

2.2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) (Blei et al. 2003) is a Bayesian generative probabilistic model for collections of discrete data. Although it has been applied to a variety of data modelling problems, it has become particularly popular for the modelling of scientific text corpora (Wei & Croft 2006; He et al. 2009; Blei & Lafferty 2007; Daud 2008). In this thesis I will use LDA predominantly to produce the latent topics that express the relationships between papers and technical terms needed for the algorithms that automatically generate reading lists.

In LDA, a document in the corpus is modelled and explicitly represented as a finite mixture over an underlying set of topics, while each topic is modelled as an infinite mixture over the underlying set of words in the corpus.

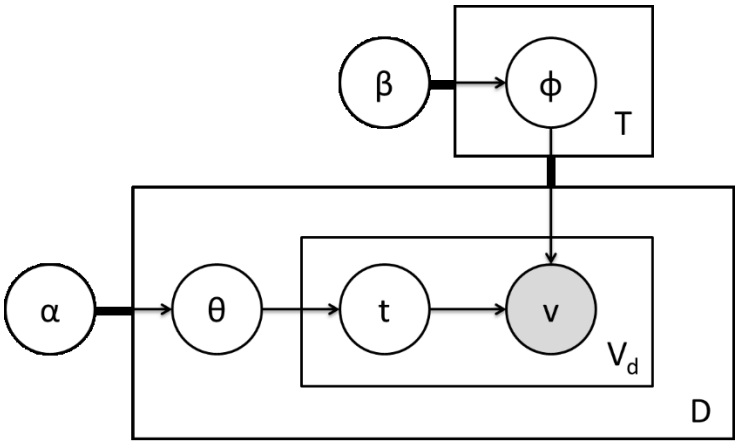


Figure 3. Graphical Model for Latent Dirichlet Allocation.

Figure 3 shows the graphical model representation of LDA. For each document in the corpus of D documents the multinomial topic distribution Θ is sampled from a corpus-wide Dirichlet distribution with hyper-parameter α (Θ represents the density of topics over the document). To produce each of the V_d technical terms in the document, three steps are taken: a topic t is selected by discretely sampling from Θ; the multinomial word distribution Φ is sampled from a corpus-wide Dirichlet distribution with hyper-parameter β (Φ represents the density of technical terms over the T topics); and finally the technical term v is selected from the universe of V technical terms by discretely sampling from Φ conditioned on topic t.

The task of calculating the distributions Θ and Φ exactly is computationally intractable. The mathematical derivation of LDA and the use of Gibbs sampling to approximate the distributions Θ and Φ are presented in detail in Section 4.3.1.

The success of LDA has made it almost synonymous with topic modelling in NLP. LDA has been used in a variety of NLP tasks such as document summarisation (Wang et al. 2009a), social network analysis (Zhao et al. 2011) and part-of-speech tagging (Toutanova & Johnson 2007). However, the LDA model is characterised by several free and seemingly arbitrary parameters: the nature of the priors and choice of the hyper-parameters α and β; the choice of method for approximating the distributions Θ and Φ; the termination criteria of Gibbs sampling; and even the number of topics to choose. From a theoretical standpoint, much research has gone into finding optimal choices for these criteria (Asuncion et al. 2009; Wallach et al. 2009a), leading to localised improvements that are by no means universal; for instance, all published improvements are domain specific. However, from a practical standpoint, LDA seems rather robust to changes in these parameters, and so they are largely ignored in the literature.

To dispel doubts about the apparent arbitrariness of LDA, this thesis not only explores different LDA parameters, but also examines what happens if a different topic modelling approach, Non-negative Matrix Factorisation, is used instead of LDA (see the experiments in Section 5.1.2).

2.2.3 Non-Negative Matrix Factorisation (NMF)

Non-negative matrix factorisation offers an approach to topic modelling that requires only two arbitrary parameters: the number of topics, and the choice of update algorithm. It is a generic matrix factorisation technique originally from the field of linear algebra (Lee & Seung 2001). A matrix X is factorised into two non-negative matrices W and H such that

$$X_{D\times V} \approx W_{D\times T} H_{T\times V}$$

The mapping from the W and H matrices of NMF to the topic modelling representation in Figure 1 is trivial. When NMF is applied to document-bag-of-technical-terms counts, the rank T of matrices W and H corresponds to the number of topics in the model. The choice of T is dependent on the corpus being modelled. Large values of T allow the product of W and H to reproduce the original matrix X more accurately, at the expense of increased computation time and storage space and decreased ability to generalise over the topical structure contained in the corpus. Smaller values of T produce matrices W and H that better generalise matrix X, but with increasing error $F(W, H) = X - WH$. If T is too small, NMF overgeneralises X, leading to topics that are too general to discriminate documents.

Lee & Seung (2001) popularised NMF by presenting two iterative algorithms for generating matrices W and H, based either on minimising the Frobenius norm (least square error) or on minimising the Kullback-Leibler (KL) divergence. A detailed exposition of these algorithms is given in Section 4.3.2.

It has been shown that NMF based on KL-divergence is closely related to PLSA and produces sparser representations in both the H and W matrices (Gaussier & Goutte 2005). Additionally, Van de Cruys et al. (2011) describe how the update rule based on KL-divergence is better suited to modelling text because it can better model Zipfian distributions. Pauca et al. (2004) and Shahnaz et al. (2006) apply topic modelling using NMF to a variety of text collections and find that variants of NMF that impose statistical sparsity (such as those that minimise KL divergence) produce more specific topics represented by the W matrix, and perform better on various NLP tasks. Similarly, Xu et al. (2003) show that the sparser topical clusters produced by NMF surpass LSI both in interpretability and in clustering accuracy.

2.2.4 Advanced Topic Modelling

A shortcoming of LDA and NMF is that they use only the lexical information inside the documents without considering interactions between documents. It is reasonable to suppose that the topics in a corpus are also influenced by various document metadata, such as their authors, their publication dates, and how they cite each other.

There are numerous extensions to LDA that incorporate external information in addition to the lexical information contained in the documents' text. Steyvers et al. (2004), Rosen-

Zvi et al. (2004) and Rosen-Zvi et al. (2010) model authors and documents simultaneously to build author-topic models, which are useful in computing the similarity of authors and in finding relationships between them. Tang et al. (2008a) model authors, publication venues and documents simultaneously to improve academic paper search. McCallum et al. (2005) and McCallum et al. (2007) model the senders, recipients and topics in email corpora for the purpose of automatic role discovery inside organisations. Ramage et al. (2009) generalise these ideas by modelling document text alongside document labels. The labels are crowd-sourced tags of web pages in their application, but they could also be author names, publication venues or university names.

Another shortcoming is that LDA and NMF assume independence between topics and are therefore unable to model correlation between the topics they generate.

Attempts to model the correlation between topics have produced several advancements to LDA. Instead of using an underlying Dirichlet distribution, Blei et al. (2004) model the correlation between topics using an underlying Nested Chinese Restaurant process, Li & McCallum (2006) use multiple Dirichlet distributions, and Blei & Lafferty (2006) and Blei & Lafferty (2007) use a correlated logistic normal distribution. Although these advanced models claim to model textual corpora better than LDA, their claims are only based on measures of perplexity and have not been evaluated using real-world applications. Chang et al. (2009a) show that topic models with better perplexity scores are not necessarily better when judged by human evaluators: they find that LDA and PLSI produce topics that are more understandable by humans than the Correlated Topic Model of Blei & Lafferty (2006). Furthermore, the hierarchical structures of these advanced models make their results difficult to apply to NLP tasks compared to simple LDA.

As will be seen, the topic models leveraged in this thesis represent documents using a bag-of-technical-terms representation rather than a bag-of-words. Wallach (2006) explores aspects of the same idea by treating the underlying documents as bags of words and bigrams. An advantage of her approach is that bigram technical terms are discovered as part of the topic modelling process. However, her model is limited to technical terms that are bigrams, without any scalable extension to longer technical terms. But longer technical terms are empirically better: Wang & McCallum (2005) and Wang et al. (2007) approach the discovery of technical terms within topics by simultaneously inferring topics and agglomerating bigrams. They report that topics described by n-grams are more interpretable by their human subjects than those described by unigrams alone.

Other extensions to LDA relevant here are those that incorporate the citation structure within a document corpus. Before exploring them, it is instructive to first study the literature around this citation structure and how it can be leveraged to provide a measure of authority in a document corpus. Section 2.3.6 continues the discussion of advanced topic models that incorporate citation structure.

2.3 Models of Authority

The previous section covered a variety of relationships that can be discovered in a corpus of papers using lexical information. Strohman et al. (2007) point out that lexical features alone are poor at establishing the authority of documents, so we now turn to the relationships between scientific papers that arise through citation behaviour. As it turns out, these relationships are of a different and often complementary nature. Together they play an important role for automatically recommending reading lists by modelling how individual papers are related to each other and to the desired field of the reading list.

In particular, the citation-based information provides us with a way of distinguishing between papers with different levels of authority, quality or significance. Measures of authority, quality or significance are subjective and so in this thesis I do not presume to pass judgement on scientific papers. Instead, I avoid this subjectivity using the same construct as does Kleinberg (1999), using his notion of "conferred authority." The choice of one author to include a hyperlink or citation to the work of another is an implicit judgement that the target work has some significance. In this thesis, the authority of a paper is a measure of how important that paper is to a community who confer that authority. This very definition of authority suggests that authoritative papers are likely candidates for inclusion in recommended reading lists.

Incidentally, a wide variety of measurements of authority in scientific literature and on the web use the citation graph between papers and web pages. This will be discussed in the upcoming sections.

2.3.1 Citation Indexes

The simplest relationship between two scientific papers is the existence of a citation link, where the author(s) of one paper chooses to cite the other paper because it has some significance to her. A set of papers *D* and the entirety of the citations between them *E* forms a citation graph *C(D,E)*. Garfield (1964) built the first large-scale citation graph for scientific papers. Today, much larger citation graphs of the global pool of millions of published papers are available, e.g., from CiteSeer[1] or Google Scholar[2], or for more focussed pools of papers, e.g. the ACL Anthology Network[3]. Unfortunately, these citations graphs are far from complete (Ritchie 2009) because the automatic discovery of citations in such large corpora is a difficult and unsolved task (Giles et al. 1998; Councill et al. 2008; Chen et al. 2008).

There are a variety of relationships that can be read off a citation graph, even if it is only partially complete. The field of bibliometrics investigates the usefulness of a variety of

[1] http://citeseerx.ist.psu.edu
[2] http://scholar.google.com
[3] http://clair.si.umich.edu/clair/anthology/index.cgi

these relationships and how they can be applied to such tasks as ranking the importance of journals or measuring the academic output of researchers.

Bibliographic Coupling (Kessler 1963) measures the number of citations two papers have in common:

$$BibliographicCoupling_{d1,d2} = \frac{|c_{\leftarrow d1} \cap c_{\leftarrow d2}|}{|c_{\leftarrow d1} \cup c_{\leftarrow d2}|}$$

where $c_{\leftarrow d1}$ and $c_{\leftarrow d2}$ are the sets of papers cited by documents *d1* and *d2*, respectively. The rationale behind this score is that pairs of papers with a high Bibliographic Coupling value are likely to be similar because they cite similar literature.

Co-citation Analysis (Garfield 1972; Small 1973) measures the number of times two papers have been cited together in other papers:

$$Cocitation_{d1,d2} = \frac{|c_{\rightarrow d1} \cap c_{\rightarrow d2}|}{|c_{\rightarrow d1} \cup c_{\rightarrow d2}|}$$

where $c_{\rightarrow d1}$ and $c_{\rightarrow d2}$ are the sets of papers that cite documents *d1* and *d2*, respectively. The rationale behind this score is that pairs of papers with a high co-citation value are likely to be similar because they are cited by similar literature.

Relative Co-cited Probability and Citation Substitution Coefficient, the new paper similarity metrics I will introduce in Section 3.2, make use of these constructs to measure the degree of substitutability of one paper with another.

It is possible to create graphs based on social relationships other than the citation graph. These graphs are typically similar to citation graphs in form and spirit. For instance, papers written by the same author should be related because they draw from the same pool of personal knowledge. Papers written by researchers who have co-authored in the past should be related because the authors have had shared interests, experiences and resources. Liben-Nowell & Kleinberg (2007) study co-authorship networks to predict future co-authorships. Papers published in the same journal are also likely to be related as they are selected for inclusion in the journal for an audience with specialised interests. Klavans & Boyack (2006) investigate a wide variety of the relationships between scientific journals and papers. Several of these relationships have proved useful when combined as features in machine learning algorithms: Bethard & Jurafsky (2010) find several graph-based relationships that are strong indicators of relationships between papers when it comes to predicting citation behaviour.

The above works were mainly concerned with using citation indexes to assess similarity between papers. But another use of citation indexes is to predict authority among the papers in a document corpus, which we turn to next.

2.3.2 Bibliometrics: Impact Factor, Citation Count and H-Index

The first published systematic measure of authority for scientific literature is Impact Factor (Garfield 1955), which measures the authority of a journal. The Impact Factor of a journal in year *Y* is the average number of citations each paper published in that journal in years *Y-1* and *Y-2* received during year *Y*. The Science Citation Index (Garfield 1964) publishes annual Impact Factors for thousands of journals. Although Impact Factor measures authority at the level of the journal, papers published in journals with a high Impact Factor are considered to have more authority than those published in journals with low Impact Factors. Although Impact Factor is still in use today, there is criticism that it might bias certain fields or be manipulated by unscrupulous publishers (Garfield 2006).

Impact Factor offers a measurement of the *current* authority of a journal – only citations to recently published papers are included in the measure. As time progresses and citation patterns change, earlier published papers become less well represented by the Impact Factor of their journal, both at the time they were published (because the current Impact Factor is meaningless at the time the paper was published) and in the present (because the historical Impact Factor is meaningless in the present). Another metric, citation count, is used as a proxy for the overall authority of an individual paper (de Solla Price 1986). The authority of a paper is correlated to the number of citations it has received over its lifetime. However, citation count also has some shortcomings. One example is the reliability of citations (MacRoberts & MacRoberts 1996), where citation counts can be influenced by citations that are biased (either consciously or unconsciously). Another is the difficulties that arise in comparing citation counts across discipline and over time (Hirsch 2005).

With the increasing availability of complete publication databases, H-Index (Hirsch 2005) was developed to mitigate the shortcomings of Impact Factor and citation count. It measures the authority of an author by simultaneously combining the number of publications of the author and the number of times those publications have been cited. An author has an H-Index score of *H* if she has received at least *H* citations for each of at least *H* publications. Although H-Index measures the authority of an author rather than that of a paper, it can be used as a proxy for the authority of each of her papers. This method has the advantage of providing an authority estimate for papers that are systematically outside the scope of other authority measures, in particular newer papers that have not yet received many citations.

2.3.3 PageRank

In the discussion so far, Impact Factor, citation count and H-Index have made only superficial use of the citation graph between papers by counting or averaging the numbers of citations to scientific papers. The PageRank algorithm (Page et al. 1998) goes a step further by attributing authority using properties of the entire citation graph, not just local neighbourhoods of the citation graph.

In its original Web context, PageRank forms the basis of the successful Google search engine (Brin & Page 1998) by rating the importance of web pages using the hyperlink graph between them. Higher PageRank scores are assigned to pages that are hyperlinked frequently, and that are hyperlinked by other pages with high PageRank scores. Using PageRank, the collective importance of a web page emerges through the network of hyperlinks it receives, under the assumption that authors of web pages hyperlink only to other web pages that are important to them.

PageRank works well for discovering important web pages, so can it be applied to science? There are some structural similarities between web pages and their hyperlinks and scientific papers and their citations. In both contexts there is a citation graph, where a link exists because one web page or paper bears some significance to another. Indeed, Chen et al. (2007) and Ma et al. (2008) report some success using PageRank to find authoritative papers in scientific literature. Both papers find a high correlation between PageRank scores and citation counts and report additionally that PageRank reveals good papers (in the opinion of the authors) that have low citation counts.

However, there are also structural differences between web pages and scientific papers. While an author of a web page is able to publish a web site with hyperlinks almost indiscriminately, a scientist has to earn their right to publish and cite. While a web page can have an unlimited number of hyperlinks, space in a scientific bibliography is limited, so for a scientific author there is a logical cost associated with citing another paper. Maslov & Redner (2008) give two reasons as to why PageRank should not be applied to the evaluation of scientific literature without modification: the average number of citations made by each paper varies widely across the disciplines; and PageRank does not accommodate for the fact that citations are permanent once published, while hyperlinks can be altered over time. Walker et al. (2007) argue that PageRank does not adequately take into account an important bias of time effects towards older papers. Their algorithm accounts for the ageing characteristics of citation networks by modifying the bias probabilities of PageRank exponentially with age, favouring more recent publications. Finally, Bethard & Jurafsky (2010) (described in more detail in Section 2.4.6) find that PageRank contributes little better than citation counts when used as a feature in their SVM model for discriminating relevant papers.

Certainly, there is ample evidence that PageRank should be useful in the recommendation of scientific literature. However, there is no clear agreement as to how best to apply it.

The PageRank algorithm is a process that iteratively allocates PageRank through the citation graph using bias and transition probabilities. The bias probabilities represent the likelihood of a reader randomly choosing a paper from the entire corpus. The reader might have some bias as to which paper they generally choose. These bias probabilities are expressed as a d-dimensional vector. The transition probabilities represent the likelihood of a reader following a citation from one paper to another. Again, the reader might have some bias as to which citation they generally prefer to follow. The transition probabilities are comprised of a $d \times d$ matrix.

For each paper *d* in a corpus of papers *D,* PageRank is calculated iteratively as

$$PR(d,k+1) = \alpha \times Bias(d) + (1-\alpha) \times \sum_{d' \in link_{in}(d)} Transition(d,d') \times PR(d',k)$$

where *PR(d,k)* is the PageRank for paper *d* at iteration *k*; *Bias(d)* is the probability that paper *d* is picked randomly from a corpus; *Transition(d,d')* is the transition probability of following a citation from paper *d'* to paper *d*; $link_{in}(d)$ is the set of all papers that cite paper *d*; and α weights the relative importance of the bias probabilities to the transition probabilities[4].

In the original PageRank algorithm,

$$Bias(d) = \frac{1}{|D|}$$

$$Transition(d,d') = \frac{1}{|link_{out}(d')|}$$

where $|D|$ is the number of pages in the corpus; $link_{out}(d)$ is the set of all pages hyperlinked by page *d*; $\alpha=0.15$.

In the context of the web, Brin & Page (1998) choose $\alpha=0.15$ because it corresponds to a web surfer following six hyperlinks on average before jumping to a random new web page ($1/7 \approx 0.15$). Chen et al. (2007) use PageRank to model the scientific literature. They recommend bias weight $\alpha=0.5$ on the basis that, on average, researchers follow just a single citation before starting with a new paper. Ma et al. (2008) agree with this analysis but indicate that this change in α has only a minor effect on the resulting PageRank scores.

Both Chen et al. (2007) and Ma et al. (2008) find a strong correlation between the PageRank score of a paper and the number of citations it receives, so PageRank is not a complete departure from citation count when measuring authority. However, in experiments they anecdotally notice that the PageRank algorithm successfully uncovers some scientific papers that they feel are important despite having a surprisingly low citation count.

2.3.4 Personalised PageRank

The PageRank algorithm produces a global ordering of the authority of connected resources in a network. The notion of global authority works well in the context of the Internet, where companies and websites compete for the attention of a generic Internet

[4] In this section and the next I have reformulated each published variation of the original PageRank algorithm to use consistent notation so that each algorithm can be compared directly by inspecting only the formulae for the bias and transition probabilities. This reformulation sometimes departs from the notations used in the original papers.

user. However, PageRank does not cater for the highly specialised situation we encounter in science, where a web page or scientific work might be authoritative to a small group of specialists that are interested in a particular topic.

There are a variety of modifications to the PageRank algorithm in the literature that attempt to "personalise" PageRank so that it can cater to highly specialised situations.

Continuing the notation from the previous section, a dimension *t* is added to the PageRank score and both the bias and transition probabilities. This dimension represents the personalisation aspect of PageRank. The iterative calculation of the Personalised PageRank for each personalisation *t* then becomes

$$PR(t,d,k+1) = \alpha \times Bias(t,d) + (1-\alpha) \times \sum_{d' \in link_{in}(d)} Transition(t,d,d') \times PR(t,d',k)$$

2.3.4.1 Altering only Bias Probabilities

Page et al. (1998) also talk about Personalized PageRank in their paper. They describe it as a means to combat malicious manipulation of PageRank scores by giving more importance in the PageRank calculations to a set of trusted web sites *t*. They alter only the bias probabilities:

$$Bias(t,d) = \begin{cases} 1 / |t|, & if\ d \in t \\ 0 & if\ d \notin t \end{cases}$$

$$Transition(t,d,d') = \frac{1}{|link_{out}(d')|}$$

They also suggest that a different set of trusted websites *t* could be chosen for different purposes. Although they do not explore this idea experimentally, it does foreshadow that personalisation might be used for specialisation.

Richardson & Domingos (2002) attempt to specialise document search by personalising PageRank on-the-fly at query time. For query *q* with corresponding topic *t*=*q*,

$$Bias(t,d) = Bias(q,d) = P_q(d)$$

$$Transition(t,d,d') = Transition(q,d,d') = P_q(d') \times P_q(d' \to d)$$

where

$$P_q(d) = \frac{R_q(d)}{\sum_{d'' \in D} R_q(d'')}$$

$$P_q(d' \to d) = \frac{R_q(d)}{\sum_{d'' \in link(d'_{out})} R_q(d'')}$$

$R_q(d)$ is the relevance of document d to query q (e.g. calculated using TF-IDF). $P_q(d)$ is the global probability of document d given query q. $P_q(d' \to d)$ is the probability of reaching page d from page d' given query q. This is the first time that both the Bias and Transition probabilities are adjusted to personalise PageRank towards a particular search need. However, the authors advise that this algorithm has space and time computational requirements up to one hundred times that of PageRank.

Haveliwala (2003) calculates a Personalised PageRank for each of a set of manually created topics. This avoids the computational scalability problem of Richardson & Domingos (2002) because he considers a fixed collection of 16 topics corresponding to the web pages in the top 16 categories of the OpenDirectory Project (ODP). These ODP categories are themselves manually created. For each topic t comprised of several documents, he creates a different Personalised PageRank ordering $P(t,d)$ by altering only the Bias term:

$$Bias(t,d) = \begin{cases} 1 / |t|, & if\ d \in t \\ 0 & if\ d \notin t \end{cases}$$

He evaluated the performance of his algorithm by a user study. The study concludes that for web pages in niche areas of interest, his Personalised PageRank can produce more accurate PageRank scores than PageRank alone.

However, there are several drawbacks to Haveliwala's method. The obvious drawback is that his topics are defined by manually curated lists of authoritative web pages. Any manual step in search algorithms of this kind are always unattractive for a variety of reasons: an expert must spend time creating lists of pages that represent each topic; the pages will eventually become out-of-date and so the expert must refresh them periodically; and the topics are domain dependent and so the work must be done again for new domains. The second drawback is that it is not clear how these topics are combined at query time: a searcher needs to know beforehand which personalised PageRank best suits his query. Finally, Haveliwala's assumption is that a researcher's interests are exactly expressed in a single topic, but what one looks for in science is typically a mixture of topics.

Jeh & Widom (2003) describe how to linearly combine several personalised PageRank scores. They show that Personalised PageRank scores (which they call basis vectors) can be linearly combined using the searcher's topic preferences as weights. Under this interpretation, Brin and Page's original PageRank is a special case of Personalised

PageRank when the Personalised PageRank scores are combined with equal weight. They also show that personalised PageRank algorithms have similar convergence properties as the original PageRank algorithm.

Haveliwala's method of personalising by altering the bias probabilities with manually created topics has been used for different domains and tasks.

Wu et al. (2006) perform spam detection by manually choosing several hand-made directories of web pages to use as initial biases.

Gori & Pucci (2006) perform personalised scientific paper recommendations. They crawl the search results of the ACM Portal Digital Library web site to collect 2,000 papers for each of nine manually selected topics. Using a subset of the papers from each topic as the bias set, they test the performance of their algorithm by evaluating how many of their top recommendations appear in the list of papers for that topic.

Agirre & Soroa (2009) use this form of Personalised PageRank to perform Word Sense Disambiguation. They run PageRank over the WordNet (Miller 1995) sense graph after modifying the bias probabilities to favour word senses that exist in their input text. The final PageRank scores identify the word senses that are most likely candidates for disambiguation.

2.3.4.2 Altering only Transition Probabilities

Narayan et al. (2003) and Pal & Narayan (2005) accomplish personalisation by focussing on the transition probabilities instead of the bias probabilities. They define topics as the bags of words assembled from the text inside the pages from each category in the ODP. This differs from Haveliwala's experiment in that the topics of Haveliwala are lists of pages from each category in the ODP. Their method still requires manually curated categories of web pages to make up their topics. Under their model, *Transition(t,d)* is proportional to the number of words in document *d* that are strongly present in the documents contained in topic *t*.

2.3.4.3 Altering both Bias and Transition Probabilities

Nie et al. (2006) produce a more computationally scalable version of the ideas presented in Pal & Narayan (2005) by associating a context vector with each document. They use 12 ODP categories as the basis for learning these context vectors. Using a naive Bayes classifier, they alter both the bias and transition probabilities to take into account the context vector associated with each page as follows:

$$Bias(t,d) = \frac{1}{|D|} C_t(d)$$

$$Transition(t,d,d') = \gamma \times Trans_{same\_topic}(t,d,d') + (1-\gamma) \times Trans_{other\_topic}(t,d,d')$$

$$Trans_{same\_topic}(t,d,d') = \frac{1}{|link_{out}(d')|}$$

$$Trans_{other\_topic}(t,d,d') = \sum_{t' \neq t} \frac{C_{t'}(d')}{|link_{out}(d')|}$$

where $C_t(d)$ is the context vector score for topic t associated with document d; $Trans_{same\_topic}(t,d)$ is the probability of arriving at page d from other pages in the same context; $Trans_{other\_topic}(t,d)$ is the probability of arriving at page d from other pages in a different context; and γ is a factor that weights the influence of same-topic jumps over other-topic jumps. Their results suggest that γ should be close to 1, indicating that distributing PageRank within topics generates better Personalised PageRank scores.

Although they rely on manually compiled ODP categories, they suggest as a future research direction the potential for abstract topic distributions, like those formed as a result of dimension reduction, to automatically determine their categories. It is one of the technical contributions of this thesis to take up this suggestion and connect topic modelling with Personalised PageRank (as is described in Chapter 3).

2.3.4.4 Personalisation by Automatically Generated Topics

While the Personalised PageRank variants described up to now require manual descriptions of topics, Yang et al. (2009) use LDA to automatically discover abstract topic distributions in a corpus of scientific papers. They alter both the bias and transition probabilities as follows:

$$Bias(t,d) = \frac{1}{|D|} P(t|d)$$

$$Transition(t,d,d') = \gamma \times Trans_{same\_topic}(t,d,d') + (1-\gamma) \times Trans_{other\_topic}(t,d,d')$$

$$Trans_{same\_topic}(t,d,d') = P(t|d,d') \cong \frac{1}{|link_{out}(d')|}$$

$$Trans_{other\_topic}(t,d,d') = \frac{1}{T}\sum_{t' \neq t} P(d,t|d',t') \cong \frac{1}{T}\sum_{t' \neq t} P(t|d)P(t'|d')$$

where T is the number of LDA topics; and $P(t|d)$ is a probability of topic t given document d, which can be read directly from the generated LDA probabilities. Like Nie et al. (2006), they achieve best results with $\gamma=1$, so their model reduces to

$$Bias(t,d) = \frac{1}{|D|}P(t|d)$$

$$Transition(t,d,d') = \frac{1}{|link_{out}(d')|}$$

Their decision that $P(d|d',t)$ does not involve any of the LDA topic distributions is surprising: their Personalised PageRank model ultimately uses the LDA topic probabilities to alter only the bias probabilities.

The model of Ding (2011) is similar to that of Yang et al. (2009), except that they use ACT (Tang et al. 2008a) instead of LDA to generate the topics. ACT is an extension of LDA that jointly models the authors and conferences of papers alongside their bag-of-words representations to generate the latent topics.

Under the models of both Yang et al. (2009) and Ding (2011), when the reader randomly jumps to a new paper, they will tend to favour papers that are closely associated with the topic. However, when they follow a citation link, they will arbitrarily pick any one of the citations with equal probability. Instead it seems intuitive that one should favour citations that are closely associated with the topic. The algorithm presented as a contribution of this thesis does exactly this, as will be shown in Chapter 3. The results of the experiments in Sections 5.2 and 5.3 bear out this intuition.

2.3.5 HITS

While this thesis primarily focuses on a variant of Personalised PageRank to produce measures of authority, it would be inappropriate not to briefly consider HITS (Kleinberg 1999), whose original title reads, "Authoritative Sources in a Hyperlinked Environment."

In the HITS model, there are two classes into which noteworthy pages can fall: hubs, which are important because they point to good authorities; and authorities, which are important because they are pointed to by many good hubs. These categories make sense in the real-world where we have authorities like Microsoft, which is an expert about the

Windows operating system in particular, while we have hubs like the Wikipedia operating system page, which is an expert at listing all the available operating systems in general.

The cyclical definitions of hub and authority are mutually reinforcing, and an iterative algorithm converges to a steady distribution. A set of hub scores *H* and a set of authority scores *A* are generated iteratively for a set of pages using two update steps for each page *d*:

$$H(d, k+1) = \sum_{d' \in link_{out}(d)} A(d', k)$$

$$A(d, k+1) = \sum_{d' \in link_{in}(d)} H(d', k)$$

Iteration continues until *H* and *A*, with appropriate renormalisation, converge to a steady state. The pages with the highest *H* scores are the hubs, and the pages with the highest *A* scores are the authorities (the goal of the algorithm is to present these pages).

Whereas PageRank only considers outbound links, HITS iterates between outbound and inbound links, making convergence slow for large datasets. It is most effective for evaluating authorities on a local neighbourhood of a dataset – perhaps the set of pages found as the result of a query (Bharat & Mihaila 2001). While HITS may work well in the context of the Internet, it is not clear exactly how it would be applied to scientific literature. One might hypothesise that literature reviews could act as hubs. However, there won't be a review of every topic imaginable, and Lempel & Moran (2000) found that a lack of reliable hubs can cause HITS to find authorities only for certain self-reinforcing regions of the corpus.

PHITS (Cohn & Chang 2000) places HITS on a more principled statistical foundation by providing probabilistic estimates of authority that have clear semantics. They apply PHITS to science with the task of identifying citation communities. SALSA (Lempel & Moran 2000) provides a more efficient algorithm for calculating hubs and authorities. Nie et al. (2006) describe a personalised version of the HITS algorithm using LDA.

2.3.6 Combining Topics and Authority

As will be seen, the synthesis of topic modelling with Personalised PageRank plays an important role in this thesis. Topic modelling is used to find the lexical relationships between papers, specifically with the intention of partitioning a corpus into several cohesive groups. Personalised PageRank is then used to locate the authoritative papers within each group. In contrast to the methods in Section 2.3.4 that use topics to modify Personalised PageRank in a relatively pipelined fashion, there are several other approaches to combining the ideas of topics and authority.

Mann et al. (2006) explore various combinations of citation- and topic-based relationships to define a variety of new bibliometrics measures such as topical impact factor, topical precedence and topical longevity. They then describe where each of these measures might be used to improve some of the known shortcomings in the field of bibliometrics.

Bharat & Henzinger (1998) build a topic-specific graph of hyperlinked documents by using the results of a search engine to form the topic for a particular query. They then use a modified version of HITS to locate the authorities on this graph. This technique is relevant in the context of this thesis if one interprets the results of a search query as the basis of a "topic".

Cohn & Hofmann (2001) simultaneously model citations and terms using probabilistic factor decomposition. While they do not specifically model topics or authority, the factors that make up their decomposition can be interpreted as topics with authority: each factor has associated with it a set of most likely terms (interpretable as the topic description) and most likely citations (interpretable as the authorities).

Erosheva et al. (2004) model a corpus using a multinomial distribution simultaneously over a bag-of-citations and bag-of-words representation of each document. They automatically discover latent "aspects" inside a document corpus, each of which is associated with a list of the most likely words (interpretable as topics) and citations (interpretable as authorities) in that aspect. These citations can be loosely interpreted as the authorities for that aspect because they appear relatively more frequently than other citations in the aspect in particular, and in the corpus in general. This measure is local, like citation count, because it does not make use of the wider citation graph in the way that PageRank does. A limitation of their algorithm is that it supports only eight or ten aspects. This reduces the usefulness of the algorithm at determining authoritative papers for arbitrary topics.

While Erosheva et al. use just the topic probabilities associated with a citation to influence a citing document, Nallapati & Cohen (2008) and Sun et al. (2008) follow the citation to allow the topics of citing documents to influence the topics of the cited document. They measure the influences that blog posts have on other blog posts that link to them. Gruber et al. (2008) also extend the model of Erosheva et al. They predict the presence of hyperlinks between web pages. Their model treats the absence of a citation between two papers as evidence that the papers have differing topics.

Chang & Blei (2009) extend the model of Nallapati & Cohen (2008) but force their model to use the same latent variables to describe the citations and the terms. They argue that earlier models perform less well because each of their latent variables model either the terms or the citations in a corpus. Unlike Gruber et al., they do not treat the absence of a citation between two papers as evidence that papers have differing topics because on-topic papers are often omitted for space reasons.

Kataria et al. (2010) additionally model each citation with a bag-of-terms citation context extracted from around the citation in the citing document.

Dietz et al. (2007) extend LDA in two different ways to incorporate citations. In their two models, the topics of cited documents are influenced by the topics of the documents that cite them. 22 papers were annotated by their authors to indicate the citations that were most influential to that paper. Their evaluation found that the LDA models that incorporate citations were significantly better at categorising papers' citations as being influential or not than LDA alone. While their model performs well at classifying the influence of the citations for a given paper, it works only for papers and citations that were present during the learning stage. It offers no mechanism for predicting influential citations for topics in general, or for combinations of topics.

All these methods that simultaneously model the citations and terms in a corpus report incremental improvements in automatically discovering topics and influential papers inside those topics. However, there are some drawbacks to tightly coupling topic discovery with citation modelling.

Firstly, all the authorities in a corpus might not be located because papers that are authoritative across several topics will be penalised. Their associated topic probabilities, and hence joint distribution probabilities, will be low because they are divided across several topics.

Secondly, and more disturbingly, these models will not locate topics that lack an authority because the authority component of the joint distribution will be near-zero. This rules out niches in a corpus where several papers are equally relevant, or young niches that do not yet have an established citation network.

Finally, jointly modelling topics and citations requires models with many thousands of factors. Erosheva et al. (2004) report using 39,616 unique words and 77,115 unique citations in their joint model. While the number of words in a language is bounded (Heaps 1978), the number of citations in a corpus grows with the size of the corpus. It therefore makes sense to partition the model where possible. Scalability is never explicitly addressed in any of the papers reviewed in this section, and indeed we can observe that the datasets against which they evaluate are generally small, with number of papers ranging from several hundred to a couple of thousand, number of citations ranging from one- to ten thousand, and number of topics in their models ranging from eight to twenty. Contrastingly, LDA has been shown to scale to corpora of millions (Newman et al. 2006), and PageRank to billions (Page et al. 1998) of documents, but these advantages are only obvious while they remain decoupled. This is the reason why the work in this thesis advocates the pipelined approaches of Section 2.3.4, which separate topic modelling, which is computationally tractable using LDA, from authority modelling, which is cheap using Personalised PageRank.

2.3.7 Expertise Retrieval

A particular area that commonly combines topics and measures of authority is expertise retrieval or expert finding. Here, one finds a list of experts for a given topic, or vice-versa, one finds a list of topics that best describes an expert's area of expertise. According to Balog et al. (2012), these techniques fall into five broad categories: generative probabilistic models; discriminative models; voting models; graph-based models; and other miscellaneous models. In various evaluations, approaches based on topic modelling have been outperformed by a range of other methods.

We don't explore expertise retrieval in this thesis as the task of selecting experts for a field is sufficiently different to generating reading lists. However, one can imagine a tentative approach to generating reading list that first selects the experts that match the desired query, and then selects a variety of the experts' papers as a reading list.

2.4 Generating Reading Lists

In this thesis I address the task of automatically generating reading lists for novices in a scientific field. In particular, I focus on the type of reading list that might currently be manually created by an expert to quickly familiarise novices with the important concepts in their field. This form of reading list is by no means the only possible manifestation of a reading list. A reading list can be regarded as any set of papers recommended for the purpose of guiding a reader through a subset of a corpus of papers. In this section I review a variety of mechanisms that might help a researcher arrive at a set of recommended papers.

2.4.1 Ad-hoc Retrieval

The simplest incarnation of a reading list is the ad-hoc retrieval of papers using IR methods like those described in Section 2.1. To generate this kind of reading list, a novice submits a search query – in whatever query syntax is available – to an IR system. The papers returned are those in a corpus that, according to the IR system, best match the novice's query. The underlying IR algorithms typically use only the lexical content of each paper and use no domain specific knowledge about the papers, such as the citation graph. The "Search using Google" facility on the ACL Anthology Network website[5] is an example of this kind of mechanism, where the paper content is indexed but the available citation graph is ignored.

Relying only on the lexical information in the papers, improvements to ad-hoc retrieval have been made by incorporating document clustering (Liu & Croft 2004) and topic modelling (Wei & Croft 2006).

Several studies show that adding information from the citation graph to ad-hoc retrieval yields substantial improvement. Meij & De Rijke (2007), Fujii (2007) and Ma et al.

[5] http://aclweb.org/anthology-new

(2008) report improved retrieval performance when citation counts or citation-based-PageRank scores are combined with more traditional term-based searching. Ritchie et al. (2006) reports improved retrieval performance when a paper's text is augmented with text from the contexts in other papers where it has been cited.

Two commercial search systems that provide ad-hoc retrieval for scientific search are Google Scholar[6] and Microsoft Academic Search[7].

2.4.2 Example-based Retrieval

Rather than generating a reading list from a term-based search query, some systems allow a paper or set of papers (a query set) to form the basis of the search query. An advantage of this mechanism is that the query set is often more representative of the novice's search need because it contains more lexical and citation-based information than a single search string. It also allows the system to better take into account the learning interest, knowledge and goals of the novice (Tang 2008). A disadvantage is that the novice has to somehow have acquired the query set in the first place.

Gipp & Beel (2009) use citation proximity analysis and citation order analysis to identify documents related to a query set. Woodruff et al. (2000) generate a reading list for a single paper by "spreading activation" over its text and citation data. El-Arini & Guestrin (2011) retrieve influential papers by modelling their connectivity in the citation graph to a query set.

Both Google Scholar and Microsoft Academic Search provide facilities to query by example.

2.4.3 Identifying Core Papers and Automatically Generating Reviews

There might be instances where the novice neither knows any meaningful search terms with which to initiate exploratory search, nor do they possess a set of core papers to use for example-based search. If they have access to a large corpus of papers that they know are relevant to the area with which they wish to familiarise themselves, there are several systems that locate the core papers inside a corpus of papers.

Cohn & Chang (2000) create customised authority lists using the PHITS algorithm. Chen et al. (2007) use PageRank to model a corpus of papers. Wang et al. (2010) combine the citation graph with statistics about the frequency of download of each paper. They combine the measures "citation approaching" and "download persistence" to recommend classical papers to novices. Radev et al. (2009a) use a variety of measures to identify core papers in a corpus, including citation count, Impact Factor and PageRank scores.

[6] http://scholar.google.com
[7] http://academic.research.microsoft.com

The commercial systems ArnetMiner[8] (Tang et al. 2008b) and Microsoft Academic Search provide mechanisms for identifying core papers in scientific literature.

There are some methods that aim at generating automatic reviews, or at least providing material for a semi-automatic review creation process. Nanba et al. (2004; 2005) use HITS to generate a short-list of papers to review, focussing on the papers that are hubs. Qazvinian & Radev (2008) and Mohammad et al. (2009b) use clustering techniques to locate core areas of a corpus, and then use the contexts around each citation to generate extracts that stand in as survey summaries.

2.4.4 History of Ideas and Complementary Literature

Similar to finding the core papers inside a large corpus of papers, there are several systems that attempt to track the flow of ideas through the literature. A novice might familiarise themselves with a new field by paying particular attention to the papers that contribute significantly towards the flow of ideas relevant to their field, particularly those papers that are considered intellectual turning points in the field.

Chen (2004) prunes the citation graph of a corpus of papers to identify three types of significant papers: turning points, pivot points and hubs. Their notion is that the flow of ideas through the literature can be mapped by the citations connecting these significant papers.

Rather than relying on the citation graph, Shaparenko & Joachims (2007) use only the lexical content of papers to discover the flow of ideas. By representing each document as a distribution over words, they use a Likelihood Ratio Test to determine the probability that a more recent document was influenced by an older document. Where there is evidence that a more recent document is using significantly more words from an older document than is expected, the older document is said to have influenced the more recent document. They build up an influence graph of these influences and identify documents that have influenced many other documents. This is a similar measure to citation count on the citation graph.

There might be instances where an experienced researcher changes their focus from one field to another. In this case they are a novice in their new field, but at the same time have a wealth of prior experience to draw from while learning the new field.

The domain of complementary literature attempts to bring a novice up to speed more quickly in a new field by relating foreign concepts in the new field (the target field) to well-known concepts in another field (the source field). This helps the novice more quickly grasp related ideas by leveraging their existing expert knowledge. Swanson & Smalheiser (1997) recommend papers from the target field literature that have many technical terms in common with papers in "intermediate literatures" that in turn have many technical terms in common with papers in the source field literature. This use of

[8] http://arnetminer.org

the co-occurrence of technical terms to elicit relationships between papers is similar in concept to the mechanism employed by co-citation analysis in the field of bibliometrics.

2.4.5 Collaborative Filtering

All the mechanisms for generating reading lists discussed so far involve a novice in isolation, or at most a novice with access to an expert. Collaborative Filtering (CF) (Goldberg et al. 1992) allows a novice to leverage relevance judgements made by groups of other researchers. The idea behind CF is that if two researchers have overlapping interests – in the case of reading lists, if two researchers have read a common set of papers – then the papers judged relevant by only one of them are probably relevant to the other. Furthermore, if an individual and a group of researchers have overlapping interests, then papers judged relevant by the group are probably relevant to the individual. CF attempts to group researchers automatically by analysing their reading habits.

CF has been successfully applied to the scientific literature. McNee et al. (2002) use CF to recommend additional citations for research papers using the existing citations in the paper as the basis for CF recommendation. Torres et al. (2004), Bogers & Van Den Bosch (2008) and Wang & Blei (2011) recommend scientific papers to researchers using the papers the researcher has already read as the basis for CF recommendation.

Ekstrand et al. (2010) apply CF to the task of automatically generating reading lists. Their system assumes that a novice already has a small set of papers from their field of interest. This set of papers is combined with a variety of lexical similarity measures and authority measures to form the basis for CF recommendation.

There are three disadvantages to using CF for the purpose of generating reading lists for a novice. Firstly, the novice has to have chosen a few papers to use as a basis for the CF recommendation, and some systems even require that the novice attach a relevance score to each of these papers. But by definition, a novice lacks the experience to choose these papers, let alone rate them. Secondly, CF requires that a large number of other researchers have read (and possibly rated) these papers and papers relevant to them. These people simply might not exist for a small topic at hand. Thirdly, CF suffers from a "cold-start phenomenon," where recommendations are generally poor where data is sparse. A novice has to have read (and possibly rated) many papers before they receive meaningful recommendations, and a paper has to have been read (and possibly rated) by many researchers before it can be meaningfully recommended. To alleviate the cold-start phenomenon, Zhou et al. (2008) use information beyond the citation graph to include additional features from the author graph (matching papers to authors) and venue graph (matching papers to publication venues).

Two commercial search systems that rely on CF to provide scientific search are CiteSeerX[9] and Citeulike[10].

2.4.6 Playlist Generation

In the area of music information retrieval, the task of playlist generation closely resembles the task of generating reading lists. Here, a playlist – a list of songs – is provided in response to a particular search need (Bonnin & Jannach 2013). This search need might be given either as some seed information (such as listening history) or a semantic description (such as "joyful songs") (Barrington et al. 2009). The ISMIR 2012 conference (Peeters et al. 2012) dedicates an entire track to the task.

Both types of search need have parallels to reading list recommendation. A listening history can be likened to a list of previously read papers. A semantic description (such as "joyful songs") can be likened to a scientific area of expertise (such as "statistical machine translation"). Where the tasks diverge is that reading list generation can use the textual information in the papers directly, while in general, playlist generation relies on tags that are manually associated with songs (Moore et al. 2012). However, music detection and classification technology such as Shazam (Wang 2006) may in future lead to the ability to use the song content directly.

2.4.7 Reference List Reintroduction

One of the experiments in this thesis, presented in Section 5.3, involves the task of reference list reintroduction (RLR). Given only the abstract or full text of a paper (with citation information redacted) as an indication of a search need, the task is to predict which papers the target papers originally cited. Evaluation by RLR is cheap and allows for large and economical data sets. RLR serves as a proxy evaluation for the task of automatically generated reading lists: the citations of a paper can be viewed as a recommended reading list by the author of a paper (presumably an expert) that provides necessary and sufficient background for the reader (potentially a novice) to understand the paper.

The major downside of evaluation by RLR is that citations are not a perfect gold standard: they can exhibit subjective inclusions and omissions (MacRoberts & MacRoberts 1996), and they often reflect the idiosyncrasies of how scientific communities cite prior work (Shaparenko & Joachims 2007). These problems are mitigated by the fact that all the systems tested are exposed to the same citation idiosyncrasies and that the subjective characteristics are averaged out by the high number of papers in the data set.

Solutions to RLR necessarily draw from the variety of techniques discussed so far in this section. Lexical similarity plays a large role because a paper and the papers it cites are likely to use similar language and terminology. Topic models are useful because papers

[9] http://citeseerx.ist.psu.edu
[10] http://www.citeulike.org

are likely to cite other papers that cover the same topics. The citation graph, and its associated measures of authority, identifies those papers that are most likely to be cited. Collaborative filtering techniques take advantage of social constructs, characterising historic patterns of citation behaviour. This variety of influences on the RLR task is mirrored by the various state-of-the-art approaches employed to perform RLR.

McNee et al. (2002) attempt the simpler task of recommending suitable additional references for a research paper. They evaluate six collaborative filtering algorithms for the task, using the existing citation graph to increase the performance of collaborative filtering where data is sparse. The primary disadvantage of their algorithms is that they require a set of existing references from which to start. They do not use any lexical information from the paper – only the citation graph.

Strohman et al. (2007) approach the task of RLR by searching for citations using only the paper text as a query to their recommendation system. They first use an IR system to select the hundred most similar papers to a query paper, and expand those results to include all the papers they cite. Then they combine several features to generate their recommendations from this pool of candidate papers: text similarity, citation counts and citation coupling, author information, and the citation graph. Their model achieves a mean-average precision (or MAP – see Section 3.2 for details) of 10.16% against a corpus from the Rexa[11] database.

Bethard & Jurafsky (2010) improve on Strohman et al. (2007) using a SVM with 19 features from 6 broad categories: similar terms; cited by others; recency; cited using similar terms; similar topics; and social habits. They achieve a MAP of 27.9% against the ACL Anthology Reference Corpus (Bird et al. 2008). They find that publication age, citation counts, the terms in citation sentences, and the LDA topics of the citing documents contribute most to the success of their model. They also find that PageRank provides little discriminative power. A drawback of their method is the large amount of information that has to be provided to create their SVM features, much of which is of a social nature and in their case is manually curated.

Daud (2008) address the same problem by altering the LDA algorithm to model word-over-topic and topic-over-citation distributions. However, they do not give enough detail about their algorithm, nor do they give quantitative results to compare against.

Tang & Zhang (2009) address the harder task of not only reintroducing the reference list of a paper, but also locating the positions in the text of a paper where those citations should be. They train a two-layer Restricted Boltzman Machine (RBM), which jointly models the topic distributions of papers and their citation relationships, to recognise the most similar papers to the sentence containing a citation. The drawback of this technique is the training data of citation contexts required to train the RBM.

[11] http://www.rexa.info

He et al. (2010) and He et al. (2011) also address the task of recommending citations and their positions in the text of a paper. Given a query paper, they compare millions of previously located sentences containing a citation to a moving window in the query paper. When a sentence matches the moving window closely enough, it is recommended as a citation sentence for the query paper at that window position. The drawback of this technique is the requirement of a database of citation sentences that have been used in the literature before. New papers will not be recommended if they have not been cited by other authors to build a sufficient catalogue of citation sentences.

Lu et al. (2011) tackle the problem of reference list reintroduction by training a translation model to map between citation contexts and their corresponding papers. Again, their model needs to be trained against a database of citation contexts.

2.5 Evaluation Metrics for Evaluating Lists of Papers

As will be described in Section 3.3, one of the contributions of this thesis is a new metric for evaluating reading lists. There are a variety of evaluation measures in the field of NLP for various tasks, none of which exactly meet the requirements for this type of evaluation. This section describes the performance measures from the NLP literature that are relevant to this type of evaluation, and which will be used in the evaluations in Chapter 5.

2.5.1 Precision, Recall and F-score

Given a set of retrieved papers D and a set of relevant papers C, precision (P) and recall (R) are defined as (Van Rijsbergen 1979)

$$P(D,C) = \frac{|C \cap D|}{|D|}$$

$$R(D,C) = \frac{|C \cap D|}{|C|}$$

F_ξ-score combines precision and recall into a single metric (Van Rijsbergen 1975)

$$F_\xi(D,C) = \frac{(\xi^2+1)P(D,C)R(D,C)}{\xi^2 P(D,C) + R(D,C)}$$

where ξ weights the relative importance of precision and recall. Generally, precision and recall are equally balanced with $\xi=1$ to produce the F-score:

$$F(D,C) = \frac{2 \times P(D,C)R(D,C)}{P(D,C) + R(D,C)}$$

Salton (1992) describes how precision-recall measurements, although ubiquitous and easy to compute, are not universally acceptable as performance measures for information retrieval. In particular, recall is incompatible with the utility-theoretic approach to

information retrieval: a document can be relevant to a query but it may not have additional utility in the context of the collection of query results. In one instance, a document might duplicate the information of another relevant document that has already been included in the results. In another instance, a document might be relevant only in the complementary context of another document that has not been included in the results (Robertson 1977).

Precision and recall – and hence F-score – have no mechanism to take into account the fact that documents might be partially relevant to a query. Their implementation classifies a document as either relevant or irrelevant. Reading list recommendation is more sensitive to this binary classification than more generic information retrieval because the resulting sets of recommended papers are inherently shorter than search results, i.e. tens of papers vs. hundreds or thousands or search results. If partial relevance is not captured by the evaluation metric, then comparison between RLR systems becomes difficult as they all have near-zero scores.

2.5.2 Mean Average Precision (MAP)

The precision, recall and F-score of an algorithm are based on unordered sets of papers: they are single-value metrics that consider all retrieved papers equally relevant.

For ranked lists, Average Precision (AP) averages the precision up to each position in the list where a relevant paper is present:

$$AP(D, C) \quad = \frac{1}{|C|} \sum_{i=1}^{|D|} I_{\{D_i \in C\}} \times P(D[1:i], C)$$

where $I_{\{D_i \in C\}} = 1$ if paper D_i is relevant (0 otherwise), and $D[1:i]$ is the list of the top i ranked papers. Note that the denominator of the average is $|C|$, so any missing relevant document is penalised as it contributes 0 towards the AP.

AP is numerically higher for lists where relevant papers are concentrated at the top. AP is an adequate additional metric in situations where one cares about the relative relevance an IR system assigns to its output documents. Buckley & Voorhees (2000) show that AP is more stable and has lower error rates than precision and recall alone. They also show that a difference in AP of 5% between two methods implies a greater than 98% chance that the higher-scoring method is better.

For a set of queries, the Mean Average Precision (MAP) of an algorithm is the average of the AP for each query. For a set of queries Q

$$MAP_Q \quad = \frac{1}{|Q|} \sum_{q \in Q} AP(D_q, C_q)$$

where D_q and C_q are the retrieved papers and the relevant papers for query q, respectively.

Because MAP is based on precision and recall, it suffers the same prejudices as F-score towards utility-theory based information retrieval. It is also unable to take into account partial relevance of the gold-standard set C_q, but as just explained, it can take into account relative relevance in the return set D_q.

In the task of reference list reconstruction, where the number of recommended papers is generally longer than the number of actual references, it is common to use MAP (Strohman et al. 2007; Tang & Zhang 2009; Bethard & Jurafsky 2010; He et al. 2010; Wang et al. 2010).

2.5.3 Relative co-cited probability (RCP)

When performing reference list reintroduction it may happen that some of the papers suggested by the system are actually relevant, although they are not in the bibliography. This may have happened for a variety of reasons such as the author being unaware of the related work or running out of space to include a citation of medium relevance (He et al. 2011). Precision- and recall-based measures will overly penalise any system because they cannot take into account such "close misses." He et al. (2011) introduce relative co-cited probability (RCP), which measures accuracy based on the assumption that papers that are similar in relevance and quality are likely to be cited by the same papers, or co-cited.

Given a corpus of papers D, a query paper d, an expert paper c_j cited by d, and a system-generated test paper r_i recommended for paper d, the RCP of recommendation r_i with respect to an individual cited paper c_i is

$$RCP(r_i, c_j) = \frac{number\ of\ papers\ in\ D\ citing\ both\ r_i\ and\ c_j}{number\ of\ papers\ in\ D\ citing\ c_j}$$

Given the set of papers C cited by D, and the set of recommendations R, the overall RCP for recommendations R with respect to actual citations C is

$$RCP(R, C) = \frac{1}{|R||C|} \sum_{r_i \in R, c_j \in C} RCP(r_i, c_j)$$

Although RCP does attempt to address the prejudice against partial relevance exhibited by precision- and recall-based measures by rewarding recommended papers for being co-cited, the formulation has its own shortcomings. Firstly, RCP will always give zero score to recent papers that have not yet received any citations, regardless of their quality or content. Secondly, RCP overemphasises the importance of partial relevance by allowing a partially relevant paper to contribute to the overall score multiple times if it matches multiple expert papers. By way of example, Figure 4 shows how RCP calculates the same score to these scenarios: (a) 5 retrieved papers that are each 20% co-cited with all of the 5 expert papers; (b) 5 retrieved papers that are each 100% co-cited with one of the 5 expert papers (the same expert paper each time); and (c) 5 retrieved papers that are each

100% co-cited with one of the 5 expert papers (a different expert paper each time). The third scenario is clearly most preferential for the evaluation of a reading list because a perfectly relevant substitute is found for each expert paper.

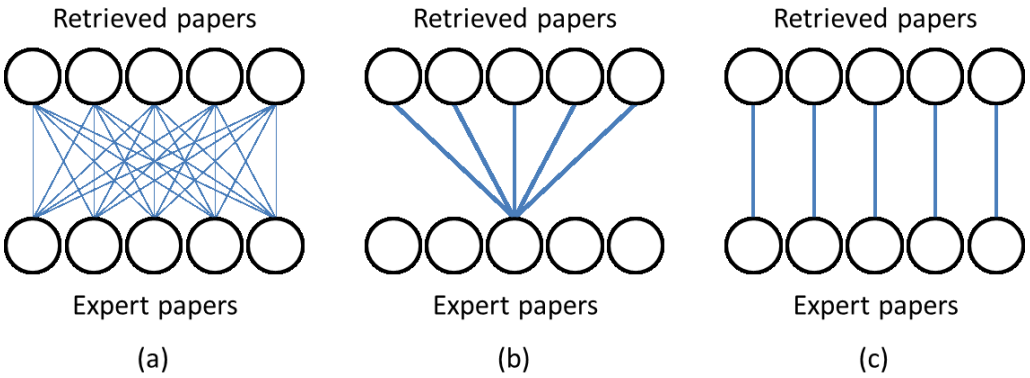


Figure 4. Three Scenarios with Identical RCP Scores.

2.5.4 Diversity

The evaluation metrics described in the previous subsections evaluate a generated list of papers against a gold-standard list of papers. They do this by comparing whether or not each paper in the generated list appears in a gold-standard list. However, an important concept that is missing from these forms of evaluation is that of diversity.

Goffman (1964) was the first to recognise that the relevance of a paper to a generated list of papers must take into account the other papers that appear in the list: papers are not independent in their informational content. Evaluation measures such as MAP judge relevance in isolation, treating the constituent papers of a list as independent.

Bernstein & Zobel (2005) tackle an obvious case of dependence between two papers. They use document fingerprint techniques to locate papers that are content-equivalent – those that have almost identical informational content. Clarke et al. (2008) attempt to define an evaluation framework for IR that accounts for diversity by finding the largest overlap between information properties of documents and probable diverse information needs that are expressed by a query. Zhai et al. (2003) make use of subtopics to evaluate how well the subtopics of an information need are covered by the subtopics contained in the generated list.

As will be seen in the next chapter, Citation Substitution Coefficient, a new paper similarity metric I will introduce in this thesis, implicitly incorporates diversity.

Chapter 3.

Contributions of this Thesis

This thesis is primarily concerned with the task of automatically generating reading lists. We learned from Section 2.4 that there are a variety of existing mechanisms for generating similar artefacts, such as identifying core papers, finding complementary literature, and recommendation using collaborative filtering. However, none of them quite address the task of automatically generating a reading list for a given field. Section 3.1 presents ThemedPageRank (TPR), the system I have built to address this task directly. TPR makes extensive use of technical terms to model a corpus of papers. Section 3.4 presents a light-weight automatic term recognition system built for use in TPR.

Because there are no existing systems for the task of automatically generating reading lists, there are also no existing mechanisms for the evaluation of such systems. Section 3.2 presents the gold-standard reading lists that I collected specifically for the task of evaluating such systems. This gold standard allows future researchers to independently build and evaluate their own systems for the automatic generation of reading lists.

While building and comparing incremental improvements to TPR, I noticed that the traditional IR evaluation metrics, such as F-score and MAP, were not granular enough to allow me to draw significant conclusions about the relative performance of different systems. Systems generally scored similarly to each other without statistically significant differences. Section 3.3 presents the Citation Substitution Coefficient metric, which is designed specifically for the comparison of a list of papers to a gold-standard list of papers. It takes into the account the citation graph to provide partial scores for papers that are close to gold-standard papers in the citation graph.

Finally, Section 3.5 introduces Qiqqa, a research management tool that I built independently alongside my PhD. In this thesis I use Qiqqa to perform two large-scale user satisfaction evaluations in the experiments described in Section 5.5. Qiqqa also provides the ability to visualise a large corpus of documents, which helped direct the design of TPR. These visualisations, and a case-study of how they might be used to perform exploratory reading, are presented in Section 3.5.3.

3.1 ThemedPageRank

The main contribution of this thesis is the creation of ThemedPageRank (TPR), an algorithm for automatically creating reading lists to help bring a novice up to speed in a scientific area previously unknown to them. This contribution addresses the third research question addressed in this thesis: *does lexical and social information contribute towards the task of automatically generating reading lists?*

The automatic generation of reading lists is a relatively recent task. Tang (2008) and Ekstrand et al. (2010) address for the first time the task of automatically generating reading lists for novices, i.e. researchers new to a field. However, both approaches tackle the task using collaborative filtering techniques, which rely on existing ratings of papers in the field – either by the novices themselves or by several of their peers. The requirement of existing manually captured ratings is at odds with the automatic generation of reading lists because, by definition, at the time their need for a reading list arises, a novice has probably not already produced reasonable ratings.

There are two starting points from which reading lists can be created automatically if one does not want to require previously-made judgement. If we start from a keyword-based description of the scientific area covered, the task becomes similar to document retrieval: the returned list should not contain many irrelevant papers, but should contain a wide variety of relevant papers. These two properties correspond to precision and recall as described in Section 2.5.1. But if we start the process of automatically creating reading lists from a paper known to be relevant to the field, the task then becomes similar to reference list reconstruction: the returned list of papers must have sufficient precision to be relevant to the query paper, have broad enough recall to cover all the topics discussed in the paper, and be notable enough to be included in the bibliography of that paper.

TPR supports both starting points by emphasising the role of topics in scientific search. Both the keyword- and paper-based descriptions of scientific areas that novices might use as input queries are converted to distributions over scientific topics. These distribution-based queries have the advantage that they satisfy the range of expressiveness required in scientific search (El-Arini & Guestrin 2011): a search need might be expressed as a single word, a technical term, a phrase or sentence, a paper abstract, a description of the field, a paper in the field, or a small collection of papers in the field. In my approach, each of these can be converted to a distribution over scientific topics. TPR then generates a reading list tailored to this distribution.

3.1.1 Modelling Relationship using Topic Models and Technical Terms

Scientific topics are fundamental to the structure of scientific literature and therefore central to the design of TPR. Scientific literature is intrinsically distributed over topics at a variety of scales: at the macroscopic level, examples are “astronomy” and “computer science”; at the microscopic level, examples are “statistical parsing” and “statistical

machine translation". TPR inherently takes into account the concept of topics at every scale by applying latent topic modelling to a corpus of papers.

Rather than using words as the underlying representation for papers in its topic models, TPR uses technical terms. This choice is driven by three factors: technical terms are important artefacts for formulating knowledge from scientific texts (Ananiadou 1994); descriptions of topics are more understandable if they are expressed in terms of technical terms rather than words (Wallach 2006; Wang et al. 2007); and the number of unique technical terms in scientific text is one to two orders of magnitude smaller than the number of unique words (Wong et al. 2009), which means that TPR scales better to large corpora (Newman et al. 2006).

Firstly, TPR automatically detects all technical terms in a corpus using the algorithm presented in Section 3.4. TPR then builds a $D \times V$ matrix containing the document-bag-of-technical-terms representation of the corpus. D is the number of documents in the corpus and V is the number of technical terms. This is matrix Ω in Figure 1, where the entry at row d, column v represents the number of times technical term v appears in the full text of document d. Topic modelling, using either LDA (Section 2.2.2) or NMF (Section 2.2.3) is then used to collapse matrix Ω into the two smaller, but denser matrices Θ and Φ, where Θ contains the document-topic representation and Φ contains the topic-technical-term representation.

Figure 5 shows the results of applying LDA-based topic modelling with 25 topics on a corpus of around 700 papers and 130 technical terms. This was the set of documents that comprised the reading material for the thesis being described here. Exhibit 1 shows a paper selected from the corpus with the title, "A note on topical n-grams." It is reasonable to assume that the paper combines "topic models" and "n-grams".

The coloured square to the right of the paper detail is a visual representation of the topics that corresponds to this paper, i.e., the paper's row in matrix Θ. Each topic is represented by a different colour, and the width of each colour is proportional to the topic's influence in the document's topic distribution. It is clear that this document is fairly evenly distributed over five topics (five coloured strips), with the "mustard" and "turquoise" topics being most representative. Exhibits 2 and 3 show the technical terms associated most with the mustard and turquoise topics, respectively. The same technical terms can occur in several topics (columns of matrix Φ). The average topic distribution for the set of technical terms is indicated by the colours of their background diagonal swatch. Exhibits 4 and 5 show the eight papers that match each of the technical terms' distributions most closely, using the similarity measure described in Section 4.3.3. Each of those papers has an associated coloured swatch representing their topic distributions.

The "mustard" topic is strongly associated with the automatic extraction of technical terms, both through its associated technical terms and its related papers. Similarly, the "turquoise" topic is strongly associated with topic modelling. Both these topics make

sense, given the title of the sample paper. It is reassuring that the sample paper in Exhibit 1 contains both these topics.

3.1.2 Modelling Authority using Personalised PageRank

The combination of automatic term recognition with topic modelling provides TPR with a scalable approach to discovering the latent topics in a corpus. This representation reveals useful lexical relationships between the topics, the papers and the technical terms in the corpus. However, lexical relationships are just a first step towards automatically building reading lists. While topic modelling alone is able to isolate candidate papers for a reading list based on their topical relevance, it lacks a mechanism for extracting the most authoritative papers from among these candidate papers.

As discussed in Section 2.3.2, measures based on citation counts have been used as proxies for authority in the scientific domain. Unfortunately, citation counts have shortcomings with respect to the reliability of citations and the lack of comparability of citation counts across discipline and over time. Not all citations are equal.

PageRank is a method that formalises the intuition that not all citations are equal. It has met with success in eliciting authoritative sources on the web (Page et al. 1998). However, the direct application of PageRank to the scientific literature has not shown consistent improvement over citation count (Maslov & Redner 2008; Bethard & Jurafsky 2010), a result that is confirmed by my experiments in Sections 5.2 and 5.3.

TPR, the main contribution of this thesis, modifies Personalised PageRank (Haveliwala et al. 2003) in a novel way that: (i) is sensitive to the distribution of topics over the papers in a corpus; and (ii) takes into account the age of papers when determining their authority. This modification addresses two intuitions as to why PageRank does not perform well when applied to scientific literature.

The first intuition about why PageRank does not perform well in the context of scientific literature is that in science, authority is topic-specific. Instead of modelling a global authority with PageRank, TPR models multiple topic-specific authorities by calculating a different Personalised PageRank for each topic it identifies in a corpus. These topic-specific Personalised PageRanks are later combined to match the topic distribution of the search query to give a query-specific Personalised PageRank score. This produces a measure of authority that is uniquely tailored to the search query.

Recall from Section 2.3.4 that there are two components to Personalised PageRank where personalisation can be implemented, *Bias(t,d)* and *Transition(t,d,d')*:

$$PR(t,d,k+1) = \alpha \times Bias(t,d) + (1-\alpha) \times \sum_{d' \in link_{in}(d)} Transition(t,d,d') \times PR(t,d',k)$$

Figure 5. Sample Results of Topic Modelling on a Collection of Papers.

where for topic *t*: *PR(t,d,k)* is the PageRank for paper *d* at iteration *k*; *Bias(t,d)* is the probability that paper *d* is picked randomly from a corpus; *Transition(t,d,d')* is the transition probability of following a citation from paper *d'* to paper *d*; $link_{in}(d)$ is the set of all papers that cite paper *d*; $link_{out}(d)$ is the set of all papers that are cited by paper *d*.

Also, recall from Section 2.3.4.4 how Yang et al. (2009) automatically generate their Personalised PageRank using topic models with the following model:

$$Bias(t,d) = \frac{1}{|D|}P(t|d)$$

$$\begin{aligned} Transition(t,d,d') &= P(t|d,d') \\ &\cong \frac{1}{|link_{out}(d')|} \end{aligned}$$

While their bias function reflects the probability of topic *t* in document *d*, their choice of *P(d|d',t)* in their transition function does not involve any of the LDA topic distributions. My idea and contribution concerns the case where the random surfer follows a citation, i.e. the transition probabilities. In my model, citations that are closely associated with a topic are favoured, whereas in the model of Yang et al, citations are chosen at random.

TPR incorporates topic probabilities into both the bias and transition functions. Specifically, *Transition(t,d,d')* takes into account the probabilities of topic *t* in not only documents *d* and *d'*, but also in the other documents *d''* referenced by document *d'*:

$$Bias(t,d) = \frac{P(t|d)}{\sum_{d'' \in D} P(t|d'')}$$

$$\begin{aligned} Transition^*(t,d,d') &= P(t|d,d') \\ &= \sqrt{\frac{P(t|d')}{\sum_{d'' \in D} P(t|d'')} \times \frac{P(t|d)}{\sum_{d'' \in link_{out}(d')} P(t|d'')}} \\ Transition(t,d,d') &= \frac{Transition^*(t,d,d')}{\sum_{d'' \in link_{in}(d)} Transition^*(t,d,d'')} \end{aligned}$$

Here, *d* is a document whose TPR is being calculated, *d'* is a document that refers to document *d* and whose TPR score is being distributed during this iteration of the algorithm. The set $link_{out}(d')$ is all the papers that are cited by paper *d'*. The set $link_{in}(d)$ is all the papers that cite paper *d*. The first probability term inside the root of the transition function ensures that TPR scores are propagated only from citing documents that are highly relevant to topic *t*. The second probability term ensures that a larger proportion of a document's TPR score is propagated to cited documents that are highly relevant to topic *t*. By combining the two probability terms using the geometric mean, combinations are penalised when either component probability is close to zero. The value *P(t|d)* can be read off directly from matrix Θ in Figure 1 (c.f. page 26). The transition matrix is

ergodic: its entries are constant, non-zero and sum to unity over each document *d*. This ensures that TPR converges to a steady state.

Figure 6 shows these concepts at play. Consider a corpus of four documents (*d*, *d'*, d''_1 and d''_2) and three citations (shown by arrows). The corpus contains two topics (green and red). The topic distribution for each document is shown by the bar-chart below the document identifier.

Exhibit 1 shows the idea behind the calculation of TPR for document *d* for the green topic. Two factors influence the TPR score for document *d*. Firstly, the green topic is dominant in the topic distribution of document *d'*, so more TPR score is likely to flow from it (indicated by the thicker green arrows in Exhibit 1 than red arrows in Exhibit 2: the sum of the thickness of the green arrows is greater than the sum of the thickness of red arrows). Secondly, document d''_1 is more relevant to the green topic than the two other documents cited by document *d'*, so it receives the bulk of the green TPR score from *d'* (indicated by the thicker green arrow). Document *d* receives hardly any green TPR score because it is less relevant to the green topic than documents d''_1 and d''_2.

Exhibit 2 shows the calculation of TPR for document *d* for the red topic. Because document *d* is more relevant to the red topic, it receives the bulk of the red TPR score (indicated by the thicker red line). This time, however, document *d'* is far less relevant to the red topic so less overall TPR score flows from it (indicated by the thinner red arrows in Exhibit 2 than green arrows in Exhibit 1).

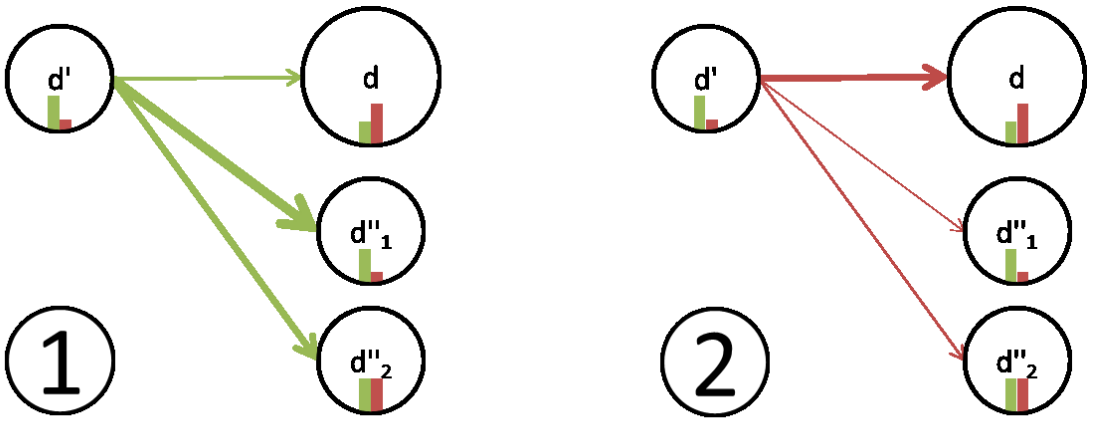


Figure 6. Examples of the Flow of TPR Scores for Two Topics.

Whereas Figure 6 gives the idea behind TPR, Figure 7 gives a numerical example calculation of one iteration of TPR for two topics (green and blue), four papers (A, B, C, D) and three citations. A larger box with A, B, C or D in the top-left corner represents a paper and its TPR calculation. $P(t|d)$ is shown in the top-right. The calculation of TPR for each topic is shown at the bottom. The calculations for the two topics are given in the respective colour. Papers A and B are not cited, so their TPR is comprised only of the bias term multiplied by α, indicated by a_i. The box associated with each citation

arrow shows the calculation of TPR contribution from a citing paper to a cited paper, indicated by b_i. Papers C and D are cited, so they gain an additional contribution from each citing paper, which is multiplied by *(1- α)*.

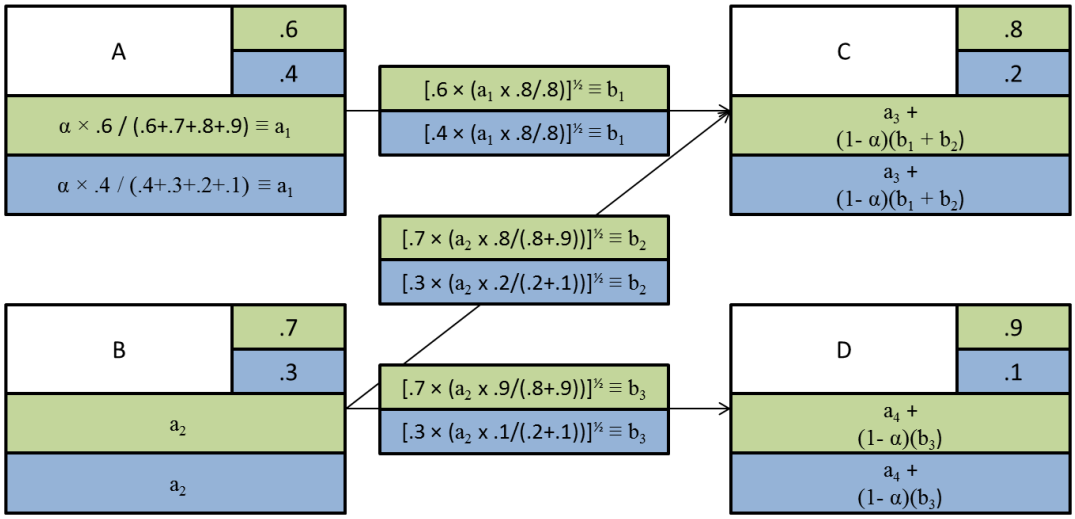


Figure 7. Example Calculation of an Iteration of ThemedPageRank.

There are two compounding effects at play here. The first is that the TPR score for topic *t* is propagated mainly through documents relevant to topic *t*. The second is that the TPR score for topic *t* accumulates mainly in documents relative to topic *t*. Together, these effects generate a Personalised PageRank document ranking that is specialised and unique to each topic. This can then be used to infer highly topic-specific authority.

The iterative calculation of TPR converges rapidly (1-3 iterations) because there are few (if any) cycles in the backwards-in-time citation-graph of scientific literature. Thus TPR can easily scale to millions of papers using a standard desktop computer (for the moderate corpus analysed in this project, TPR converges in less than a second).

○

The second intuition about why a straightforward application of PageRank does not perform well in the context of scientific literature is that there are structural differences between the web and scientific literature. Scientific papers and their citations are frozen as soon as they are published, which tends to cause older papers to score higher PageRank scores overall (Walker et al. 2007). In contrast, web pages are subject to constant change, adding links to newly relevant pages and deleting links to outdated pages, so age has negligible effect on PageRank score.

TPR addresses this problem by adjusting the Personalised PageRank scores based on their age: the scores of older papers are adjusted downwards to compensate for the increased time over which they have attracted citations. The ablation experiments in Chapter 5 show that TPR is reasonably stable to changes in the method of age adjustment,

so it uses the simplest: the Personalised PageRank scores are divided by the age of the paper in years to produce the final age-adjusted ThemedPageRank scores.

3.1.3 Query Model

At this point, TPR has the capacity to automatically fragment a corpus of papers into topics, and for each of these topics, to produce an age-adjusted Personalised PageRank score for each paper. The papers with the highest scores for each topic are the most relevant to that topic, according to TPR.

However, for the task of automatically generating reading lists, a more generalised and expressive query model is necessary. A novice may wish to generate a reading list from a search need expressed in many different ways, as we have seen in Chapter 1: as a single word, a technical term, a phrase or sentence, a paper abstract, a description of the field, a paper in the field, or a small collection of papers in the field.

TPR appeals to matrices Θ and Φ in Figure 1 to convert each of these search needs into a distribution over scientific topics, and then generates a reading list tailored to this distribution. Although the rows of matrix Φ contain the distribution of each topic over the set of technical terms, with appropriate normalisation, the columns φ_i of matrix Φ can be interpreted as the pseudo-distributions of each technical term over the topics.

The following mechanisms might be used to convert a search need into a search topic distribution, ξ:

- **Single technical term**, satisfying the search need, "generate a reading list based around this technical term." If the search query is a single known technical term (i.e., was in the vocabulary of technical terms during the topic modelling stage), then the corresponding normalised column φ_i of matrix Φ is used as the topic distribution, i.e. $\xi = \varphi_i$.
- **Multiple technical terms**. If the search query consists of multiple known technical terms, then the average of the single technical term distributions is used as the topic distribution, i.e. $\xi = \frac{1}{N}\sum_{i=1}^{N} \varphi_i$.
- **Unknown words or technical terms**, satisfying a free-form search query. In this case, we cannot appeal directly to matrices Θ and Φ in Figure 1, as the words or technical terms in the search query are not represented in the LDA topic representation because they were not present in the vocabulary during the topic modelling stage. Instead we perform a Lucene TF-IDF-based document search over the papers in the corpus. The 20 highest scoring papers are selected, and their corresponding paper distributions are averaged to produce a topic distribution in the same way as for the "multiple known papers" mechanism. The assumption behind this approach is that the top papers are those that best match the words in the search query. Although they might individually represent a wide variety of topics, their average topic distribution offers a most likely topic

distribution for the search query. This is the query mechanism used in the experiment in Section 5.2.

- **Single known paper**, satisfying the search need, "generate a reading list based on the content of this paper." If the search query is a single known document (i.e., was in the corpus of papers during the topic modelling stage), then the corresponding row θ_i of matrix Θ is used as the topic distribution, i.e. $\xi = \theta_i$. This is the query mechanism used in the experiment presented in Section 5.3.
- **Multiple known papers**. If the search query consists of multiple known papers, then the average of the single paper distributions is used as the topic distribution, i.e. $\xi = \frac{1}{N}\sum_{i=1}^{N} \theta_i$.
- **Unknown papers**. Finally, if an unknown paper or set of papers is presented as the search query, their combined full-text can be used as the search query in two ways. Firstly, the combined full-text can be scanned for known technical terms and a topic distribution can be built from the weighted average of the topic distribution of the technical terms. Secondly, the combined full-text could be used as a query for a Lucene TF-IDF-based document search over the papers in the corpus, similar in function to the "unknown word or technical terms" mechanism.

Regardless of how it was generated, the query topic distribution ξ is then used to linearly combine the multiple topic-specific TPR vector of scores into a unique ThemedPageRank vector of scores tailored to the search query q for each document d as follows:

$$TPR(q,d) = \sum_{t=1}^{T} \xi_t \times TPR(t,d,\infty)$$

where T is the number of topics, and ξ_t is the component of the query distribution allocated to topic t. $\sum_{t=1}^{T} \xi_i = 1$, by construction. This formula incorporates the ideas of Jeh & Widom (2003) about linearly combining Personalised PageRank scores (c.f. Section 2.3.4.1).

Figure 8 illustrates this calculation. The graph along the top shows the query distribution ξ, which is dominated by topics 9 (light green) and 2 (red). Below the graph are the TPR vectors for each topic, ordered such that the paper with the highest TPR score is first in the vector. The final query-specific ThemedPageRank is shown on the right. Notice how the ordering of the papers in the final query-specific ThemedPageRank more closely resembles the ordering of the topic-specific ThemedPageRank associated with the topics with higher probability in the query distribution. This is a consequence of their higher ξ_i values in the linear combination of topic-specific ThemedPageRank scores.

ThemedPageRank reports the most authoritative papers for the search query as those with the highest query-specific TPR scores.

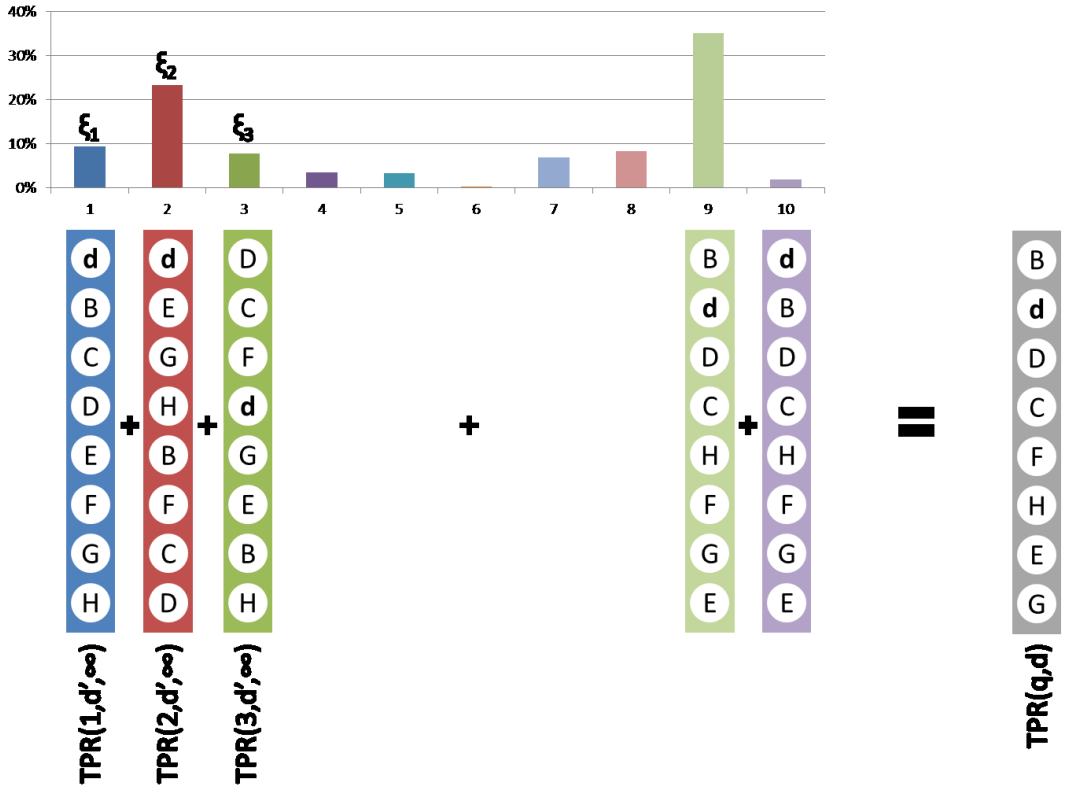


Figure 8. Calculating a Query-Specific ThemedPageRank Score.

3.1.4 Incorporating New Papers

As new papers are published, it is desirable that they can be efficiently incorporated into the document corpus used by ThemedPageRank so that they might immediately become available for recommendation.

There are three components to the calculation of TPR: automatic technical term generation; topic modelling; and the generation of Personalised PageRank scores. Topic modelling is by far the most expensive of the three operations by two orders of magnitude. Fortunately, there is a rich literature for efficiently scaling topic modelling (at least in the context of LDA and NMF) to corpora of millions of papers.

However, it is not necessary to recalculate TPR after each addition of a new paper. Using a strategy similar to the query model for "unknown papers," it is possible to cheaply generate a topic distribution for a paper from the existing topic model. Under the assumption that both the topics and technical terms remain constant, this process first scans the new paper for existing technical terms, and then combines the topic distributions of those technical terms in a weighted average to generate a topic distribution for the paper. Calculation of TPR scores is almost instantaneous, so they can be recalculated whenever a new set of papers is incorporated into the corpus.

It only becomes necessary to regenerate the technical terms and the topic model from scratch once enough new papers have been added and the old set of technical terms and topics no longer adequately represents the corpus as a whole.

3.2 Gold-Standard Reading Lists

Evaluation of reading lists is fraught with difficulty. The most problematic is the fact that we can never get into the mind of the novice to truly determine whether one reading list is better than another at improving their understanding of a field. The best we can do is rely on external judgements either by the novices, who by definition are not in a position to know how well they know the field, or by experts, who are probably better able to judge the relevance of the reading material for the novice. Of course, we will still have to accept the fact that the experts' judgements are inherently subjective.

One of the contributions of this thesis is the creation of a set of gold-standard reading lists collected from experts. This answers the first question this research addresses: *can experts make reading lists when given instructions, and how they go about doing so?*

Appealing to experts to judge the relevance of reading lists in an ad-hoc fashion is expensive, and has the additional disadvantage that the experiment cannot be repeated under the same conditions. The creation of gold-standard reading lists solves these two problems: the expensive process of creating the gold-standard can be amortised by reusing the gold-standard for many experiments.

Little research has been done to automatically construct reading lists directly from the scientific literature. Of the little that has been done, most comes from the field of collaborative filtering, where there is less interest in the contribution of the paper texts themselves, and more interest in the contribution of user ratings. This might explain why little effort has been made before to collect expert gold-standard reading lists against which the generated reading lists might be tested.

The collection of the gold standard reading lists serves three purposes for this thesis. Firstly, it produces a reusable gold-standard set of reading lists against which algorithms can be tested and ranked against each other, using the general ideas of the test collection paradigm (Cleverdon et al. 1966; Cleverdon 1960). Secondly, it allows the experts' behaviour to be observed while they generate the reading lists. This gives some insight into what experts believe a reading list should contain, and also in how they go about building their reading list. Thirdly, it confirms that reading lists can be created on demand by an expert.

Each reading list in the gold-standard is comprised of papers judged relevant to a particular topic or area by an expert. Relevance judgements are known to differ across experts and for the same expert at different times (Voorhees 2000), and this is likely to also be a problem in the gold-standard gathered in this work. Despite this, it has been shown that relevance judgements can be a stable form of evaluation for information retrieval (Burgin 1992; Buckley & Voorhees 2000).

The test collection paradigm (Cleverdon et al. 1966; Cleverdon 1960) requires three components: a corpus of papers; a set of queries; and, for each query, a list of relevant papers from the corpus.

3.2.1 Corpus of Papers

The experiments in this thesis use the November 2010 ACL Anthology Network (AAN) (Radev et al. 2009b) as their corpus. The AAN contains 15,388 papers drawn from various NLP conferences, journals and workshops since the mid-1960s. There are 72,463 corpus-internal citations.

This choice of corpus is motivated by the fact that the AAN contains a large proportion of the entirety of high-quality papers ever published in the field of Natural Language Processing (NLP). The papers in the corpus cite a large number of other papers in the same corpus (Ritchie 2009), which is important for the TPR calculations explained in Chapter 3. Another benefit of using the AAN is that I have a background in the field of NLP, so I was able to make informal subjective evaluations of the quality of technical terms and topics while developing the methods reported in this thesis.

For the collection of gold-standard reading lists, the use of the AAN has two advantages. Firstly, it is easy to find experts in one's own field, and these experts are familiar with the concept of gold-standards. Secondly, it was very likely that the AAN would contain the majority of papers chosen by the experts.

3.2.2 Subjects and Procedure

From each expert, my goal was to obtain two artefacts: a name for their field; and a reading list of papers contained in the AAN for their field.

Eight gold-standard reading lists were produced by experts from the NLP departments of two universities (five from the University of Cambridge, three from the University of Edinburgh). All experts had an NLP-related PhD degree and several years of research experience.

For geographical reasons, the experts in each of the two groups were recruited in a slightly different manner, although the creating of the reading lists proceeded identically. For each of the five experts in the first group, a one-hour face-to-face interview was scheduled verbally in September 2011 for later in the same week. Just before the interview the expert was given the instructions shown in Figure 9 by email. For each of the three experts in the second group, an email was sent in October 2011 with the instructions shown in Figure 10 and a face-to-face interview was scheduled for later in the same month.

> Your task is to construct a reading list that is suitable for an MSc student who wants to do a small research project in your area. They need enough information to build up a sufficient background to decide which aspects of your field most interest them. Armed with this knowledge they can then pick a certain aspect for their research project and independently research more around that aspect.
>
> For as many papers as possible, please could you give a few words to describe why you included the paper in the list?
>
> One limitation is that the papers need to be in the ACL Anthology Network. A list of eligible papers is attached.

Figure 9. Instructions for Gold-Standard Reading List Creation (First Group).

> Your task is to choose a research topic suitable for an MSc student unknown to you who wants to do a small research project in your area, and to create a reading list for it. The topic should be of a size that is reasonable for this scenario, and the reading list should cover all relevant aspects of the topic. Your selection of papers needs to be in the ACL Anthology Network.
>
> In preparation for our meeting, please could you choose a topic and email it back to me, along with your availability. We will create the reading list together during our meeting (alternatively, you may already have a reading list ready, which is also good). I have attached the metadata of the papers in the ACL Anthology to this email, which we will need during the meeting.

Figure 10. Instructions for Gold-Standard Reading List Creation (Second Group).

At the beginning of the interview, an electronic version of the AAN was made available to the expert so that they could use it to confirm that each of their selected papers was in the AAN. They used their own university workstation, which meant they had access to their own emails, past papers, and the Internet, which enabled them to perform among others, Google searches, Google Scholar searches, and searches on journal websites.

During the interviews, the interviewer (the author of this thesis) answered only procedural questions. To any questions from the expert regarding the form, length or content of the reading lists, the interviewer answered only "Whatever seems reasonable for the task." The instructions and guidance were intentionally minimal so as to influence the experts as little as possible. In this way, the manner in which the experts created their reading lists could be observed (see Section 3.2.4).

The result of each interview was a text file containing the name of the expert-chosen field, and a catalogue of the papers in their reading list.

3.2.3 Lists Generated

Table 2 shows the topics of and the number of papers in each reading list. It also shows the number of papers that the expert would like to have included in the reading list, but could not because they were not in the AAN corpus. The gold-standard reading lists themselves are available in Appendix A.

Topic	Papers in AAN	Papers not in AAN
concept-to-text generation	16	5
distributional semantics	14	1
domain adaptation	11	0
information extraction	9	2
lexical semantics	14	0
parser evaluation	4	3
statistical machine translation models	5	0
statistical parsing	22	0

Table 2. Number of Papers in Each Gold-standard Reading List.

3.2.4 Behaviour of Experts during the Interviews

The following anecdotal results emerged during the interviews:

- Interviews lasted about 30 minutes, the shortest being 20 minutes and the longest being 50 minutes.
- Most experts understood the task almost immediately without requiring more information about the form of a reading list.
- Experts had an intuition for the number of papers that should be in a reading list, although this number differed from person to person. When asked after the interview about the length of their list, experts commented that it was influenced by the breadth of the subject area, the level of the novice they had in mind, and the availability of worthwhile papers.
- Some experts already had a core reading list for their topic: one for a class they taught; and another for a book chapter they had recently written.
- Some experts already knew the core authors and papers in the field, and only had to search for a few less-memorable papers. Generally they found these papers by scanning the bibliography sections of the core papers.
- Some experts used search tools like Google Scholar or the ACL Anthology website to recall the names or authors of papers. For some, having access to the supplied AAN database was sufficient.
- When a paper was not available in the AAN, an expert generally was able to find a replacement paper. On two occasions did experts mention that an important part of their field was not at all represented in the AAN, so they limited the broader scope of their field to those topics contained in the AAN.

- Some of the easily recalled papers were those that won the best paper prize at a conference.
- While most experts focussed on the precursors and originators of ideas in their field, one expert focussed on the most recent papers in their field.

3.3 Citation Substitution Coefficient (CSC)

As mentioned in Section 3.2, the evaluation of reading lists is difficult. In this thesis we use gold-standard reading lists to overcome some of the difficulty, but gold-standard reading lists still suffer from the fact that they are inherently subjective. Experts have preferences and their relevance judgements may differ. Also, valid substitutes for papers may be available and be found by the system because they have similar content, similar publishing time frames, or describe joint inventions.

Sections 2.5.1 and 2.5.2 discuss some of the arguments against relying solely on precision, recall, F-score and MAP: they are unable to take into account subjectivity and partial relevance. Because F-score does not take into account partial relevance, it tends to assign low scores to automatically generated reading lists. Low scores approaching zero make meaningful comparison between different systems difficult, particularly when testing for statistical significance. One solution is to decrease variance by increasing the sample size of the gold-standard, but collecting gold-standard reading lists is expensive. This situation demands a metric that achieves statistical significance even for small data sets while at the same time allowing for a fine-grained comparison of the quality of different systems' reading lists.

Section 2.5.3 describes RCP, which rewards partially relevant recommended papers for being frequently co-cited with gold-standard papers. However, RCP penalises recent papers and it is incompatible with the utility-theoretic approach to information retrieval: it assigns equal score to reading lists with many similar partially relevant papers as it does to reading lists with diverse papers that are perfectly relevant to specific papers in the gold-standard.

A contribution of this thesis is the introduction of Forward and Reverse Citation Substitution Coefficient (FCSC and RCSC), which not only tackle partial relevance but are also compatible with the utility-theoretic approach to information retrieval.

FCSC and RCSC estimate partial relevance using the degree of substitutability between expert papers and system-generated ones. The degree of substitutability between two papers is related to the number of links in the shortest path between them in the citation graph. Reporting both FCSC and RCSC scores gives a good overall picture of system performance, particularly when read together with the F-score.

3.3.1 Definition of FCSC and RCSC

FCSC, a metric between 0 and 1, gives higher scores to system papers closely related to expert papers by citation distance: the FCSC of each expert paper, E_i, is related to the inverse of the number of nodes in the minimal citation graph connecting the expert paper to any system paper, S_j. Expert papers directly retrieved by the systems score 1, whereas expert papers citing or cited by a system-generated paper (i.e., those related to the expert paper by a citation path of length 1) score ½, and so on. An expert paper not connected in the citation graph to a system-generated paper scores zero.

$$FCSC_{E_i} = \max_j \left\{ \frac{1}{C[E_i, S_j]} \right\}$$

where $C[E_i, S_j]$ is the number of edges in the shortest path between expert paper E_i and system paper S_j in the citation graph. The overall system FCSC score is the average FCSC of all the expert papers:

$$FCSC_{system} = \frac{1}{|E|} \sum_{i=1}^{|E|} FCSC_{E_i}$$

FCSC corresponds to the role of recall in document retrieval, which measures how many expert papers are found (or recalled) by the system-generated result set. Whereas recall assigns zero score to any missing expert papers, under FCSC, a missing expert paper might still receive partial score through its connection to system papers in the citation graph.

No meaningful information retrieval evaluation can be based on recall alone: a naïve implementation can score recall of 1 by returning all possible papers. RCSC is introduced to mirror the equally important role of precision. RCSC, which ranges between 0 and 1, is related to the inverse of the number of nodes in the minimal citation graph connecting each system paper to any expert paper. The overall system RCSC score is the average RCSC of all system papers:

$$RCSC_{S_j} = \max_i \left\{ \frac{1}{C[E_i, S_j]} \right\}$$

$$RCSC_{system} = \frac{1}{|S|} \sum_{j=1}^{|S|} RCSC_{S_j}$$

Whereas precision assigns zero score to system papers that are not in the list of expert papers, under RCSC, a system paper might still receive partial score through its connection to expert papers in the citation graph.

For both FCSC and RCSC, up to ten edges are followed in the citation graph when searching for a matching paper. After that, a score of zero is assigned to a test paper.

3.3.2 Worked Example

Figure 11 shows a sample calculation of FCSC and RCSC scores for six system papers (A-F) being compared with five expert papers (G-K). Papers L-P are situated on the citation graph connecting the system-generated papers to the expert papers. Citations are shown as blue edges between the papers. The direction of the citation is not important.

When calculating RCSC, paper A is directly connected to paper G, so it scores a RCSC of 1. Paper B is connected to two expert papers with paper L as a node on the minimum citation graph connecting onwards to papers G and H. The shortest path is two edges long, so paper B scores an RCSC of ½. Paper C has four edges in its shortest connection to expert paper I, so it scores ¼. Paper F is disconnected from all of the expert papers, so it scores 0. The overall RCSC score is the average of the individual paper RCSC scores, i.e. 0.46.

FCSC is calculated similarly. Note that expert paper G is connected to both system-generated papers A and B. Because the shortest path is one edge to paper A, it scores an FCSC of 1. Paper K scores 0 because it is disconnected from any system-generated paper. The overall FCSC score is the average of the individual paper FCSC scores, i.e. 0.45.

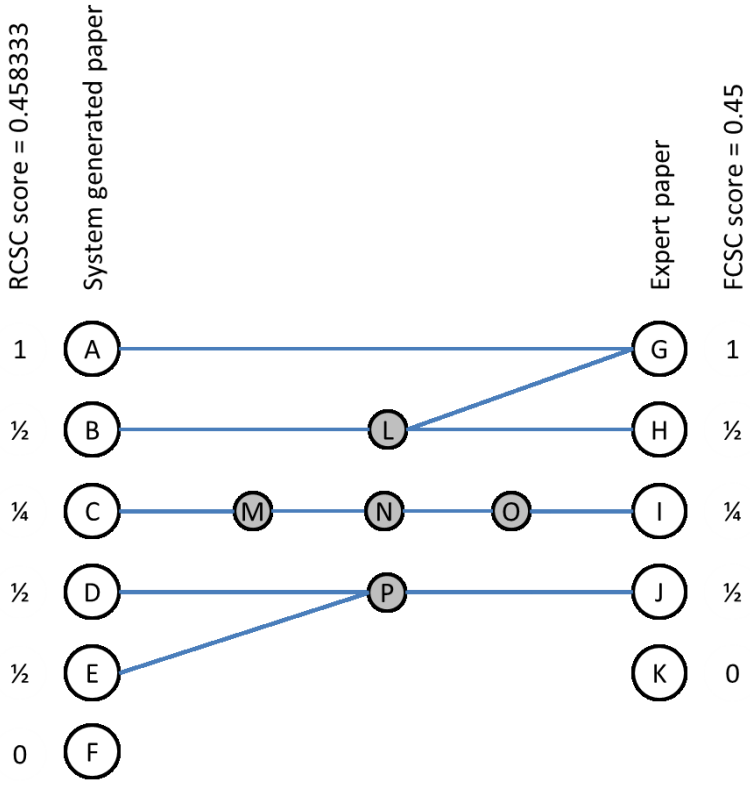


Figure 11. Sample Calculation of FCSC and RCSC Scores.

3.3.3 Alternative Formulations

FCSC and RCSC use an inverse-linear decay function to relate the number of nodes in the shortest citation path to a final score. It is not clear that an inverse-linear decay function ($CSC(n) = 1/n$, where each successive hop scores 1, 1/2, 1/3, 1/4, 1/5...) is the most appropriate of all possible decay functions, but empirically, CSC seems rather robust to changes in the functional form of this reduction both when testing quadratic decay ($CSC(n) = 1/n^2$, where each successive hop scores 1, 1/4, 1/9, 1/16, 1/25...) and exponential decay ($CSC(n) = 1/2^{n-1}$, where each successive hop scores 1, 1/2, 1/4, 1/8, 1/16...).

It may be argued that inbound and outbound citations should be weighted differently when traversing the citation graph because one direction of citation might be more informative of authority than another. While this argument does bear investigation, it is not explored in this thesis. An argument for weighting inbound citations higher is that to receive inbound citations, a paper must have passed an implicit quality assessment to be cited. A counter-argument for weighting outbound citations higher is that when making an outbound citation, an author of a citing paper has to carefully select which papers to cite because of restrictions to published paper length.

3.3.4 Evaluation

FCSC uses the same expert-generated list of papers across each system, so it allows for significance testing via non-parametric paired tests such as the Wilcoxon signed-ranks test. RCSC, with different sets of underlying system-generated papers, is suitable only for standard parametric statistical testing, i.e. non-paired tests.

As discussed in Section 2.5.4, diversity is an important concept to address in the evaluation of generated reading lists. By its very definition, RCSC automatically includes some notion of this diversity. Under the presumption that the gold-standard reading list already represents a diverse enough set of papers, any machine-generated reading list that is not diverse enough to match all the concepts present in the gold-standard reading list will score a very low RCSC.

Ideally, any new metric should undergo a series of calibration tests, comparing the ordering of CSC scores with those of more traditional IR metrics and potentially even calibrating CSC scores on a wide range of IR tasks against user acceptance scores. An empirical evaluation might include comparing the CSC scores to human intuition in a systematic way. For instance, for a given expert-generated reading list, the expert might have been asked which other papers they would have accepted as a substitution for each paper in their list and a score for each substitution (e.g. they might give a 0.8 score if paper A was substituted but 0.6 if paper B was substituted). This would have provided independent calibration points against which CSC might have been evaluated. Holes in the substitution paper list could then be filled using an evaluation metric based on textual similarity.

Although detailed evaluation of CSC as a metric is beyond the scope of my research, we can still get some idea of its suitability in a theoretical sense by checking that its extreme cases make sense:

- If the system generated and expert reading lists are identical, FCSC and RCSC equal 1.
- If one list contains a strict superset of the other, then one of FCSC or RCSC will be smaller than 1.
- If none of the papers in the two lists overlap, and there are not connected in the citation graph or are connected by an unreasonably long path, FCSC and RCSC equal 0.

3.3.5 Summary

In summary CSC is beneficial because:

- It has some measure of "closeness" of hit, unlike precision and recall. This allows for better evaluation of expert-generated gold-standard reading lists because it addresses the problem of subjectivity of expert choice.
- CSC offers a measure in both the forward and reverse directions, giving granular equivalents of recall and precision.
- It can support significance testing: non-parametric tests for RCSC and parametric tests for FCSC.
- It works with recent papers because it takes into account what they cite in addition to whether or not they have been cited.
- It is compatible with the utility-theoretic approach to information retrieval. If many system-generated papers have similar content, they will tend to have the same short route to the same expert paper. The remaining expert papers will be penalised because they are relatively distant in the citation graph to these system-generated papers and have few other alternatives where there might be a shorter route.
- It is cheap to compute. At first glance, it would appear that the search space generated by recursively following citations should grow exponentially. However, papers relevant to a particular area of science tend to cite each other, so relatively small cliques of inter-referencing papers emerge inside large corpora.
- It incorporates the concept of diversity.

I will now turn to another contribution of my thesis, which is based on the working hypothesis that technical terms are more appropriate for building latent topic models in science than words are.

3.4 Light-Weight Title-Based Automatic Term Recognition (ATR)

Technical terms are important artefacts for representing knowledge in and formulating knowledge from scientific texts (Ananiadou 1994; Frantzi & Ananiadou 1997). Sager et al. (1980) describe technical terms as the linguistic representation of the concepts in a particular subject field. They have been shown to benefit a wide variety of NLP tasks (Kim et al. 2010), and play an important role in this thesis because they form part of the underlying representation of scientific papers. Technical terms exemplify tasks (e.g., *statistical machine translation, sentence compression*); methods (e.g., *Naïve Bayes, SVM*); implementations (e.g., *C&C, Indri*); resources (e.g., *Wordnet, The ACL Anthology Network*); features (e.g., *alignment score, MAP*); and many other artefacts. The prevalence and importance of technical terms in science is a well-documented phenomenon (Justeson & Katz 1995).

Despite this, technical terms can be difficult to understand to novices because they have particular meaning when used in a scientific context (Kircz 1991). Also, novices do not find synonymous or related technical terms immediately obvious or predictable. This makes their use in scientific search difficult.

Any system intended primarily for use by novices can therefore not rely on those novices' background knowledge to provide technical terms for its operation. Technical terms must be sourced either from an expert-generated repository of technical terms, or they must be discovered automatically. Technical terms sourced from experts are domain dependent, expensive to generate, and do not adapt over time without the dedication of further expensive resources. In this work I therefore resort to automatic term recognition (ATR) for the provision of technical terms.

Automatic Term Recognition (ATR) from corpus data has a long history: detailed historical reviews are provided by Kageura & Umino (1996) and Castellvi et al. (2001). It has also been the subject of dedicated competitive conferences such as Task 5 of SEMEVAL-2010 (Kim et al. 2010).

Most methods for ATR rely on one or more of four broad heuristics:

- Lexical statistics: such as word counts; word co-occurrence statistics; word blacklists and whitelists; e.g., (Sparck-Jones & Needham 1968), (Daille 1995), (Matsuo & Ishizuka 2004), (Kim et al. 2009).
- Grammar rules: most technical terms are agglomerations of nouns or nouns and adjectives; e.g., (Justeson & Katz 1995), (Park et al. 2002).
- Document position: technical terms are more likely to appear in specific sections of a paper, such as the title, the abstract, and the methods section; e.g., (Lopez & Romary 2010), (Treeratpituk et al. 2010).
- Supervised machine learning techniques: an automated process acquires rules from previously marked-up data; e.g., (Frank et al. 1999), (Lopez & Romary 2010).

The HUMB system (Lopez & Romary 2010) makes use of all four heuristics, although they report that grammar rules do not contribute significantly to the performance of their algorithm. While HUMB performed best at the ATR Task at SEMEVAL-2010, it requires substantial infrastructure: supervised training of a variety of learning machines; access to several human-generated domain-specific datasets; and accurate parsing and OCR of PDFs. Other state-of-the-art systems at SEMEVAL-2010 have similarly high infrastructure requirements.

There is a definite trade-off between the performance of an ATR system and its infrastructure and processing requirements. One of my goals was to reduce these requirements as much as possible without sacrificing performance. To do so, it is instructive to summarise the core properties of various ATR systems:

- Most technical terms are bigrams or longer, and n-grams that are terms are significantly more prevalent in text than non-term n-grams (Justeson & Katz 1995) (for $n \geq 2$).
- Technical terms can be long, e.g., "Fletcher–Reeves Nonlinear Conjugate Gradient Method." Most systems that examine the full document text are limited to unigrams or bigrams. An exception is Treeratpituk et al. (2010), which supports up to 4-grams.
- Paper titles contain many technical terms: words in the title are around 3 times as likely to be technical terms as words in the abstract, and around 50 times as likely as words from the whole paper (Nguyen & Luong 2010). Technical terms contained in titles are also most likely to be relevant (Treeratpituk et al. 2010).
- Technical terms do not start or end with conjunctions, prepositions or pronouns (Eck et al. 2008). While this might remove some technical terms that contain ambiguous stop-words, in practice this does not seem to be a problem.
- Complex terms (e.g., "statistical machine translation") usually contain simpler terms (e.g., "machine translation") (Daille et al. 1994).
- One can also observe that most of the unigram technical terms are acronyms.

I use these properties to build a light-weight title-based ATR algorithm.

While HUMB represents the state-of-the-art in ATR, it and its close competitors all require substantial infrastructure, which tends to be domain dependent and must be rebuilt for new domains.

The method I present in this thesis is light-weight and requires little infrastructure. The only infrastructure is a list of stop-words and the Scrabble lexicon. While it may not perform as well as state-of-the-art, it offers a straightforward way to discover technical terms in a corpus that is easy to implement and is readily portable to new domains.

The algorithm is as follows:

- Generate the n-grams that appear in two or more titles of papers.

- Remove the unigrams that are common English words if they appear in the official Scrabble TWL98 or SOWPODS word lists[12], leaving behind only the unigrams that are proper nouns or non-English words. The Scrabble lexicon was used because, by construction, it includes all English words that are not proper nouns or acronyms.
- Remove all n-grams that start or end with auxiliary stop-words (e.g. conjunctions, prepositions, personal pronouns). Again, these stop-words can be retrieved from word lists or dictionaries that indicate parts-of-speech.
- Remove the n-grams whose total frequency in the corpus is within 25% of the frequency of their subsuming (n+1)-gram (so '*machine translation'* remains, but '*statistical machine*' is removed because it only appears in the context of the subsuming technical term '*statistical machine translation*').
- Finally, remove the least frequent 75% remaining unigrams and bigrams.

This algorithm adds to the list of technical terms for a corpus any acronyms that appear in the titles. This is done automatically by extracting words consisting of only uppercase characters in titles that are of mixed case.

3.5 Qiqqa: A Research Management Tool

Automated term recognition (ATR) and topic modelling play important roles in this thesis as building blocks for the automatic generation of reading lists. Both areas are the focus of continued and active research activity. However, a theme evident across both tracks of research activity is that the evaluation of each task is difficult.

3.5.1 Evaluating Automated Term Recognition and Topic Modelling

The commonly accepted method of evaluating ATR systems (Kim et al. 2010) is to use publicly available gold-standard datasets that consist of a corpus of documents manually annotated with technical terms by human experts. While this method successfully evaluates some aspects of ATR systems – specifically the ability of systems to recall expert-generated technical terms – there is criticism that this evaluation methodology is far from comprehensive. Indeed there are a variety of alternatives advocated in the literature. Frank et al. (1999) propose exact term matching. Mihalcea & Tarau (2004) suggest treating semantically similar terms as being correct. Zesch & Gurevych (2009) argue that near-misses should receive partial credit. Litvak & Last (2008) contend that application-based methodologies are the only way to truly evaluate ATR. Regardless of which of these evaluation methodologies is best, they all require gold-standard datasets, which are expensive to create and are domain specific.

In the area of topic modelling, the lack of a single compelling evaluation methodology is equally evident (Wallach et al. 2009b). The authors of contemporary topic modelling techniques suggest that a reduction in perplexity is evidence that their algorithms are

[12] http://www.isc.ro/en/commands/dictionaries.html

improvements to state-of-the-art (Blei et al. 2004; Li & McCallum 2006; Blei & Lafferty 2006). This is a weak and indirect method of evaluation as there is no obvious link between the perplexity of topic distributions and human interpretation of topics. Chang et al. (2009a) show that topic models that have higher performance based on measures of perplexity are not necessarily better topic models as judged by human evaluators. To test the quality of generated topics using LDA and two other topic models they conduct a user satisfaction evaluation using Mechanical Turk[13]. They argue that user satisfaction evaluations have the advantage of directly measuring human response to a hypothesis, and that they can scale relatively cheaply to large sample populations.

I fully subscribe to this view and built a system that allowed me to measure the quality of my algorithms directly by user satisfaction evaluation.

3.5.2 User Satisfaction Evaluations using Qiqqa

During the first few months of working on this PhD thesis I found it increasingly difficult to manage the volume of PDF papers that I was reading. These difficulties arose not only from the perspective of keeping track of which papers I had already downloaded and read, but also from the perspective of remembering what information in each paper was significant for my research. After surveying the tools available for managing this process, I found none that could adequately do the job.

To address the problem, I built the research management tool, Qiqqa[14], and made it publicly available for download. Over the past three years Qiqqa has enlisted over fifty thousand users, and has grown into a research management system that does far more than I originally envisaged.

The areas of functionality in Qiqqa that are relevant to the work presented here are those that allow users to explore their library of papers in PDF format. The first aspect automatically locates technical terms in the papers in a user's library using the algorithm presented in Section 3.4. The second aspect detects themes in the papers using LDA with an underlying bag-of-technical-terms representation using the algorithms presented in Section 3.1.1.

While both aspects of functionality are intended as building blocks for the larger tasks of more easily locating interesting papers, they both present the opportunity to perform user satisfaction evaluations of both automated term recognition and topic modelling algorithms in a system used by thousands of researchers. Both user satisfaction evaluations enlist Qiqqa users to make a subjective judgement on the results of my algorithms applied to the users' own libraries of papers. Qiqqa enables an evaluation audience that significantly outsizes that of Chang et al. (2009a), who enlist only eight human evaluators using Mechanical Turk.

[13] http://www.mturk.com
[14] http://www.qiqqa.com

There are two fundamental disadvantages that apply to any user satisfaction evaluation: the experimental design needs to be kept simple and it is difficult to conduct the experiments in a controlled environment. The experiments that use Qiqqa are described in Section 5.5.

3.5.3 Visualisation of Document Corpora using Qiqqa

Another function of Qiqqa is the ability to visualise a corpus of papers. The visual representation is a graph of nodes representing the papers, authors and themes, and edges representing connections between the nodes such as "cited-by", "cites" or "author-of". These visual representations are similar to those used in co-citation analysis (White & McCain 1998; Chen 1999; Noel et al. 2002), understanding citation networks (Elmqvist & Tsigas 2007; Schäfer & Kasterka 2010), detecting citation influences (Dietz et al. 2007), understanding research trends (Lee et al. 2005; Mann et al. 2006; Boyack et al. 2009), clustering around themes (Schneider 2005; Eisenstein et al. 2011), finding related papers (Gipp & Beel 2009), and finding authoritative papers (Eales et al. 2008).

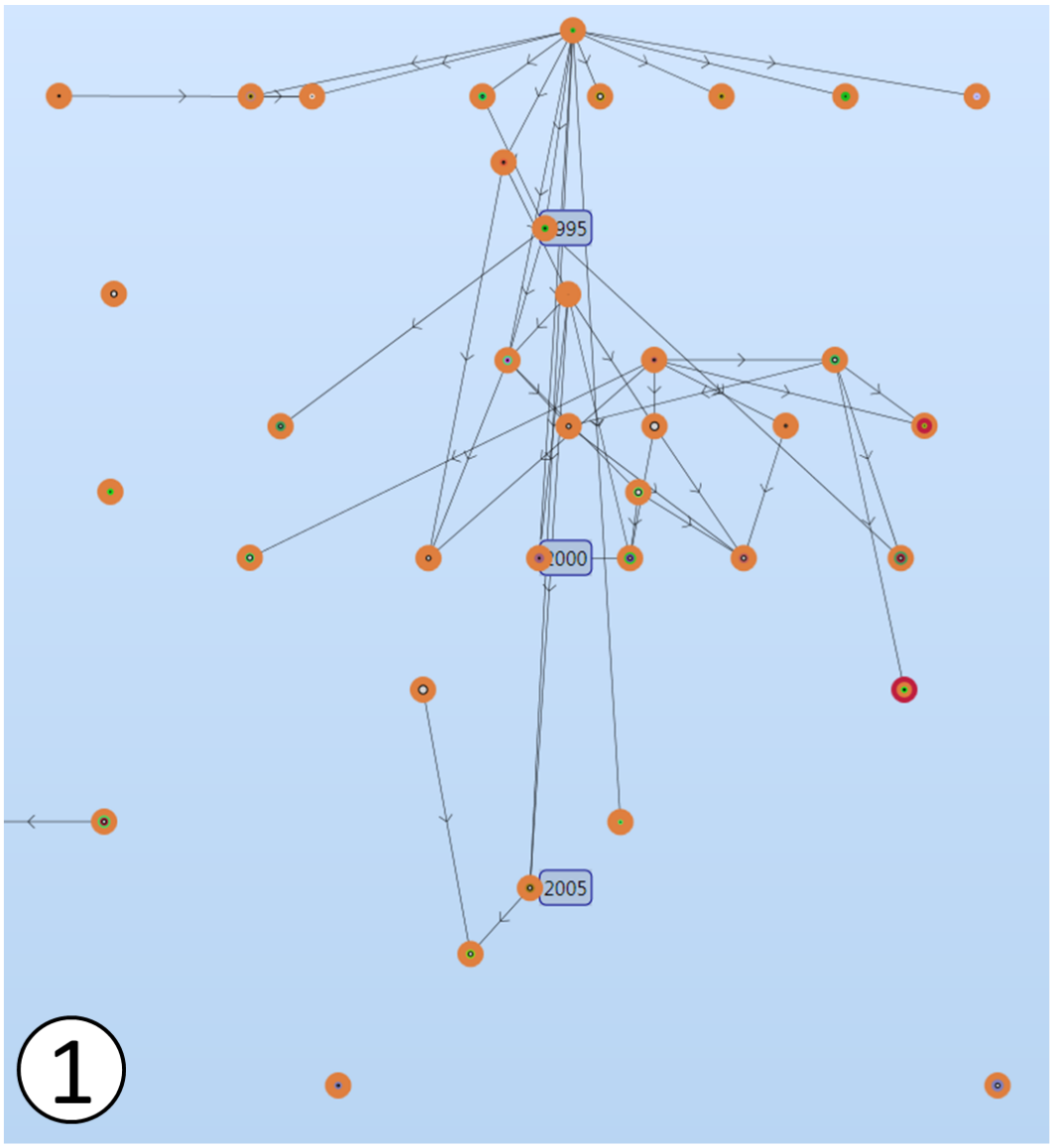
Visualisation techniques play important roles both in improving the effectiveness of information retrieval systems and in understanding the content and relationships inside large document corpora (Hearst 2009). Specifically, Nguyen et al. (2007) present a variety of techniques for visualising massive document collections to allow a searcher to analyse the relationships between scientific documents and retrieve documents more relevant to their search need. Elmqvist & Tsigas (2007) visualise scientific citation networks to improve performance on tasks relating to finding review papers and other influential papers. Noel et al. (2002) use visualisations of co-citation relationships to detect areas of scientific corroboration. Finally, Havre et al. (2000) and Shahaf et al. (2012) use visualisations to follow significant thematic changes through large document collections.

Although not directly used in any evaluation in this thesis, the ability to visualise regions of the ACL Anthology supported the day-to-day development of the TPR algorithm, which is the core contribution of this work. It allowed me to visually evaluate the quality of relationships detected by different versions of the system. The visualisations that were most useful for the development of TPR were those that simultaneously exposed two types of relationships: those implied by the citation graph, and those implied by the topic modelling distributions. The citation graph is depicted as edges between nodes and quickly exposes the relationships between papers that arise through citation. The topic modelling distributions are represented by colouring each document node according to their corresponding topic distribution. This reveals the relationships between papers that arise through the presence of similar topics in their text.

3.5.3.1 Relationships between Topics, Technical Terms and Papers

Figure 12 depicts the relationships that arise between topics, technical terms, and papers through the use of topic modelling distributions and the citation graph. This visualisation

technique allows the user to examine the relationships from the perspective of any topic, paper or technical term. In particular, Figure 12 explores the relationships that are displayed when the user queries the technical term “rhetorical parsing”.



Similar technical terms:
0.112: discourse structure
0.103: RST
0.056: rhetorical structure
0.055: cue phrases
0.052: discourse relations
0.049: rhetorical relations
0.045: discourse markers
0.031: computational linguistics
0.026: rhetorical structure theory
0.023: discourse relation
0.018: structure of discourse
0.017: 11
0.016: discourse structures
0.0 urse parsing
0. se segmentation
0. se analysis
0.0 tructure
0.010: discourse connectives

2

Similar papers:
0.956: 1996 Toward A Synthesis Of Two Accounts Of Discourse Structure by Moser,
0.910: 1986 Assertions From Discourse Structure by Mann, William C.; Thompson, S.
0.870: 1993 The Representation Of Interdependencies Between Communicative Go
0.845: 1998 Identifying The Linguistic Correlates Of Rhetorical Relations by Corston-
0.813: 1993 Intentionality In A Topical Approach Of Discourse Structure by Van Kupp
0.811: 2004 A Framework For Feature Based Description Of Low Level Discourse by
0.803: 1993 Planning For Intentions With Rhetorical Relations by Haller, Susan M.
0.798: 1992 A Problem For RST: The Need For Multi-Level Discourse Analysis by Mo
0.775: 1993 On Discourse Relations, Rhetorical Relations And Rhetoric by Sidner, Ca
0.775: 1993 Textual Constraints, Rhetorical Relations And Communicative Goals And
0.774: 2000 The Rhetorical Parsing Of Unrestricted Texts: A Surface-Based Approach
0.770: 1998 A Surface-Based Approach To Identifying Discourse Markers And Eleme
0.763: 1993 Rhetoric And Intentions In Discourse by Dale, Robert
0.7 Using Cue Phrases To Determine Rhetorical Relations by Knott, Alistair
0. e Rhetorical Parsing Of Unrestricted Natural Language Texts by Marc
0. ointing To Events by Schilder, Frank
0.7 Coherence in Spoken Discourse by Tappe, Heike; Schilder, Frank
0.726: 1995 Investigating Cue Selection And Placement In Tutorial Discourse by Mos

3

Figure 12. The Relationships Involving the Technical Term “rhetorical parsing”.

To find the papers that are most similar to this technical term, the topic distribution of the term is compared to the topic distributions of all the papers in the corpus using Jensen-Shannon divergence. The 50 most similar papers are selected for visualisation and are represented by circles in Exhibit 1 of the figure. Citations between papers (i.e. the edges in the citation graph) are depicted by a directed line from the citing paper to the cited paper. They are arranged using an iterative algorithm where:

- Older papers drift vertically upwards and newer papers drift vertically downwards.
- Papers connected by a citation drift towards each other with a constant force.
- Papers drift away from each other with a force that is inverse-square to the distance between them.

A paper is represented by concentric circles that correspond to topics in the topic model. Each topic is represented by a different random colour. The area of each topic's circle is proportional to the probability mass of the topic in the topic distribution of the paper. In the case of Figure 12, the papers are all strongly associated with the orange topic, although there are some papers that are represented by both the orange and red topics.

Several points are apparent from Exhibit 1 in Figure 12:

- The papers that are most similar in topic distribution to the technical term "rhetorical parsing" are predominantly in the orange topic.
- The papers are densely connected in the citation graph, and the 50 most similar papers cite each other significantly more than would be expected on average.[15] This does suggest that papers strongly related by topic have a tendency to cite each other.
- The papers span the period from 1990 to 2006, as can be seen by their vertical position relative to the blue "year" markers. The field of "rhetorical parsing" has developed over many years.
- There are red and green topics that also influence some of these papers. This suggests that the red and green topics are weakly related to the field "rhetorical parsing," but may be worthy of further exploration.

Exhibit 2 in Figure 12 shows the technical terms with the most similar topic distributions to the technical term "rhetorical parsing." The top terms such as "discourse structure," "RST" (an acronym for rhetorical structure theory) and "cue phrases" would indeed be recognised as relevant to someone versed in the field of "rhetorical parsing".

[15] There are N=13,613 papers in the cleaned-up AAN corpus, with $N \times (N-1)/2 = 92,650,078$ potential citations between them. Of those, only 55,249 edges exist in the AAN citation graph, which is 0.06% of the total potential. For 50 papers, there are 1,225 potential citations. Assuming a relatively uniform distribution of citations across 50 random papers, we would expect only $0.06\% \times 1,225 = 0.73$ citations.

Finally, Exhibit 3 lists the papers with most similar topic distributions to the technical term "rhetorical parsing." Again, these papers are remarkably relevant, as is evident from their titles.

3.5.3.2 Exploratory Reading Strategies

Figure 13 presents a case-study of two possible approaches to exploratory reading. At the bottom centre, slightly enlarged, is an initial paper that a novice might have read. The paper is influenced predominantly by two topics: beige and green. How should a novice choose what to read next during their exploratory reading?

The first approach (shown in Exhibit 1) is reference expansion, which is what a novice might do when approaching a new area. After reading an initial paper, the novice might move onto reading all the papers that initial paper cited. Then for each of those papers they might follow the citations once more. Exhibit 1 shows two iterations of citation expansion.

The first observation is that two iterations of this approach already result in a large number of papers – 14 from the first iteration and 107 from the second. There is no obvious mechanism for the novice to choose beyond the first 14 papers what to read next. If we rely on only the information from the citation graph (via citation expansion), then the next generation of 107 papers appear equally relevant. It is also clear that the citation graph of two iterations of cited papers is highly connected. This might suggest that citation count is not a reasonable metric for differentiating between possible next reading targets, as many papers have similar citation counts.

It is also noteworthy that a large number of the first iteration of cited papers are from the beige and green topics, which makes sense because the author of the initial paper cited them directly. However, after the second iteration of citation expansion, the papers are influenced by a variety of other topics. These topics may or may not be relevant to the novice, but a novice is unfortunately not in a position to make that judgement.

The second approach a novice might use is ThemedPageRank (TPR). Given an initial paper, TPR orders all other papers in the corpus, offering the novice a prioritised and less cluttered view. Exhibit 2 shows the top 33 papers recommended by TPR. The 9 papers surrounded with a white box are those that were cited by the initial paper. Only 4 of the papers cited by the initial paper do not appear in the top 33 TPR recommendations. From this selection of recommended papers, several observations can be made:

- Both the beige and green topics are relevant to the initial paper because they are strongly present both in papers that are directly cited by the initial paper and in papers further away from the initial paper in the citation graph.
- Two early papers (towards the top of the visualisation, predominantly influenced by the green topic) are cited by many of the other recommended papers and are probably originators or early influencers of the ideas in the initial paper. This

idea is perhaps represented by the green topic that influences both them and the initial paper.

- The initial paper cites several papers that seem completely off-topic (represented by the purple, pink and blue topics). These might have been cited not because they are directly relevant to the research in the initial paper, but rather because they describe a specific technique or resource. This information could be useful to a novice before reading a paper so that they can read it in context. By exploring the purple, pink and blue topics, the novice might find other related techniques or resources.

3.6 Summary

This chapter presented the key contributions of this thesis. Section 3.1 presented ThemedPageRank (TPR), the proposed solution of this thesis towards its third research question: *does lexical and social information contribute towards the task of automatically generating reading lists?* In Sections 5.2 and 5.3, TPR will be evaluated by combining the gold standard reading lists described in Section 3.2 together with the Citation Substitution Coefficient metric described in Section 3.3. These same gold standard reading lists provide an answer to the first research question: *can experts make reading lists when given instructions?* Section 3.4 presented the light-weight automatic term recognition system that provides the technical terms required by TPR. Finally, Section 3.5 introduced Qiqqa, which will be used in Section 5.5 to perform two large-scale user satisfaction evaluations.

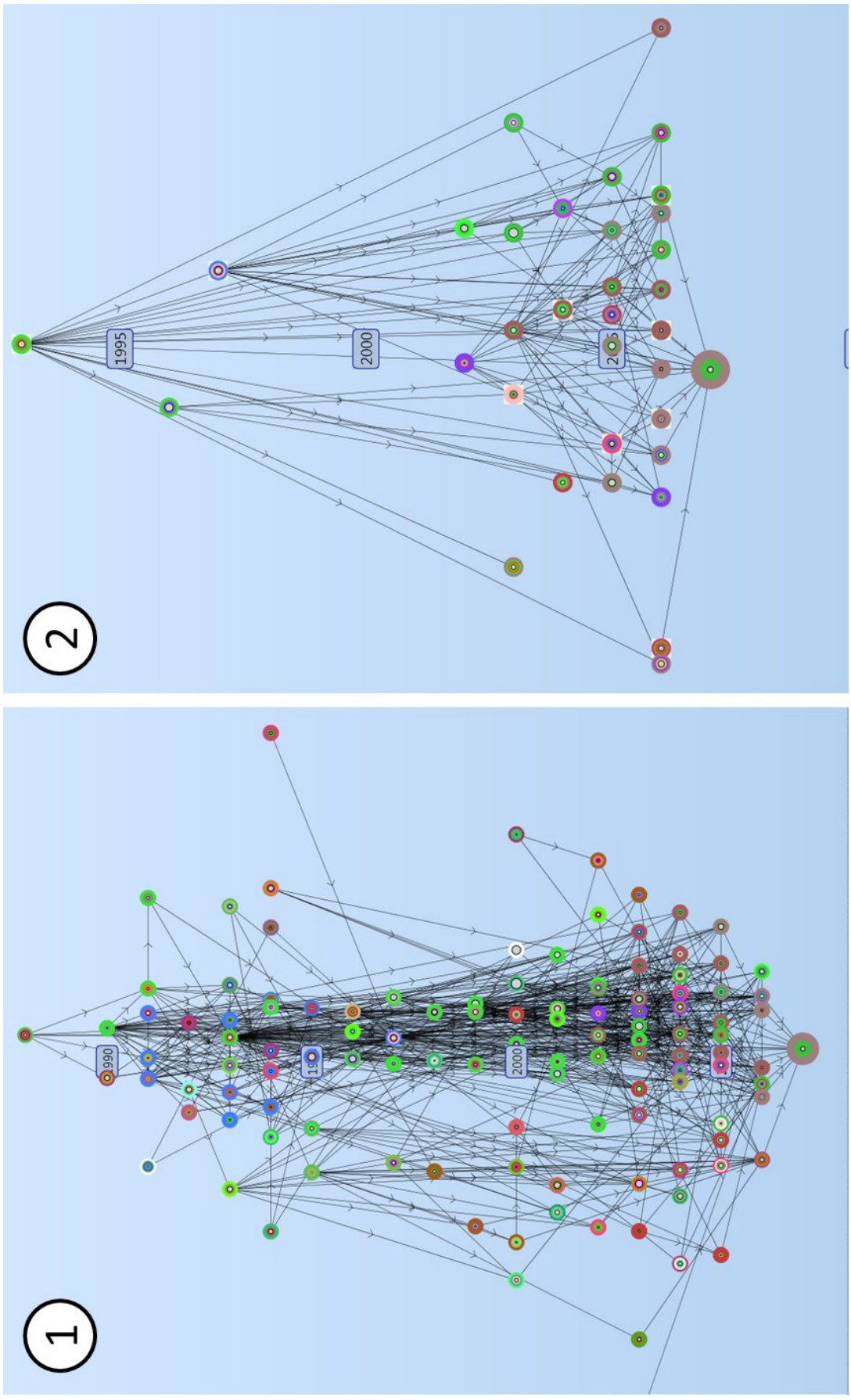


Figure 13. Examples of Recommended Reading for a Paper.

Chapter 4.

Implementation

4.1 Corpus

This thesis uses the November 2010 ACL Anthology Network (AAN) (Radev et al. 2009b) as its corpus. The AAN contains 15,388 full-text documents and 72,463 edges in the AAN-internal citation graph. Each document is referred to by a unique ACL ID. In an earlier but comparable 2005 version of the AAN, the ratio of internal references (references within the AAN) to all references (references within and without the AAN) had been determined at 33% (Ritchie 2009).

For the purposes of the experiments in this thesis, it is necessary to filter out some corrupted documents that are contained in the AAN. This corruption came about presumably through invalid PDF processing, or through human error while assembling the corpus. The filter process removes:

- Documents with a length of fewer than 100 characters;
- Documents with an ACL ID starting with L08 because they are in an XML-based format as opposed to plain text;
- Documents that do not contain the word "the" at least five times. This heuristic detects documents that consist of random sequences of letters;
- And, documents containing only control characters (below ASCII 32).

This process reduces the corpus to 13,613 papers and 55,249 edges in the citation graph.

Figure 14 shows the distribution of AAN-internal citations received by AAN papers. The x-axis indicates the number of citations received by a paper, and the y-axis expresses the number of papers with that number of inbound citations. Notice the downward linear trend on a log-log scale, which suggests a Zipfian distribution in the number of inbound citations. In total only 7,485 of the 13,613 papers in the AAN received at least one corpus-internal citation. There are 2,093 papers that are cited only once and 6,270 papers received 10 or fewer citations. 40 papers received 100 or more citations with the most-cited paper being cited 586 times. It is a well-known phenomenon that a considerable percentage of papers receive no citations at all (Kostoff 1998; Schwarz et al. 1998).

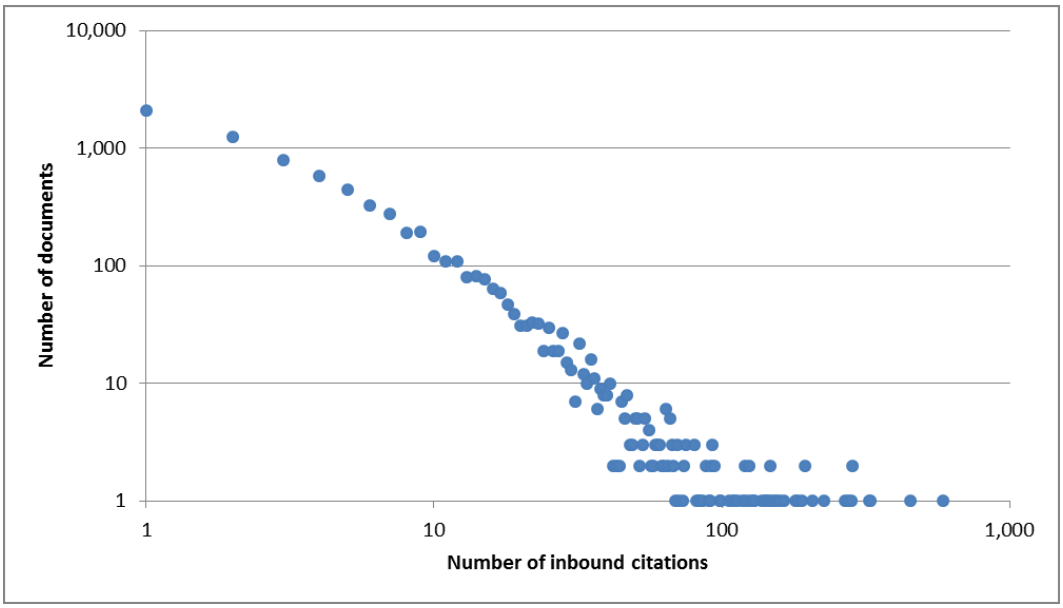


Figure 14. Distribution of AAN-Internal Citations.

4.2 Technical Terms

I implemented the algorithm described in Section 3.4 using C#. It takes approximately 2 seconds to generate the technical terms for the entire AAN corpus on a standard desktop computer. 4,407 technical terms are produced, 1,079 of which are acronyms.

I have already shown some examples of the technical terms generated for the AAN in Table 1 on page 25. Table 3 shows the distribution of the lengths of the automatically generated technical terms. Given that 1,079 of the 1,299 unigram technical terms are acronyms, it is clear that a large proportion of the non-acronym technical terms are bigrams or longer. The longest technical terms found in the AAN are presented in Table 4. It is arguable they they are too long to be technical terms, but keep in mind that they are a concept that has been mentioned in the title of two or more papers and have appeared in the full text of several other papers (as can be read from the "Uses" column).

Length	Count
1	1,299
2	845
3	1,490
4	467
5	174
6	68
7	37
8	13
9	7
10	4
12	3

Table 3. Distribution of Lengths of Automatically Generated Technical Terms.

Length	Uses	Technical term
8	69	minimum error rate training for statistical machine translation
8	22	detection and correction of repairs in human-computer dialog
8	19	figures of merit for best-first probabilistic chart parsing
8	6	terminological simplification transformation for natural language question-answering systems
8	6	experience and implications of the umist japanese project
8	5	identifying perspectives at the document and sentence levels
8	4	automated alignment and extraction of bilingual domain ontology
8	4	automatic component of the lingstat machine-aided translation system
8	3	word-pair identifier to the chinese syllable-to-word conversion problem
8	3	knowledge-based machine assisted translation research project - site
8	2	incremental construction and maintenance of minimal finite-state automata
8	2	unsupervised induction of modern standard arabic verb classes
8	2	contexts and answers of questions from online forums
9	19	named entity transliteration and discovery from multilingual comparable corpora
9	10	augmented transition network grammars for generation from semantic networks
9	5	combining linguistic and machine learning techniques for email summarization
9	3	approach for joint dependency parsing and semantic role labeling
9	2	developed by using mails posted to a mailing list
9	2	named entity recognition based on conditional random fields models
9	2	effect of dialogue system output style variation on users
10	8	large-scale induction and evaluation of lexical resources from the penn-ii
10	7	comparing human and machine performance for natural language information extraction
10	3	new york university description of the proteus system as used
10	3	prosodic and text features for segmentation of mandarin broadcast news
12	4	communicative goals and rhetorical relations in the framework of multimedia document generation
12	3	chinese word segmentation and named entity recognition based on conditional random fields
12	3	hybrid named entity recognition system for south and south east asian languages

Table 4. Longest Automatically Generated Technical Terms.

4.3 Topic Models

TPR relies on topic modelling to provide relationships between the papers and technical terms in the document corpus. The experiments in this thesis explore both LDA (Section 2.2.2) and NMF (Section 2.2.3) as the underlying topic modelling apparatus.

To explore the stability of ThemedPageRank with a different topic modelling technique, the ablation experiments in Chapter 5 compare a variant of ThemedPageRank that uses LDA to generate latent topics with a variant that uses NMF.

For both LDA and NMF, I implemented my own algorithms in C# with a uniform topic modelling interface. This interface allows for experiments that can compare various scenarios in an automated fashion: e.g., differing underlying document representations (bag-of-words vs. bag-of-technical-terms); switching topic modelling models (LDA vs. NMF); and changing the number of topics.

4.3.1 Latent Dirichlet Allocation (LDA)

LDA (Blei et al. 2003) is a Bayesian generative probabilistic model for collections of discrete data that discovers relationships between papers and technical terms.

Under the LDA model with two parameters α and β, the distribution of a corpus D, is the joint probability of all documents, so

$$P(D|\alpha,\beta) = \prod_{d \in D} P(d|\alpha,\beta)$$

The distribution of document d is the sum of all possible combinations of topic distributions Θ that might make up that document, and words in the document that might be made up of those topics t, so

$$P(D|\alpha,\beta) = \prod_{d \in D} \int P(\Theta_d|\alpha) P(V_d, t|\Theta_d, \alpha, \beta) d\Theta_d$$

Expanding the joint probability of all words in the document gives

$$P(D|\alpha,\beta) = \prod_{d \in D} \int P(\Theta_d|\alpha) \left(\prod_{v \in V_d} P(v, t|\Theta_d, \beta) \right) d\Theta_d$$

The distribution of word v is the sum of all possible combinations of topic distributions that might make up the choice of that word, so

$$P(D|\alpha,\beta) = \prod_{d \in D} \int P(\Theta_d|\alpha) \left(\prod_{v \in V_d} \sum_{t \in T} P(t|\Theta_d) P(v|t,\beta) \right) d\Theta_d$$

The distribution of a word *v* given a topic is the sum of all possible combinations of word distributions Φ that might make up that topic, so

$$P(D|\alpha,\beta) = \prod_{d\in D}\int P(\Theta_d|\alpha)\left(\prod_{v\in V_d}\sum_{t\in T}P(t|\Theta_d)\int P(\Phi_t|\beta)P(v|t,\Phi_t)d\Phi_t\right)d\Theta_d$$

For the Dirichlet distributions Θ and Φ, with parameters α and β,

$$P(\Theta_d|\alpha) = \frac{\Gamma(\sum_{t=1}^{T}\alpha_t)}{\prod_{t=1}^{T}\Gamma(\alpha_t)}\prod_{t=1}^{T}\Theta_{d,t}^{\alpha_t-1}$$

$$P(\Phi_t|\beta) = \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\prod_{v=1}^{V}\Phi_{t,v}^{\beta_v-1}$$

The task of calculating the distributions Θ and Φ exactly is computationally intractable. There are a variety of approaches to approximating these distributions. In their original paper, Blei et al. (2003) present an approximation using Variational Expectation Maximization (VEM). Minka & Lafferty (2002) use Expectation Propagation (EP). Asuncion et al. (2009) review a variety of inference algorithms including ML estimation, variational Bayes, MAP estimation and collapsed variational Bayes, and collapsed Gibbs sampling. They conclude that the performance difference between these different inference algorithms is negligible.

ThemedPageRank relies on Gibbs sampling (Steyvers & Griffiths 2007) to approximate Θ and Φ because it is straightforward to implement, and it has been shown that Gibbs sampling converges more rapidly than either VEM or EP (Griffiths & Steyvers 2004).

Initially, each technical term in each document is assigned to a random topic. Then each technical term in each document is re-evaluated against the entire corpus to allocate it to a more likely topic. The probability that a technical term *v* in document *d* should be allocated to topic *t* given all the other technical term allocations *T*, is proportional to

$$P(t_{d,v} = t|T) \propto \frac{DT_{d,t}+\alpha}{\sum_{t'}(DT_{d,t'}+\alpha)} \times \frac{TV_{t,v}+\beta}{\sum_{v'}(TV_{t,v'}+\beta)}$$

Here $DT_{d,t}$ is the frequency of topic *t* in document *d* and $TV_{t,v}$ is the frequency of technical term *v* in topic *t*. Note that the frequencies $DT_{d,t}$ and $TV_{t,v}$ are calculated after removing the topic allocation of the technical term being considered for reallocation. These calculated probabilities are used to randomly assign the technical term to a new topic. The repeated reallocation of the technical terms in the documents to a more representative topic causes their allocations to converge to a steady state, at which point the iterative process terminates.

Because LDA is completely unsupervised, it is not obvious when to terminate training. One approach is to terminate when there is no further or negligible reduction in in-sample perplexity after each iteration (Newman et al. 2006). Another is to terminate once the number of words that change topics after each iteration reaches an equilibrium where the topics associated with each word are unchanged or change back and forth. In this thesis, 400 iterations are calculated, which can be seen in Figure 15 to be more than sufficient to reach this equilibrium.

The matrices Θ and Φ in Figure 1 are then calculated by integrating over the steady-state topic allocations as follows:

$$\Theta_{\mathrm{d,t}} = \frac{DT_{d,t} + \alpha}{\sum_{t\prime}\left(DT_{d,t\prime} + \alpha\right)}$$

$$\Phi_{\mathrm{t,v}} = \frac{TV_{t,v} + \beta}{\sum_{v\prime}\left(TV_{t,v\prime} + \beta\right)}$$

It is an open research question as to how best to choose the values for the hyper-parameters α and β and the number of topics T (Asuncion et al. 2009).

In this thesis TPR uses only constant symmetric priors with parameters as presented in the original LDA paper (Blei et al. 2003), but it has been shown that an asymmetric prior over the topic-document distributions can make substantial improvement over a symmetric prior (Wallach et al. 2009a) in some domains.

TPR uses the parameters $T=200$ topics, $\alpha=2/T$ and $\beta=0.01$. These choices reflect the parameters commonly used in the literature.

LDA via Gibbs sampling scales linearly with the number of topics and the size of the underlying vocabulary (Newman et al. 2006).

Figure 15 shows the running times for 400 iterations of LDA for various numbers of topics. On a 12-CPU desktop machine, 400 iterations of LDA with 100 topics require 18 minutes, 200 topics require 33 minutes and 400 topics require around 70 minutes. It is also clear from Figure 15 when the point is reached where Gibbs sampling achieves equilibrium: after the "elbow" in the lines, additional iterations yield increasingly small differences in the number of words changing topics.

If we use the technical term vocabulary instead of the much larger word vocabulary, an immediate speedup can be observed. Because ThemedPageRank uses a vocabulary of only the 1,000s of technical terms of a corpus of papers rather than the 100,000s of unique words, the calculation of the relationships between papers and technical terms is at least 100 times faster than would be using the entire word vocabulary (remember that LDA is linear in the size of the vocabulary). This is evident in Figure 16, where NFIDF (using the technical terms, n-gram-frequency-inverse-document-frequency) converges remarkably more quickly than TFIDF (using the entire vocabulary, term-frequency-

inverse-document-frequency) for 400 iterations of 200 topics. Consequently, it is likely that ThemedPageRank would scale from the ACL Anthology corpus of 15,000 papers to corpora of millions of papers and still be computable on a standard desktop computer.

In addition to the speed-up offered by a smaller vocabulary, LDA is amenable to implementations that run in parallel across multiple CPUs or multiple networked machines. My LDA implementation of LDA makes use of this parallelisation. Porteous et al. (2008) achieve speed-ups over traditional LDA by up to eight times by improving their Gibbs sampling strategy to take advantage of concentrated topic probability masses. Nallapati et al. (2007) perform VEM in both a multiprocessor architecture and a distributed setting. Asuncion et al. (2008) distribute Gibbs sampling across several processors, each of which communicates their local topic information locally in an asynchronous fashion. Wang et al. (2009b) implement LDA on top of both MPI and MapReduce. They then improve its parallelisability by reducing inter-computer communication costs (Liu et al. 2011). Newman et al. (2007) implement Gibbs sampling in a distributed setting in two ways: the more complex version augments LDA with a hierarchical Bayesian extension that has a theoretical guarantee of convergence; the simpler version approximates LDA by distributing the data across P processors and periodically updating a global state. My implementation of LDA used by TPR follows this latter approach.

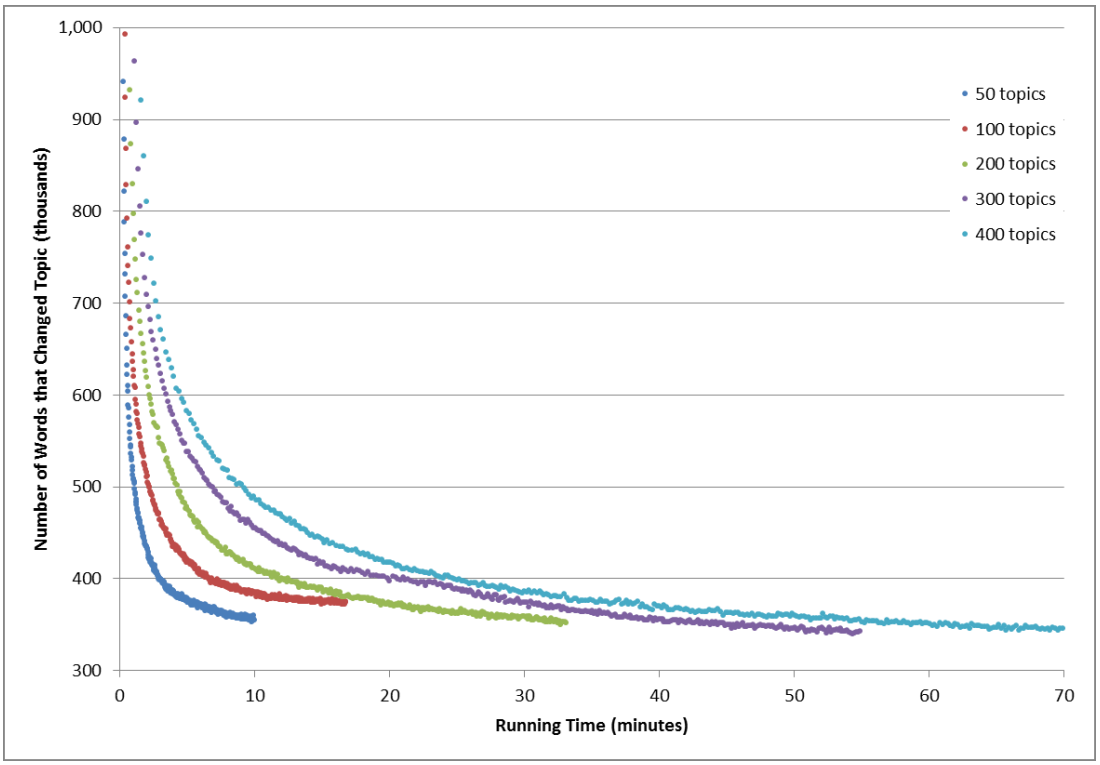


Figure 15. Comparison of Rates of Convergence for LDA Topic Modelling.

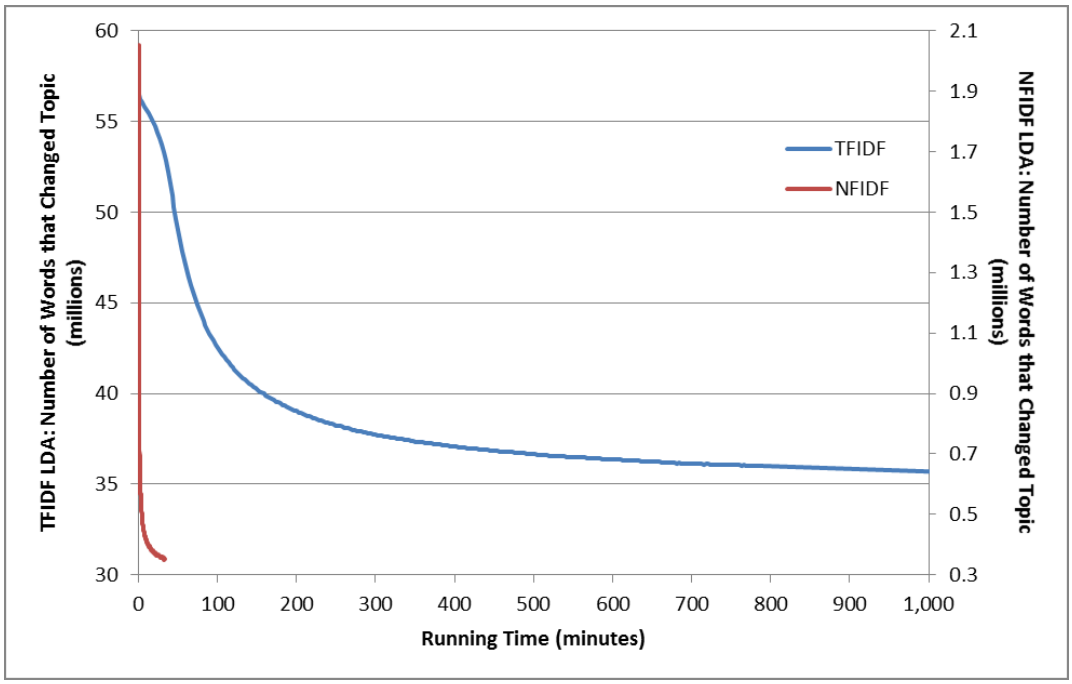


Figure 16. Comparison of Rates of Convergence for TFIDF vs. NFIDF LDA.

4.3.2 Non-negative Matrix Factorisation (NMF)

While LDA was built with NLP tasks in mind, NMF is comparable to LSA in that it is a dimensionality reduction technique that was derived for application in an area of mathematics completely unrelated to NLP. It therefore does not explicitly take into account the concepts of papers, technical terms, and topics. It is up to the person using NMF to map these concepts into the framework of NMF.

Lee & Seung (2001) present two algorithms for performing NMF, based either on minimising the approximation error measured using the Frobenius norm (least square error) or measured using the Kullback-Leibler (KL) divergence. Van de Cruys et al. (2011) conclude that the update rule based on KL-divergence is better suited to modelling text because minimisation of the Frobenius norm supposes a Gaussian distribution in the underlying model, while it is well known that the frequency distribution of words in text follows Zipf's law. The version of NMF explored in this thesis uses update rules that minimises the Kullback-Leibler (KL) divergence.

Recall from Section 2.2.3 that NMF factorises a given matrix X into two non-negative matrices W and H such that

$$X_{D\times V} \approx W_{D\times T} H_{T\times V}$$

As the dimension T is reduced, the product of W and H reproduces the information in matrix X with increasing error, $F(W,H)$. Using Kullback-Leibler (KL) divergence to measure this error,

$$F(W,H) = D(X||WH)$$

$$= \sum_{i,j} (X_{i,j} \log \frac{X_{i,j}}{WH_{i,j}} - X_{i,j} + WH_{i,j})$$

To find the *W* and *H* matrices that minimise *F(W,H)*, we iteratively alternate between two update rules:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} X_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \qquad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} X_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}}$$

Lee & Seung (2001) prove that KL divergence is non-increasing under these update rules, and the update rules become invariant iff *W* and *H* are at a stationary point of the divergence. This guarantees finding a (local) minimum in KL divergence. Lin (2007) proves that these alternating update rules converge, and offers a modification to the update rules that accelerate their convergence. The update rules can be trivially run in parallel across multiple CPUs. My implementation of NMF takes advantage of this parallelisation.

Figure 17 compares the convergence speeds of NMF vs. LDA. To perform 400 iterations for 200 topics, it is clear that LDA converges substantially more quickly than NMF. This is already evidence that LDA might be a better choice for topic modelling than NMF for performance reasons. This trend continues in the experimental results in Sections 5.2 and 5.3. However, keep in mind that one advantage of NMF is that it requires no parameterisation except the number of topics.

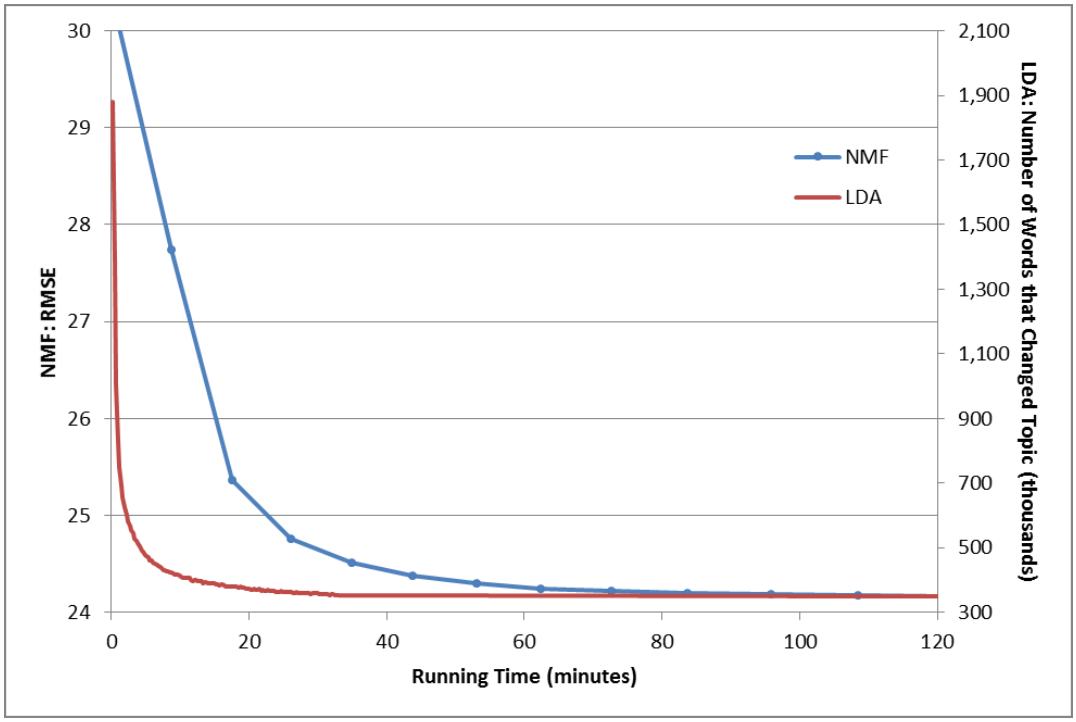


Figure 17. Comparison of NMF vs. LDA Convergence Speeds.

4.3.3 Measuring the Similarity of Topic Model Distributions

Topic modelling provides a mechanism for finding distributions of documents over topics (matrix Θ in Figure 1) and distributions of topics over technical terms (matrix Φ). As explained in Chapter 3, these distributions are fundamental to TPR both in the calculations of Personalised PageRank and in the weights applied by the query model to combine the Personalised PageRank scores.

Notwithstanding their usefulness inside the apparatus of TPR, the topic distributions of documents and technical terms are informative in their own right. The similarity of two documents can be measured by comparing the similarities of their topic distributions, as represented by θ_i and θ_j. Similarly, technical terms can be compared to each other. Interestingly, this mechanism also allows us to compare non-like entities, namely technical terms and documents.

One technique for measuring the similarity between two documents is the dot-product of their distribution vectors θ_i and θ_j (Steyvers & Griffiths 2007). A larger product indicates a stronger relationship between the documents. Here, the similarity $S_{i,j}$ between documents i and j is calculated by

$$\mathrm{S_{i,j}} = \sum_k \theta_i^{(k)} \times \theta_j^{(k)}$$

Another technique is to measure the Jensen-Shannon divergence (Lin 2002) between their probability distribution vectors: a smaller divergence indicates a stronger relationship. Here, the similarity $S_{i,j}$ between documents i and j is calculated by

$$\begin{aligned}\mathrm{S_{i,j}} &= \mathrm{JS(P,Q)} \\ &= \frac{1}{2}[KL(P||M) + KL(Q||M)]\end{aligned}$$

where

$$\mathrm{M} = \frac{1}{2}(\mathrm{P} + \mathrm{Q})$$

$$KL(X||M) = \sum_k X^{(k)} log \frac{X^{(k)}}{M^{(k)}}$$

JS(P,Q) is the Jensen-Shannon divergence between the distributions P and Q, and *KL(X||M)* is the Kullback-Leibler divergence (MacKay 2003) between the distributions X and M.

Other measures of similarity that have been used in the domain of topic modelling are Absolute Difference and the Hellinger Distance, introduced by Blei & Lafferty (2007). The system built for the experiment described in Section 5.4 uses Jensen-Shannen

divergence to suggest similar documents and similar technical terms because Jensen-Shannen divergence has some useful properties:

- It is zero if the two vectors are identical;
- It increases with increasing differences between two distributions;
- It is always finite;
- It is symmetrical;
- Its square-root is a metric; and

These properties produce a similarity score between two papers that can simultaneously be interpreted as percentage scores and used as weights to combine Personalised PageRank scores.

4.4 Examples of ThemedPageRank

Before moving onto the experimental evaluation of TPR, it is instructive to first provide two anecdotal examples of the use of TPR. The first example lists the 27 topics that are automatically detected by TPR in the corpus of around 740 papers I read over the past three years during my PhD research. The second lists the recommended reading that ThemedPageRank suggests for the full-text of Chapters 2 to 4 of this thesis.

4.4.1 Topics Suggested by ThemedPageRank for this Thesis

The Qiqqa library that I used to keep track of my reading for my PhD contains 740 papers. Using the Expedition feature described in Section 3.5.2 and used in the user satisfaction evaluation of Section 5.5.2, Qiqqa generates the following topics and associated technical terms. I leave it up to the reader to evaluate the resulting technical terms and topics.

- `citation; citations`
- `lsa; automatic term; term extraction; evaluation; latent semantic analysis`
- `hits; search engine; web search; world wide web; link analysis`
- `n-gram; keyphrase extraction; information retrieval; KX; automatic keyphrase extraction`
- `pagerank`
- `citation; scientific articles; structure of scientific articles; authors; evaluation`
- `text segmentation; dynamic programming; topic segmentation; reputation; texttiling`
- `similarity`
- `topics`
- `summarization; evaluation; text summarization; multi-document summarization; sigir`
- `collaborative filtering; research papers; recommender system; paper recommendation; sigir`
- `authors`
- `em; parallel; kl; plsa; PSBLAS`

- lda; topics; latent dirichlet allocation; plsi; em
- hierarchy; tng; chinese restaurant process; PLDA+; nested chinese restaurant process
- lsi; svd; JUNG; document clustering; random indexing
- natural language; crf; CRF; natural language processing; information extraction
- bayesian; 2010; variational methods; graphical models; topic-based
- metadata; digital libraries; tutorial; HCI; information extraction
- text mining; scientific publications; h-index; user study; citation networks
- clustering
- similarity; natural language; graph-based; information retrieval; evaluation
- evaluation; information retrieval; sigir; stemming; topics
- topics; topic models; topic model; lda; topic modeling
- nmf; matrix factorization; non-negative matrix factorization; clustering; document clustering
- co-citation; citation; citation recommendation; similarity; clustering

4.4.2 Bibliography Suggested by ThemedPageRank for this Thesis

While the bulk of my reading for this PhD involved papers from outside the ACL Anthology Network (AAN), it is interesting to examine the bibliography that ThemedPageRank suggests for this thesis if the universe of available papers were limited to just those in the AAN.

To generate the bibliography, the full-text from Chapters 2 to 4 of this thesis is used as an input query to TPR using the first of the "Unknown papers" mechanisms described in Section 3.1.3. These are the highest scoring papers from the AAN, in descending score order. The four underlined papers are those that are actually cited in this thesis. While evaluating these results, it is important to keep in mind two points. Firstly, the bulk of techniques used in this thesis originate from outside the AAN. Secondly, the 64 pages that comprise Sections 2 to 4 cover a broad variety of topics (sic), so the 20 recommendations are necessarily going to be quite general.

J93-2004: Building A Large Annotated Corpus Of English: The Penn Treebank
Marcus, Mitchell P.; Marcinkiewicz, Mary Ann; Santorini, Beatrice
1993 Computational Linguistics

D09-1026: Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora
Ramage, Daniel; Hall, David; Nallapati, Ramesh; Manning, Christopher D.
2009 EMNLP

P09-2074: Markov Random Topic Fields
Daume; III, Hal
2009 ACL-IJCNLP: Short Papers

D08-1054: HTM: A Topic Model for Hypertexts
Sun, Congkai; Gao, Bin; Cao, Zhenfu; Li, Hang
2008 Conference On Empirical Methods In Natural Language Processing

P02-1040: Bleu: A Method For Automatic Evaluation Of Machine Translation
Papineni, Kishore; Roukos, Salim; Ward, Todd; Zhu, Wei-Jing
2002 Annual Meeting Of The Association For Computational Linguistics

W09-2206: Latent Dirichlet Allocation with Topic-in-Set Knowledge
Andrzejewski, David; Zhu, Xiaojin
2009 Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing

A88-1030: Finding Clauses In Unrestricted Text By Finitary And Stochastic Methods
Ejerhed, Eva I.
1988 Applied Natural Language Processing Conference

D09-1146: Cross-Cultural Analysis of Blogs and Forums with Mixed-Collection Topic Models
Paul, Michael; Gîrju, Roxana
2009 EMNLP

J96-1002: A Maximum Entropy Approach To Natural Language Processing
Berger, Adam L.; Della Pietra, Vincent J.; Della Pietra, Stephen A.
1996 Computational Linguistics

J98-1004: Automatic Word Sense Discrimination
Schütze, Hinrich
1998 Computational Linguistics

P08-2004: Dimensions of Subjectivity in Natural Language
Chen, Wei
2008 Annual Meeting Of The Association For Computational Linguistics

L08-1005: The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics

Bird, Steven; Dale, Robert; Dorr, Bonnie Jean; Gibson, Bryan; Joseph, Mark; Kan, Min-Yen; Lee, Dongwon; Powley, Brett; Radev, Dragomir R.; Tan, Yee Fan
2008 LREC

W09-0206: Positioning for Conceptual Development using Latent Semantic Analysis
Wild, Fridolin; Hoisl, Bernhard; Burek, Gaston G.
2009 Proceedings of the Workshop on Geometrical Models of Natural Language Semantics

P09-1070: Latent Variable Models of Concept-Attribute Attachment
Reisinger, Joseph; Paşca, Marius
2009 ACL-IJCNLP

W09-2002: Topic Model Analysis of Metaphor Frequency for Psycholinguistic Stimuli
Bethard, Steven; Tzuyin Lai, Vicky; Martin, James H.
2009 Proceedings of the Workshop on Computational Approaches to Linguistic Creativity

N09-2029: Contrastive Summarization: An Experiment with Consumer Reviews
Lerman, Kevin; McDonald, Ryan
2009 HLT-NAACL, Companion Volume: Short Papers

W07-1514: A Search Tool for Parallel Treebanks
Volk, Martin; Lundborg, Joakim; Mettler, Mael
2007 Linguistic Annotation Workshop

W03-0404: Learning Subjective Nouns Using Extraction Pattern Bootstrapping
Riloff, Ellen; Wiebe, Janyce M.; Wilson, Theresa
2003 Conference On Computational Natural Language Learning CoNLL

N09-1054: Predicting Response to Political Blog Posts with Topic Models
Yano, Tae; Cohen, William W.; Smith, Noah A.
2009 HLT-NAACL

W09-2106: An Application of Latent Semantic Analysis to Word Sense Discrimination for Words with Related and Unrelated Meanings
Pino, Juan; Eskenazi, Maxine
2009 Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications

P98-2127: Automatic Retrieval and Clustering of Similar Words
Lin, Dekang
1998 COLING-ACL

4.5 Summary

This chapter presented the implementation details of the algorithms in this thesis. Section 4.1 described the AAN, the corpus used for most of the experiments in this thesis. Sections 4.2 and 4.3 described the technical implementations of the lightweight automatic term recognition algorithm and the topic modelling algorithms used by TPR, respectively. These algorithms, combined with the evaluation in Section 5.4 contribute towards the second research question addressed in this thesis: *does the exposition of relationships between papers and their technical terms improves the performance of a novice in exploratory scientific search?* Finally, Section 4.4 provided two anecdotal use-cases for TPR.

Chapter 5.

Evaluation

Section 3.1 introduced ThemedPageRank (TPR), the primary contribution of this thesis. The performance of TPR is evaluated in two ways. The first evaluation, described in Section 5.2, uses the gold standard reading lists presented in Section 3.2 in the task of automatically generating reading lists. Along with standard IR metrics such as MAP, F-score and RCP, performance results are reported using the Citation Substitution Coefficient established in Section 3.3. In this first evaluation, TPR significantly outperforms two state-of-the-art commercial search systems. While TPR performs well in this task, the collection of gold-standard reading lists is necessarily small because of the cost of collection of each gold-standard reading list. Therefore the second evaluation, described in Section 5.3, is performed on a much larger scale, using the proxy task of reference list reconstruction. TPR significantly outperforms a state-of-the-art system designed specifically for the task of reference list reconstruction. In both experiments, TPR is also compared against numerous ablation system and baseline systems investigate the reasons for the performance of TPR. These systems are summarised in Section 5.1.

Section 5.4 investigates the usefulness of technical terms and topic modelling in the task of exploratory scientific search. A task-based evaluation is used to compare how well novices are able to produce reading lists using two systems. The first system is TTLDA, a precursor to TPR that suggests related technical terms and papers alongside its search results. The second is Google Scholar, a state-of-the-art scientific search system. While no significant difference in performance is evident from the experimental results (which is encouraging in itself), it is evident that the novices search more widely for papers using TTLDA with relevant technical terms previously unknown to them. Using Google Scholar, the novices tend to reuse the same technical terms given to them as part of the search task.

Finally, Section 5.5 presents two simple user satisfaction evaluations that make use of the Qiqqa system presented in Section 3.5. The experiment enlists thousands of users of Qiqqa to independently evaluate the quality of the important technical terms and topics automatically generated from their own collection of PDFs. Almost two-thirds of the users were satisfied with their automatically generated technical terms and topics.

5.1 Comparative Ablation TPR Systems and Baseline Systems

To gain some insight into the reasons for the performance of TPR, the experiments in Sections 5.2 and 5.3 compare TPR not only to real-world state-of-the-art competitor systems, but also against a series of comparative ablation and baseline systems. Each ablation system (those in Sections 5.1.2 to 5.1.5) is a fully-fledged equivalent of TPR

with one component removed or altered in some way. This allows for the investigation of the individual contribution of that component towards the performance of TPR. Each baseline system (those in Section 5.1.6) explores a straightforward implementation of a theoretical component of TPR.

5.1.1 Comparing LDA Bag-of-technical-terms vs. Bag-of-words

TPR models papers using a bag-of-technical-terms representation rather than the traditional bag-of-words representation. There are several reasons for this decision. Firstly, technical terms are important artefacts in science and I claim that the technical terms contained in a paper are more representative of the scientific content of the paper than simple words. Secondly, the universe of technical terms in a corpus is smaller than the universe of words, so the algorithms run more efficiently and consume less memory. And finally, I believe that topics comprised of technical terms are easier to interpret for humans than topics comprised of words. At the very least it is instructive to explore whether the performance of TPR is not diminished by the choice of bag-of-technical-terms over bag-of-words. To investigate the performance of TPR using a bag-of-technical-terms representation rather than a bag-of-words representation, one ablation system is tested:

- **TPR-BAG-OF-WORDS** uses a bag-of-words representation of the documents when calculating the topic probabilities.

5.1.2 Comparing LDA vs. NMF

TPR relies on LDA to provide the topic probabilities that are used in the Personalised PageRank calculations. To investigate the dependency of TPR on LDA, one ablation system is tested:

- **TPR-NEG-MAT-FAC** uses NMF with 200 topics instead of LDA to provide the topic probabilities.

5.1.3 Comparing Bias-only vs. Transition-only Personalised PageRank

An aspect of TPR that makes it different to the varieties of Personalised PageRank in the literature (see Section 2.3.4) is that it alters both the bias and transition probabilities using probabilities automatically derived from LDA topic probabilities. To investigate the incremental improvements realised through the bias and transition, two ablation systems are tested:

- **TPR-BIAS-ONLY** uses the TPR bias probabilities and the original Personalised PageRank transition probabilities.
- **TPR-TRANS-ONLY** uses the original Personalised PageRank bias probabilities and the TPR transition probabilities.

5.1.4 Comparing Different Forms of Age-tapering

Another aspect of TPR that makes it different to the previously published varieties of Personalised PageRank is that it adjusts the Personalised PageRank scores depending on the age of the papers. Walker et al. (2007) suggest that an exponential decay is appropriate, but my own experiments showed similar performance using simple division. Figure 18 shows the difference between the linear and exponential decays. Linear decay penalises older papers much more quickly than the exponential decay. TPR divides the final Personalised PageRank scores by the age in years of the documents. To investigate the sensitivity of TPR to age-tapering, two ablation systems are tested:

- **AGE-NONE** applies no age adjustment to the final Personalised PageRank scores.
- **AGE-EXP** applies an exponential decay to the final Personalised PageRank scores with a half-life of eight years – similar to the $\tau=8$ found in Walker et al. (2007). Under this model, the probability that a paper will be cited halves every 8 years.

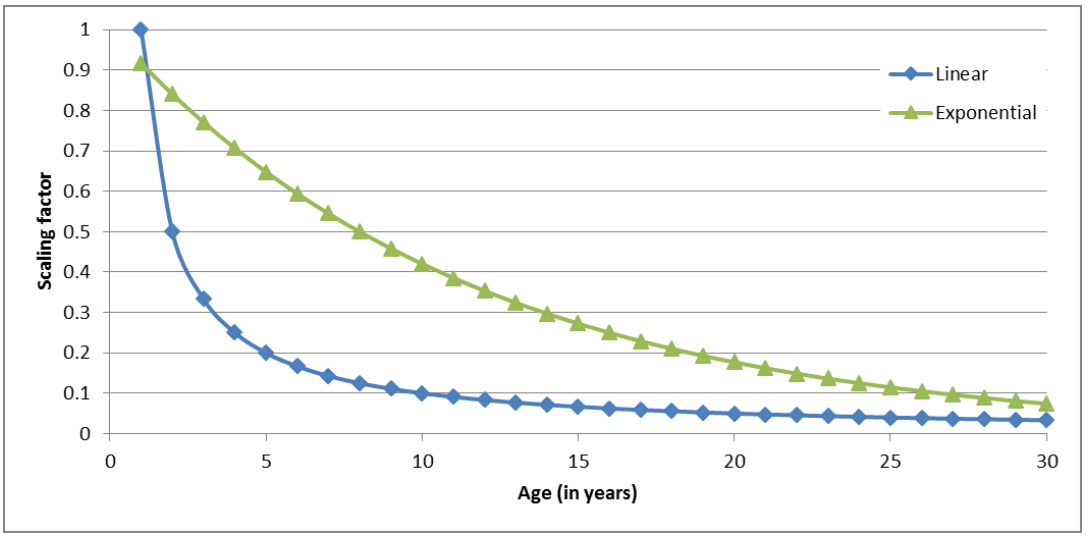


Figure 18. Scaling Factors for Two Forms of Age Adjustment.

5.1.5 Comparing Different Numbers of Topics

TPR uses LDA with 200 topics. The number of topics was chosen in agreement with previous research on similar-sized corpora (Wallach 2002; Mimno & Blei 2011; Wilson & Chew 2010; Chemudugunta et al. 2007; Chambers & Jurafsky 2011). To investigate the sensitivity of TPR to the number of topics in the topic model, four ablations systems are tested:

- **TOPICS-50** uses an LDA model with 50 topics.
- **TOPICS-100** uses an LDA model with 100 topics.
- **TOPICS-300** uses an LDA model with 300 topics.

- **TOPICS-400** uses an LDA model with 400 topics.

5.1.6 Comparing Baseline Components of TPR

This section describes the baseline tests that explore the effectiveness of each of the general components that go into TPR for the task of generating reading lists. These components are TFIDF, Citation Count, Global PageRank and Age-Adjustment and Generic Topic Modelling.

The first test ranks documents by the TFIDF similarity score of each paper with the query. TFIDF, as the most common weighting scheme, is an obvious candidate for a baseline. The TFIDF implementation used here is provided by Lucene[16]. The corpus is indexed with Lucene.NET v2.9.2 using standard out-of-the-box TFIDF parameters.

- **TFIDF** uses nothing but the TFIDF score for paper *p*. *Score(p) = TFIDF(p)*.

The next two ablations explore Citation Count and Global PageRank. They are interesting because they have been used widely in the literature as a proxy for authority. The citation count for a paper is the number of papers in the AAN that cite the paper. The PageRank value for each paper is the Global PageRank score that is generated by calculating PageRank (i.e., with no topic specialisation) on the AAN citation graph with $\alpha=0.5$. Both Citation Count and Global PageRank are then multiplied with the Lucene TFIDF score. While this combination provides only a crude combination of the papers' textual features (via TFIDF) and social features (via Citation Count or Global PageRank), it does allow us to compare the relative performance of Citation Count and Global PageRank.

- **TFIDF-CITCOUNT** multiplies the TFIDF score for paper *p* with the citation count for that paper. *Score(p) = TFIDF(p) × CitCount(p)*.
- **TFIDF-PAGERANK** multiplies the TFIDF score for paper *p* with the vanilla PageRank for that paper. *Score(p) = TFIDF(p) × PageRank(p)*.

Adjusting these authority measures to take into account the age of papers corresponds to the age-adjustment component of TPR. Therefore two additional ablations divide the Citation Count and Global PageRank scores by the age of the paper in years.

- **TFIDF-CITCOUNT-AGEADJ** multiplies the TFIDF score for paper *p* with the citation count for that paper and divides by the paper's age in years. *Score(p) = TFIDF(p) × CitCount(p) / Age(p)*.
- **TFIDF-PAGERANK-AGEADJ** multiplies the TFIDF score for paper *p* with the vanilla PageRank for that paper and divides by the paper's age in years. *Score(p) = TFIDF(p) × PageRank(p) / Age(p)*.

[16] http://lucene.apache.org/core/

Finally, the ablation tests explore the ability of Topic Modelling (LDA) to generate reading lists on its own. As a simple test, documents are ranked by their similarity to the query using KL-divergence (as described in Section 4.3.3) over the topic distributions.

- **LDA-SIMILARITY** uses the KL-divergence between the topic distribution of paper *p* and the query. *Score(p) = KL(p)*.

5.2 Experiment: Comparison to Gold-standard Reading Lists

This experiment uses the gold-standard reading lists created by human experts, as described in Section 3.2, to directly assess the performance of ThemedPageRank in the task of automatically generating reading lists.

5.2.1 Experimental Design

Using the gold-standard field name as a search query, the systems' task is to produce reading lists from the AAN corpus described in Section 3.2.1 that mimic the gold-standard reading lists. The comparison between system-generated reading lists and gold-standard reading lists is performed using a variety of evaluation metrics: FCSC and RCSC, MAP, F-score and RCP (see Section 3.3). To gain additional insight into the performance of the components that comprise TPR, this experiment also investigates the individual ablation systems described in Section 5.1.

ThemedPageRank (TPR) is compared against two widely-used state-of-the-art search systems, Google Scholar (GS) and Google Index Search specialised to the ACL Anthology Network (AAN). It is also compared to a Lucene TFIDF (TFIDF) baseline. Google Index Search and Google Scholar were chosen for this experiment because they represent widely-used state-of-the-art commercial search engines: Google Scholar has established itself as reliable resource for scientific search; and Google Index Search is offered as the default search system behind the ACL Anthology website. Google Scholar is used similarly as a baseline in El-Arini & Guestrin (2011). Lucene TFIDF was chosen because it represents a commonly-used out-of-the-box search system that is easily and objectively reproducible.

Google Index Search and Lucene TFIDF rely on only lexical statistics, making no use of the notion of authority. TPR and Google Scholar incorporate the citation graph as a proxy for authority. As a commercial system, nothing is published about the exact algorithms behind Google Scholar, but it probably also includes additional proxies of authority specific to science, such as the identification of influential authors, journals and schools. For this reason, we would expect Google Scholar to be the toughest competitor to TPR.

For the ThemedPageRank system ("TPR"), the topic name suggested by the expert must first be converted into a topic distribution using the "unknown words or technical terms" method described in Section 3.1.3. To do this, a Lucene TFIDF keyword-based search is used to find the documents in the gold-standard corpus that are most similar to the

topic name. Then, using the topic model, the document topic distribution θ_d is retrieved for each document d (each corresponding to a different θ_i in Figure 1). The query topic distribution θ_q is the average of the document topic distributions:

$$\theta_{q,t} = \frac{1}{|D|} \sum_{d=1}^{D} \theta_{d,t}$$

A TPR score is calculated for each document using this query topic distribution and the *D=20* highest-scoring papers are recommended as the TPR reading list.

For the Google Scholar system ("GS"), the Google Scholar website[17] is used to generate the reading list for each gold-standard query. After submitting the topic name as the search query, the search results are parsed automatically to exclude papers that are not contained in the gold-standard corpus. On average, this resulted in filtering out about a third of the search results – these are the papers that are published in non-AAN venues. The top 20 remaining papers are recommended as the GS reading list.

For the Google Indexed AAN Search system ("GIS"), the "search via Google" search box on the ACL Anthology website[18] is used to generate the reading list for each gold-standard query. After submitting the topic name as the search query, the search results are parsed semi-automatically to exclude papers that are not present in the gold-standard corpus. Most of these are papers that have been published in AAN-venues but have been added to the ACL Anthology since 2010 when the gold-standard corpus snapshot was taken. The first 20 remaining papers are recommended as the GIS reading list.

For the Lucene TFIDF system ("TFIDF"), Lucene.NET v2.9.2 was used with standard TFIDF parameters to index the gold-standard corpus of papers. The topic name is used as the search query and the 20 highest-scoring papers are recommended as the TFIDF reading list.

5.2.2 Results and Discussion

Table 5 shows the results of the gold-standard comparison. The results for TPR are shown in bold and system scores that beat TPR are highlighted in red italic. Where TPR has the highest score, significance is reported with the TPR score and is with reference to the next best system. Where another system beats TPR, significance is reported with the other score, and compares it to TPR. All other significance levels of combinations of systems are not computed. Significance levels are indicated in superscript, with asterisks using * for the 5% level, ** for the 1% level, *** for the 0.1% level, and with a tilde using ~ to indicate no significant difference. For the metrics where significance can be computed using a Wilcoxon signed-rank test, it is indicated by a superscript W.

[17] http://scholar.google.com
[18] http://aclweb.org/anthology-new/

Otherwise significance is computed using Student's t-test, and indicated by a superscript S.

Recall from Section 2.5 that there are many evaluation metrics with different strengths and weaknesses. Also recall that FCSC and RCSC are the new evaluation metrics that I propose in Section 3.3.

Using the measures FCSC, RCSC and RCP, TPR comfortably beats all the other systems with significance at the 0.1% level. Using F-score, TPR beats the others at the 1% level. Using MAP, TPR is beaten by GIS at the 1% level and by GS at the 5% level.

As we were expecting, the significance levels for F-score and MAP are much lower than the other metrics. This is caused by the fact that using the relatively unforgiving F-score and MAP, systems perform either particularly well or particularly badly on each reading list. This leads to a large standard deviation, which in turn leads to lower significance probabilities. This effect was one of the motivations behind my proposing a more granular, and therefore more differentiating, measure like CSC.

	^{S}RCSC	^{W}FCSC	^{S}MAP	^{S}F	^{S}RCP
TPR	*****0.456**	*****0.563**	**0.043**	****0.128**	*****0.012**
GIS	0.317	0.527	***0.054*	0.117	0.008
GS	0.364	0.519	**0.049*	0.112	0.008
TFIDF	0.330	0.412	0.008	0.040	0.003

Table 5. Results for the Comparison to Gold-Standard Reading Lists.

Table 6 shows the results for the ablation systems described in Chapter 5. The results are presented using the same convention as in Table 5. Under most measures, TPR with the bag-of-technical-terms as the LDA document model significantly outperforms the more expensive bag-of-words model (line "TPR-BAG-OF-WORDS"). Using LDA as its topic modelling component also tends to outperform using NMF (line "TPR-NEG-MAT-FAC"), although not remarkably so.

Under most measures, TPR performs substantially better by modifying both the bias and the transition probabilities of its Personalised PageRank component (line "TPR-BOTH" vs. "TPR-BIAS-ONLY" and "TPR-TRANS-ONLY"). Results are less clear when it comes to the choice of time decay function. TPR performs marginally better (at only a 5% level of significance) using exponential decay (line "AGE-EXP") rather than linear decay (line AGE-LINEAR) to time-adjust Personalised PageRank scores.

Using 300 topics (line "TOPICS-300") rather than 200 topics (line "TOPICS-200") produces significant improvements to the performance of TPR. Increasing the number of topics further to 400 does not produce better results – in fact the performance of TPR starts declining once we go beyond a maximum around 300 topics. As described in Section 5.1.5, my choice of 200 topics in the final TPR system was based on the observation that in the literature it has become the default number of topics for corpus

sizes like that of the AAN corpus. It is unfortunate that these ablation tests were performed after all the other experiments, because they uncovered the fact that there is room to explore the effect of the number of topics on the performance of TPR.

Finally, some interesting results emerge when analysing TFIDF, citation counts, global PageRank scores, and topic modelling in isolation. As expected, TFIDF alone performs poorly, which is due to the fact that it uses only lexical information. Incorporating the proxies for authority, namely citation count and global PageRank, produce marginally better results, although it is only when they are age-adjusted that they produce significantly better results. Using only the lexical similarity information provided by topic modelling (line "LDA-SIMILARITY") does only slightly better than TFIDF.

	^{S}RCSC	^{W}FCSC	^{S}MAP	^{S}F	^{S}RCP
TPR	****0.456**	**0.563**	$^{\sim}$**0.043**	***0.128**	**0.012**
TPR-BAG-OF-WORDS	0.450	0.549	0.026	0.110	**0.013*
TPR-NEG-MAT-FAC	0.444	**0.568*	0.043	0.122	**0.013*
TPR-BIAS-ONLY	0.440	0.541	0.027	0.118	$^{\sim}$*0.012*
TPR-TRANS-ONLY	0.397	0.475	0.009	0.051	0.007
TPR-BOTH (TPR)	*****0.456**	*****0.563**	*****0.043**	****0.128**	**0.012**
AGE-NONE	0.436	0.526	0.025	0.110	0.013
AGE-LINEAR (TPR)	**0.456**	**0.563**	**0.043**	**0.128**	**0.012**
AGE-EXP	**0.458*	**0.566*	***0.051*	***0.146*	***0.014*
TOPICS-50	0.450	0.529	0.022	0.089	0.012
TOPICS-100	0.458	0.557	0.030	0.126	0.013
TOPICS-200 (TPR)	**0.456**	**0.563**	**0.043**	**0.128**	**0.012**
TOPICS-300	****0.482*	****0.608*	****0.085*	****0.174*	****0.017*
TOPICS-400	0.445	***0.589*	****0.079*	****0.157*	***0.014*
TFIDF	0.330	0.412	0.008	0.039	0.003
TFIDF-CITCOUNT	0.359	0.419	0.006	0.047	0.003
TFIDF-PAGERANK	0.360	0.450	0.004	0.041	0.003
TFIDF-CITCOUNT-AGEADJ	0.442	0.491	0.016	0.073	0.003
TFIDF-PAGERANK-AGEADJ	0.407	0.512	0.016	0.088	0.008
LDA-SIMILARITY	0.332	0.467	0.025	0.079	0.006

Table 6. Ablation Results for the Automatic Generation of Reading Lists.

5.3 Experiment: Reference List Reconstruction

The experiment in Section 5.2 addresses the task of automatically generating reading lists. Its evaluation against gold-standard reading lists directly tests the goal of this thesis. However, the size of the gold standard, while large enough to provide statistically significant results, begs the question of how the performance of TPR might scale on a much larger dataset. Unfortunately the collection of expert-generated gold-standards is expensive, so we are forced to look elsewhere for proxy evaluations.

The evaluation described here uses the proxy task of Reference List Reconstruction (RLR, which has been described in Section 2.4.6) to measure the quality of TPR's recommendation judgements. Given only the abstract or full text of a paper (with citation information redacted) as an indication of a search need, the task is to predict which papers the target paper originally cited. Recall that an anecdotal example of the operation of TPR in the task of RLR was given in Section 4.4.2.

5.3.1 Experimental Design

This experiment is comprised of two parts.

In the first part, ThemedPageRank is compared against a state-of-the-art system built specifically for the task of RLR (Bethard & Jurafsky 2010). To reproduce their evaluation exactly, RLR is performed on the same subset of 800 test papers in the AAN that they used in their experiment. MAP scores are reported for comparison with their results.

The second part of this experiment explores the application of TPR to the task of reference list reintroduction on a larger scale. Evaluation is done on the full corpus AAN, a much larger test set of around 10,000 papers. To gain some insight into the reasons for the performance of TPR, this experiment again uses the individual ablation systems described in Chapter 5. FCSC, RCSC, MAP, F-score and RCP are reported.

The first part of the experiment compares TPR against the system of Bethard & Jurafsky (2010), henceforce B&J. The corpus used is the same papers that B&J used in their experiment: only those papers in the AAN published in or before 2004. To ensure that this experiment compared system performance over the exact same set of query papers, Steven Bethard supplied me with a list of the paper IDs they used as queries in their experiment, and the resulting average precision score for each query paper. These 794 query papers are all the 2005/6 papers citing five or more other papers in the corpus. To generate an average precision score using TPR, a search query topic distribution is generated for each query paper from the technical terms appearing in their titles and abstracts (using the "multiple technical terms" mechanism in Section 3.1.3). The top 100 papers recommended by TPR for each paper are then compared against their actual reference lists to generate an average precision for each paper, as described in Section 2.5.2. Finally, the MAP is calculated for TPR for comparison with the MAP score of B&J's system.

The second part of the experiment investigates the series of ablations described in Chapter 5 in the task of Reference List Reconstruction. Although the first set of ablation tests in the experiment in Section 5.2 already provided evidence of my claim that the individual components of TPR are advantageous to the final performance of TPR, I decided to run a second set of ablation tests in the RLR scenario to further substantiate the trends found in the first ablation test runs. This experiment uses the gold-standard corpus described in Section 3.2.1, which is the same as used in the experiment in

Section 5.2. The 1,500 papers citing ten or more other papers in the corpus were used as queries in the evaluation. Their topic distribution (i.e. θ_i in Figure 1) was used for a search query per paper (using the "single known paper" mechanism in Section 3.1.3). For each paper *d* in the experiment, its query topic distribution is used to generate a unique ThemedPageRank *TPR(d,d')* tailored to paper *d* over the corpus of papers *d'*. The papers *d'* are then sorted by their ThemedPageRank and the 100 top-scoring papers are recommended as the citations for paper *d*. These recommended citations are compared against the actual citations for that paper using the variety measures detailed in Section 3.3: Citation Substitution Coefficients (RCSC and FCSC), MAP, F-score and RCP.

5.3.2 Results and Discussion

Table 7 shows the results for the evaluation of TPR against B&J. When reintroducing the reference list of B&J's 800 papers, TPR, a simple unsupervised algorithm, outperforms the current state-of-the-art method, which is supervised and substantially more complex. This difference is significant at the 5% level using the Student's t-test.

	^{S}MAP
B&J	0.287
TPR	***0.302**

Table 7. Results for Reference List Reintroduction.

It is remarkable that TPR outperforms Bethard & Jurafsky's system for several reasons. Firstly, B&J is built specifically for the task of RLR, while TPR is more general: it is built for the tasks of automatic reading list recommendation. Secondly, B&J uses a variety of sources of data above and beyond the information used by TPR. Both systems use the paper text, publication year and citation graph, but B&J additionally use authorship, co-authorship, school and publication information and snippets of text from citing documents to use as citation contexts. Thirdly, besides building topic models and calculating PageRanks, B&J requires the expensive training of a support vector machine with a variety of features. Finally, B&J is expensive to compute at build time and at query time, while TPR is expensive only at build time. At query time, B&J requires various comparisons of the query paper to every paper in the corpus, while TPR requires only a cheap constant-time weighted combination of Personalised PageRanks.

Table 8 presents the results for the second part of the experiment that explores the ablations described in Chapter 5. The results are presented using the same convention as in Section 5.2.2. For this task, the bag-of-words representation (line "TPR-BAG-OF-WORDS") outperforms the bag-of-technical-terms representation (line "TPR") used in the topic modelling step. This is in contrast to earlier findings. However, the trend that LDA (line "TPR") is superior to NMF (line "TPR-NEG-MAT-FAC"), which we observed earlier in Section 5.2.2, is continued here.

Again, modifying both the bias and transition probabilities (line "TPR-BOTH") significantly outperforms systems where only one of the two is performed (lines "TPR-BIAS-ONLY" and "TPR-TRANS-ONLY"). This vindicates our surprise at Yang et al. (2009) that they alter only the bias component of their algorithm and ignore the transition component. In both this and the previous experiment, an important contribution to the performance of TPR is that both the bias and transition components are modified.

Linear age-adjustment of PageRank (line "AGE-LINEAR") marginally outperforms exponential adjustment (line "AGE-EXP"), and significantly outperforms not performing any age-adjustment at all (line "AGE-NONE").

Again, the choice of 300 topics (line "TOPICS-300") appears to be optimal for applying topic-modelling on the AAN to TPR. This is unfortunate but could not be changed for the experiments that follow as this was only discovered later on.

Finally, it is evident that citation count (line "TFIDF-CITCOUNT") and global PageRank (line "TFIDF-PAGERANK") can improve the performance of simple TFIDF (line "TFIDF"), and even more so when age-adjusted (lines "TFIDF-CITCOUNT-AGEADJ" and "TFIDF-PAGERANK-AGEADJ"), although nowhere near the performance of TPR. Similarity measures using LDA (line "LDA-SIMILARITY") again slightly outperforms using TFIDF on its own.

	^{S}RCSC	^{W}FCSC	^{S}MAP	^{S}F	^{S}RCP
TPR	**0.448**	**0.825**	**0.268**	**0.158**	**0.009**
TPR-BAG-OF-WORDS	*˜0.451*	****0.843*	****0.291*	****0.166*	*˜0.010*
TPR-NEG-MAT-FAC	0.422	0.771	0.202	0.133	0.008
TPR-BIAS-ONLY	0.440	0.800	0.233	0.147	0.009
TPR-TRANS-ONLY	0.386	0.643	0.081	0.077	0.004
TPR-BOTH (TPR)	***0.448**	****0.825**	*****0.268**	*****0.158**	*****0.009**
AGE-NONE	0.438	0.804	0.242	0.149	0.009
AGE-LINEAR (TPR)	***0.448**	***0.825**	**˜0.268**	**˜0.158**	**˜0.009**
AGE-EXP	0.444	0.820	0.266	0.157	0.009
TOPICS-50	0.444	0.786	0.222	0.141	0.008
TOPICS-100	0.450	0.814	0.256	0.154	0.009
TOPICS-200 (TPR)	**0.448**	**0.825**	**0.268**	**0.158**	**0.009**
TOPICS-300	****0.450*	****0.833*	****0.279*	****0.162*	*˜0.010*
TOPICS-400	0.436	0.814	0.255	0.153	0.009
TFIDF	0.356	0.657	0.062	0.085	0.005
TFIDF-CITCOUNT	0.374	0.673	0.092	0.091	0.005
TFIDF-PAGERANK	0.363	0.661	0.100	0.086	0.005
TFIDF-CITCOUNT-AGEADJ	0.384	0.679	0.108	0.094	0.006
TFIDF-PAGERANK-AGEADJ	0.384	0.677	0.106	0.093	0.006
LDA-SIMILARITY	0.392	0.666	0.115	0.091	0.007

Table 8. Ablation Results for Reference List Reintroduction.

5.4 Task-based Evaluation: Search by Novices

The wider goal of this thesis is the automatic generation of reading lists, and the previous two experiments more or less directly evaluate TPR in achieving this goal. The current experiment, however, explores the value of two fundamental components of TPR for actual search: automatic technical term recognition and topic modelling. It focuses on the particular hypothesis that documents modeled by technical terms and topic models retain sufficient semantic content to be useful to a novice for exploratory scientific search, and can thus form the foundation for the harder task of reading list generation.

This experiment presents a search system (called TTLDA) that combines the automatic recognition of technical terms (see Section 4.2) with LDA topic modelling (see Section 4.3) to augment traditional search. TTLDA presents similar papers with their associated technical terms: related papers can be read in context and understanding of each paper can be enriched by the group of technical terms associated with it.

I claim that this improves novices' scientific search experience in comparison to standard keyword and citation based search in two ways: (i) novices internalise previously unknown technical terms during their search with the system; and (ii) they find more non-obvious papers relevant to their information need.

5.4.1 Experimental Design

Using a task-based evaluation, this experiment simulates exploratory scientific search. Given the name of a scientific field, a novice is given the task to create a reading list using the operations given by one of two test systems, the conditions in this experiment. These systems are TTLDA, an early precursor to TPR that makes use of technical terms and topic modelling, and a suitably disguised version of Google Scholar (GS), a state-of-the-art bibliographic search system. The evaluation metrics are quality scores given to the reading lists by experts who also supplied the initial field names. Because there is repetition bias, a Latin-square design was used. A similar search task and evaluation is presented in Ekstrand et al. (2010).

I recruited four experts from the Computer Laboratory at the University of Cambridge. Each expert was asked to provide the name of their chosen scientific field, knowing that they would have to later determine whether or not novice-selected papers from the AAN were relevant to their field. The fields suggested by four experts were: "*distributional semantics*", "*statistical parsing*", "*domain adaptation*" and "*parser evaluation*".

Eight test subjects (the novices), who did not previously know these scientific fields, were recruited from masters-level classes. In individual 20-minute sessions, these novices found and ranked what they felt were the most relevant papers in each field using alternatively TTLDA or GS. Both systems look similar and operate on identical indexed material. A Latin-square design was used in order to avoid item bias: each subject used an alternate system for each session (the alternate sets of instructions are included in

Appendix B); systems were used in different orders; and fields were presented in different orders.

Once the novices had produced reading lists for each of the fields, the papers in each field were pooled. The pooled papers for each field were marked as relevant or irrelevant by the expert of that field. Evaluation is performed using pooled relevance judgments (Buckley & Voorhees 2004) with human judgments, comparing each of the papers in the subjects' ranked reading lists to the experts' relevance lists, and reported using precision-at-rank-n (P@n). For each rank n, the number of relevant papers in the top-n ranked papers in each list is calculated, and the average precision is reported. In the context of the current experiment, it is reasonable to use precision rather than CSC because experts make their judgements on the novice-selected papers after they have been selected. There is therefore no need to check the similarity of novice-selected papers to expert-selected papers.

In order to ensure a fair comparison between TTLDA and GS, each system provides comparable functionality and an almost identical graphical user interface (GUI). Both systems use the AAN (see Chapter 4) as their underlying corpus. In both systems, return lists of 30 papers are presented to the user.

TTLDA offers a simple free-form text search facility because GS also provides this. TTLDA uses Lucene TFIDF over the AAN corpus to generate search results for a given search query.

For the GS system, a façade with the same simple free-form text search facility is inserted between Google Scholar and the searcher, hiding the fact that Google Scholar is the underlying search system. When a user enters a search query, it is forwarded to Google Scholar. The Google Scholar search results are automatically parsed, filtered to remove non-AAN papers, and only the first 30 papers appearing in the AAN are presented to the user as the paper search results.

Both systems offer the functionality of "*related papers*" and "*cited by*". For "related papers," TTLDA presents the 30 most similar papers in the corpus as determined by relatedness calculations using topic distribution similarity, as described in Section 4.3.3. GS presents the 30 related papers offered by Google Scholar, again filtered to the AAN. For "cited by," both TTLDA and GS list the citing papers from the AAN citation graph. The only visual difference between TTLDA and GS is that TTLDA shows relevant technical terms for each paper (the most similar technical terms to each paper using the same topic distribution similarity as before), while GS shows relevant "*contextual snippets*" harvested from the Google Scholar search results. While this is the only visual difference in the search system, it is a major point in my claim that the presence of these technical terms improves novices' scientific search experience. To generate its initial search results, TTLDA uses TFIDF, presumably a fairly impoverished IR mechanism compared to the state-of-the-art GS system. To compensate for this, TTLDA relies

heavily on the claim that the presentation of relevant technical terms improves scientific search more than does the presentation of contextual snippets.

Figure 19 and Figure 20 show screenshots of sample search results for TTLDA and GS, respectively. The search systems can be used as follows:

1. A novice enters a free-form text search query. When they press ENTER, the top 30 papers relevant to their search query appear below it. Each paper is summarised by items (2)–(5).
2. The title, year, authors and journal of each paper is listed. Clicking on the paper title opens the PDF for that paper.
3. For TTLDA, a list of technical terms relevant to that paper is listed next. Selecting a technical term replaces the current list of papers with 30 papers relevant to that technical term. For GS, a "contextual snippet" is shown.
4. Selecting the words "Related papers" replaces the current list of papers with 30 papers relevant to that paper.
5. Selecting "Cited by nn" replaces the current list of papers with papers that cite that paper. nn is the number of papers in the AAN that cite the paper.

Note that the window titles were added for the purpose of this description. In the actual experiments they read "Paper search."

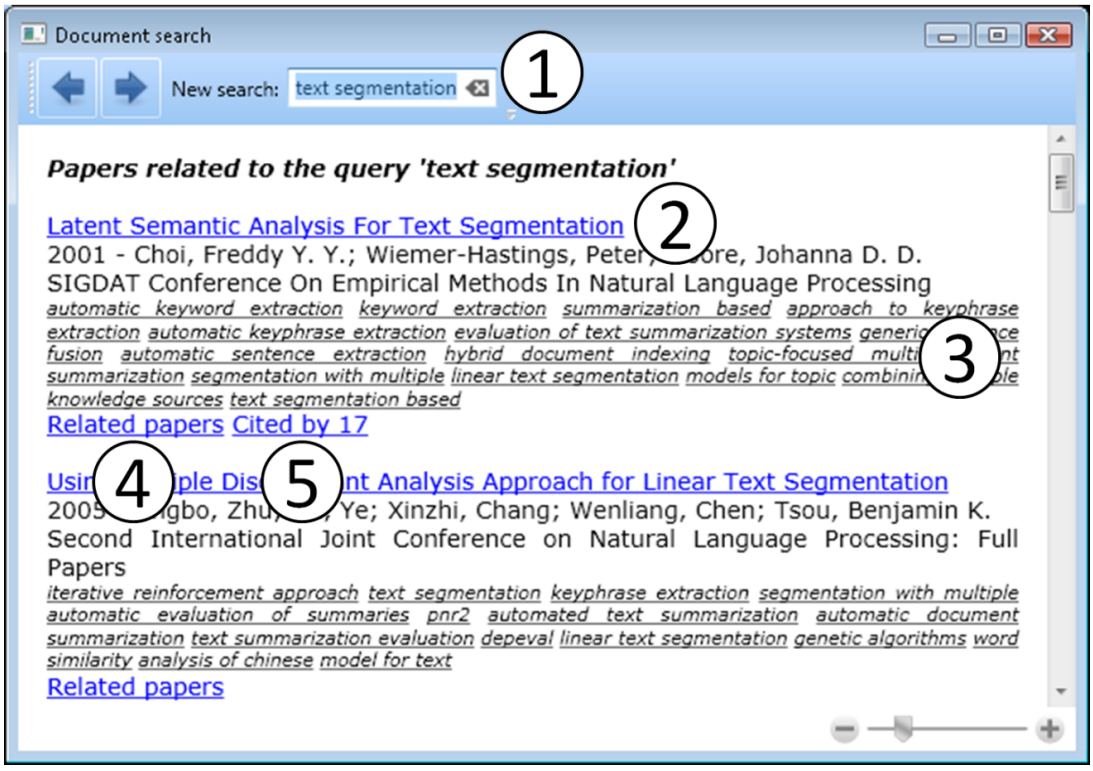


Figure 19. Screenshot of a Sample TTLDA List of Search Results.

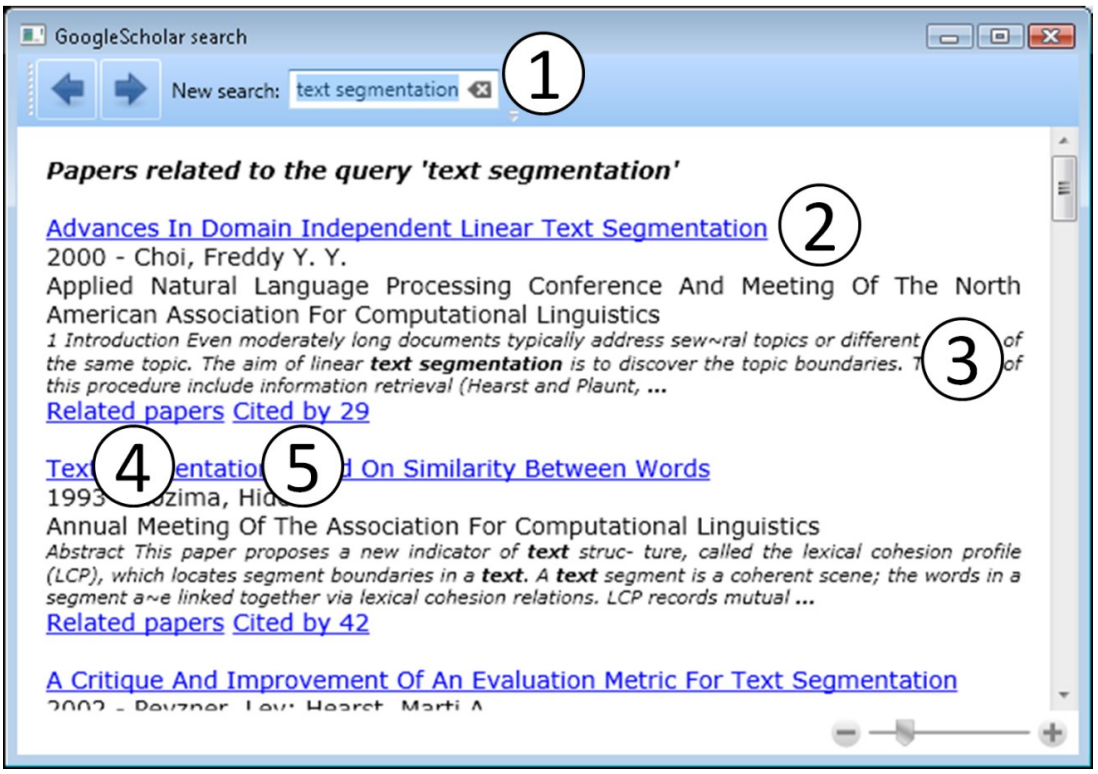


Figure 20. Screenshot of a Sample GS List of Search Results.

5.4.2 Results and Discussion

The eight subjects performed a total of 101 queries and selected a total of 438 papers in 32 lists across the 4 scientific fields using the 2 different systems. The shortest result list produced by the subjects is 4 papers long, while the longest is 25 papers long.

Figure 21 and Figure 22 show the average P@n for TTLDA and GS. The rank is represented along the x-axis, while the precision at that rank is represented along the y-axis. Figure 21 is unscaled, and shows that both systems perform similarly with P@n scores above 80%. Figure 22 is a detailed view of Figure 21, where 1-standard deviation error ranges have been added. As can be seen, there is no clear winner between TTLDA and GS up to rank 15, and even up to rank 25, the error ranges confirm that TTLDA and GS perform similarly without significant difference at the 95% level.

The results substantiate the hypothesis that documents modeled by technical terms and topic models retain sufficient semantic content to be useful to a novice for exploratory scientific search, at least compared to a state-of-the-art search system like Google Scholar. We are thus confident that technical terms and topic models can form an adequate foundation for the task of reading list generation.

Figure 23 shows the total number of relevant and irrelevant documents discovered using each system. The x-axis indicates how many of the highest-ranked documents were included from each novice-generated reading list. We can see that the number of relevant

documents remains relatively similar at all ranks, in the limit 183 for GS vs. 175 for TTLDA.

While GS does retrieve more relevant papers than TTLDA, both the P@n and relevance counts results are noteworthy given the relative simplicity of the TTLDA system compared to GS, i.e. what we can speculate is a more complex state-of-the-art production search system. GS presumably has at its disposal citation graphs, information about author, school and publishing venue.

The results are more interesting when the subject-selected papers are categorised according to how difficult it was to find them. The definition of a "hard" paper is one where the novice can not simply match the field name to the title of a paper. My technical solution was to classify as "hard" any paper whose title does not contain any Porter-stemmed (Porter 1980) words from the field name. We are far more interested in the hard papers because finding easy papers does not require a human's time as it can be automated. The difficult and interesting cases of relevance are those where different lexical items are used in the query and relevant papers.

Figure 24 and Figure 25 show the average precision-at-rank-n for TTLDA and GS for the "hard" papers. TTLDA substantially outperforms GS (finding 34 vs. 12 papers with an average P@n of 63% vs. 53%). This difference is significant at the 95% level. These results motivate that the relationships induced by topic modelling over technical terms helped the subjects find more non-obvious papers which were relevant to their information need than GS was able to provide.

A similar effect can be observed when we consider those cases during the experiment when subjects used the search bar to enter their own formulation of queries. Remember that they are not required to do so because they can perform most searches simply by clicking. However, whenever they do, we have an indication of whether they encountered some new technical terms from the material presented to them during the search so far. Presumably, they did not know these technical terms before the experiment.

Therefore, it was particularly interesting to me to compare what kind of queries were issued when using the TTLDA and GS systems. Table 9 presents the frequency of "hard" queries performed by the subjects. This time, a "hard" query is one that contains none of the words from the name of the field. This finds the occasions when the subject is exploring more widely in their search, rather than just typing in the name or parts of the name of the scientific field to find relevant papers. We find that when subjects use TTLDA, they search more frequently than when using GS – 61 queries vs. 40. This is a good effect because it might indicate that the subjects have a wider variety of way to articulate their search need, presumably using terms they have learned during the experiment. But more importantly, using TTLDA, they also search more frequently using relevant keywords that are not obvious from field names or paper titles. 28% of subjects' queries are "hard" using TTLDA vs. 8% for GS. This might indicate that

novices internalise and use previously unknown technical terms during their search with TTLDA. This would be a positive result because remember how much trouble novices have with new technical vocabulary.

	Easy	Hard	Total
GS	37 (93%)	3 (7%)	40 (100%)
TTLDA	44 (72%)	17 (28%)	61 (100%)

Table 9. Number of Easy vs. Hard Queries Performed by Novices.

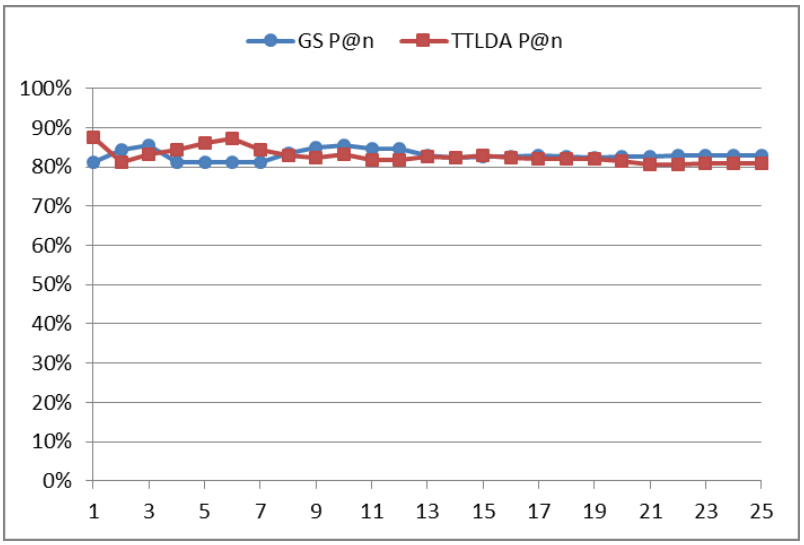


Figure 21. Precision-at-Rank-N for TTLDA and GS.

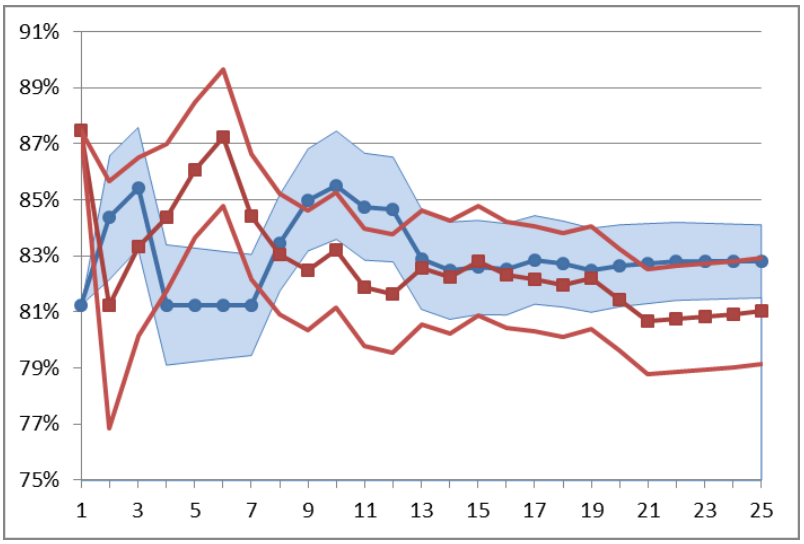


Figure 22. Precision-at-Rank-N for TTLDA and GS (detail).

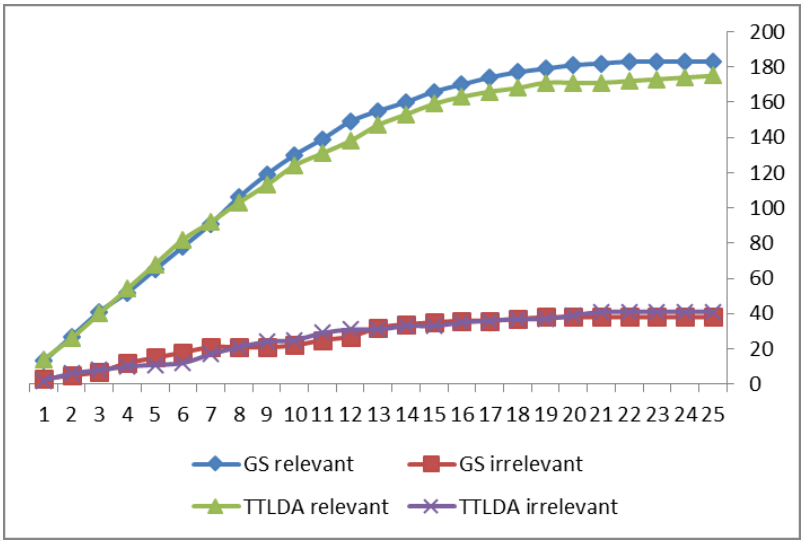


Figure 23. Relevant and Irrelevant Papers Discovered using TTLDA and GS.

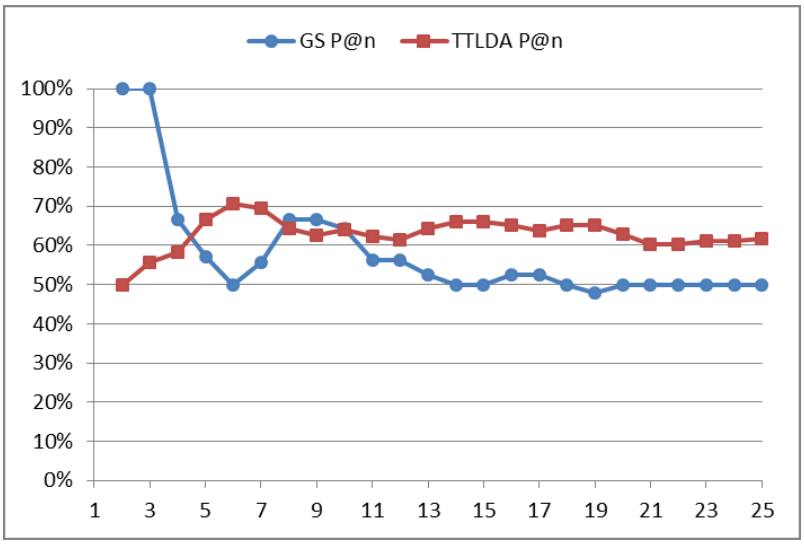


Figure 24. Hard-Paper Precision-at-Rank-N for TTLDA and GS.

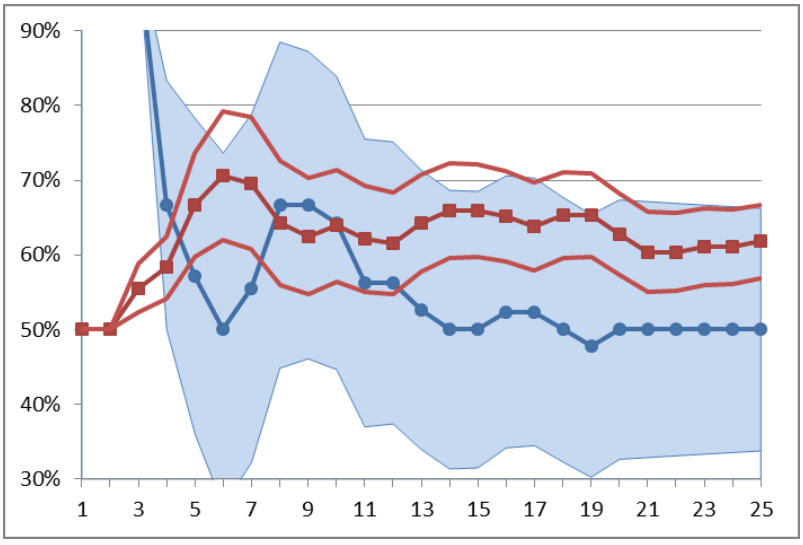


Figure 25. Hard-Paper Precision-at-Rank-N for TTLDA and GS (detail).

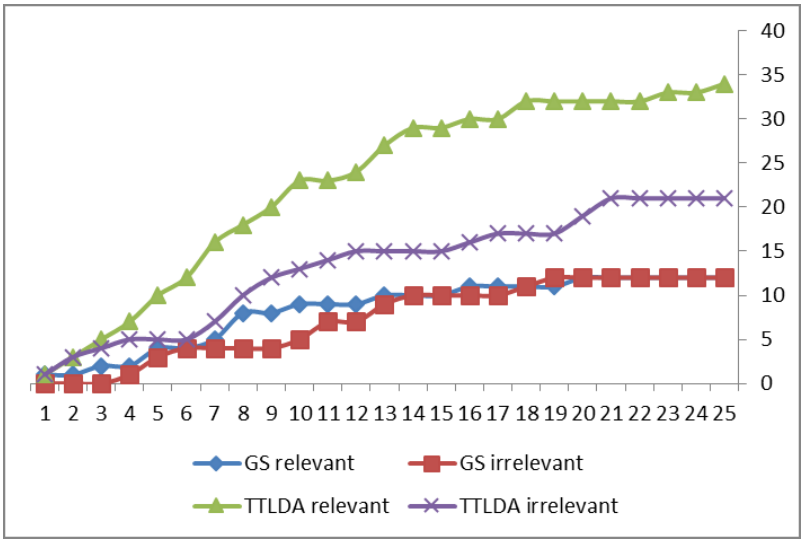


Figure 26. Relevant and Irrelevant Hard-Papers Discovered using TTLDA and GS.

5.5 User Satisfaction Evaluation: Technical Terms and Topics

Automated term recognition and topic modelling play important roles in this thesis as building blocks for the automatic generation of reading lists. As mentioned in Section 3.5.1, while both areas are the focus of continued and active research activity, the evaluation of each task is difficult.

The following experiment employs Qiqqa, a research management tool that I built concurrently with the research reported here, and which is being used by over 50,000 real-world users to manage their own collections of scientific research papers. The self-reported breakdown of the background of the users of Qiqqa is such that 65% of the users are PhD students, 15% are postdocs or full-time academic researchers, 5% are master's students, and the remaining 15% are professionals such as medical researchers, financial analysts and lawyers. Their PDF libraries range in size from tens to thousands of papers, with an average of 200 papers.

The existence of Qiqqa, which uses some of the facets of TPR, allows me to engage the user base in user satisfaction evaluations of the tasks of automated term recognition and topic modelling. Both evaluations ask users to make a subjective judgement on results relevant to their own library of PDF papers in their own field.

5.5.1 Testing the Usefulness of Technical Terms

The first part of this experiment tests the subjective usefulness of technical terms to Qiqqa users. Numerous methods exist for evaluating the correctness of automated technical term recognition systems. This experiment takes a straightforward approach: to automatically generate technical terms on a corpus of papers owned by users of Qiqqa, and then to ask the users to rate their quality.

5.5.1.1 Experiment

Figure 27 shows the Qiqqa user interface for automatically generating *AutoTags* (or technical terms in the nomenclature of this thesis) for a library of PDF documents. The list of PDF documents in the user's library are shown to the right of Exhibit 1. To the left of Exhibit 1, the user has selected the AutoTag filter tab. To the left of Exhibit 2, the user presses the Refresh button to regenerate the AutoTags for their paper collection. Once generated, the AutoTags are associated with the papers that contain them in their full-text. The algorithm presented in Section 3.4 is applied to the titles of the papers in the user's collection. The automatically generated AutoTags are shown to the left of Exhibit 3. The user can filter by an AutoTag by clicking on it. Only the papers associated with the selected AutoTag are shown in the paper list. Exhibit 4 shows the user satisfaction evaluation. After refreshing their AutoTags, the user is prompted to judge positively ("thumbs up") or negatively ("thumbs down") the entire list of automatically generated AutoTags with the following instructions:

We are always trying to improve our algorithms. How do you rate the quality of the AutoTags that Qiqqa generated for you?

The user is not obliged to participate in the user evaluation, so any results are a consequence of the user's voluntary choice to participate. 2,173 users participated.

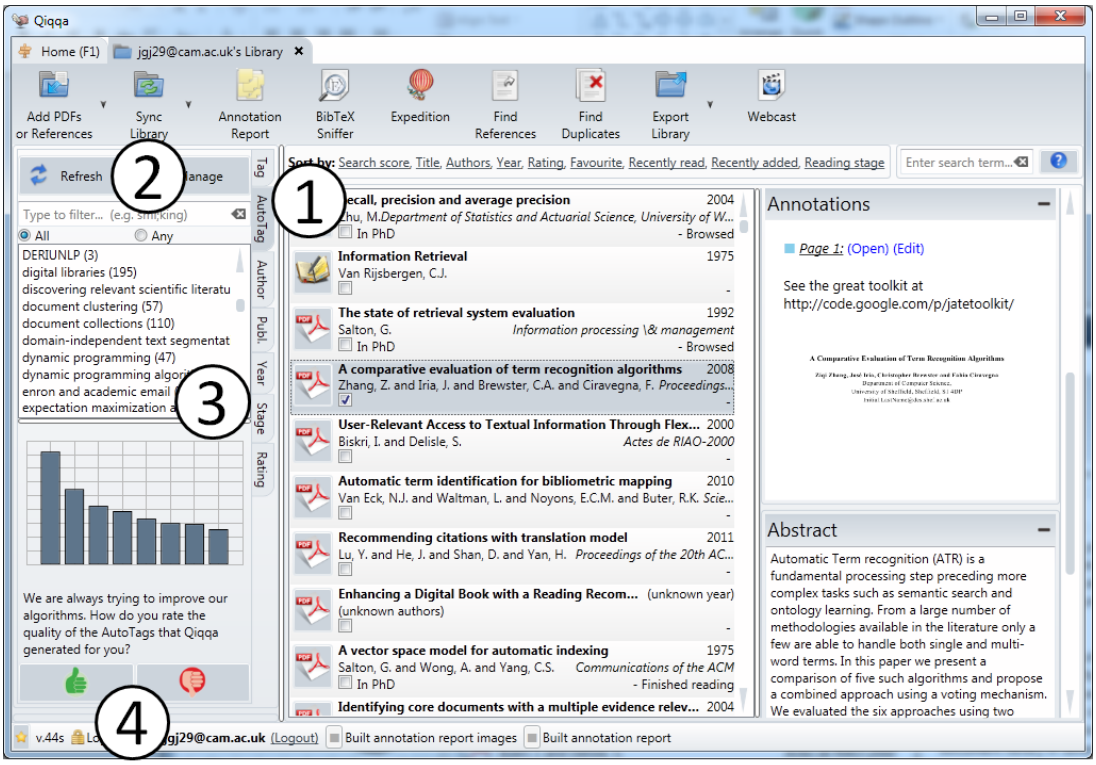


Figure 27. Screenshot of Qiqqa's Recommended Technical Terms.

5.5.1.2 Results and Discussion

Table 10 shows the results of the user satisfaction evaluation. 1,434 (66%) of the 2,173 users that participated in the evaluation give the automatically generated technical terms a "thumbs up", indicating that the automatically generated technical terms are useful to them. Where users submitted a quality judgement more than once (potentially after refreshing their AutoTags a second time), only their first judgement was included in these results.

One hypothesis of what causes negative reviews might be user collections that are too small for the ATR algorithm to work well. The ATR algorithm relies on repetitions of technical terms in the titles of different papers to detect them. It is likely that small collections of papers might not exhibit enough of this overlap to find relevant technical terms. To explore if there is a marked difference in satisfaction for large or small libraries (greater than or fewer than 50 PDFs, respectively), the results are broken down further. The results show that the size of the collection indeed seems to influence the users' quality judgements. 68% of users with a large library give the automatically generated technical terms a "thumbs up", while only 60% of users with a small library do so.

	All		Large		Small	
Thumbs down	739	(34%)	495	(32%)	244	(40%)
Thumbs up	1,434	(66%)	1,065	(68%)	369	(60%)
Total	2,173	(100%)	1,560	(100%)	613	(100%)

Table 10. Results of User Satisfaction Evaluation of Technical Terms.

5.5.2 Testing the Usefulness of Topic Modelling

The second part of the current experiment tests the usefulness of topic modelling. Recall from Section 2.2.4 that a variety of ad-hoc methods exist for evaluating the results of topic modelling, ranging from subjective descriptions of their quality to heuristic mathematical measurements such as perplexity and entropy. This experiment uses subjective descriptions. During the experiment, Qiqqa is used to automatically generate topics for users' paper collections, and then to ask the users for their opinion on the quality of the topics.

5.5.2.1 Experiment

Figure 28 shows the Qiqqa user interface that is used to automatically generate the *Expedition Themes* (or topics in the nomenclature of this thesis) for a collection of papers. Exhibit 1 allows the user to select the number of themes to generate. This number defaults to the square-root of the number of papers in their library, a heuristic carried over from the literature where 200 topics is generally recommended for corpora of 40,000 papers. At Exhibit 2 they press the "Refresh Expedition" button to regenerate the themes for their library. First the AutoTags of the library are generated and associated with each paper in the library to form the bag-of-technical-terms representation for each paper. Next the topics are generated using the LDA bag-of-technical-terms document model described in Section 4.3.1. Once the topics are generated, they are shown in Exhibit 3. The topics are described by listing the technical terms that have the highest probability mass in each topic distribution. Each theme can be expanded, as illustrated in Exhibit 4, to show the papers whose topic distribution is most concentrated on the chosen theme. Exhibit 5 shows the user satisfaction evaluation. After generating their Themes, the user is prompted to judge positively ("thumbs up") or negatively ("thumbs down") the entire list of automatically generated Expedition Themes with the following instructions:

We are always trying to improve our algorithms. How do you rate the quality of the Expedition that Qiqqa generated for you?

The user is not obliged to participate in the user evaluation, so any results are a consequence of the user's voluntary choice to participate. 1,648 users participated.

5.5.2.2 Results and Discussion

Table 11 shows the results of the user satisfaction evaluation. 1,093 (66%) of the 1,648 users who participated in the evaluation give the automatically generated technical terms a "thumbs up", indicating that the automatically generated topics are useful to them.

To explore if there is a marked difference in satisfaction for large or small libraries (greater than or fewer than 50 PDFs – or 7 topics, respectively), the results are broken down further. 68% of users with a large library give the automatically generated technical terms a "thumbs up", while 62% of users with a small library do so. This suggests that topic models are more useful to users with a larger number of PDFs in their libraries, in analogy to the results from Section 5.5.1.

	All		Large		Small	
Thumbs down	555	(34%)	382	(32%)	173	(38%)
Thumbs up	1,093	(66%)	809	(68%)	284	(62%)
Total	1,648	(100%)	1,191	(100%)	457	(100%)

Table 11. Results of User Satisfaction Evaluation of Topic Modelling.

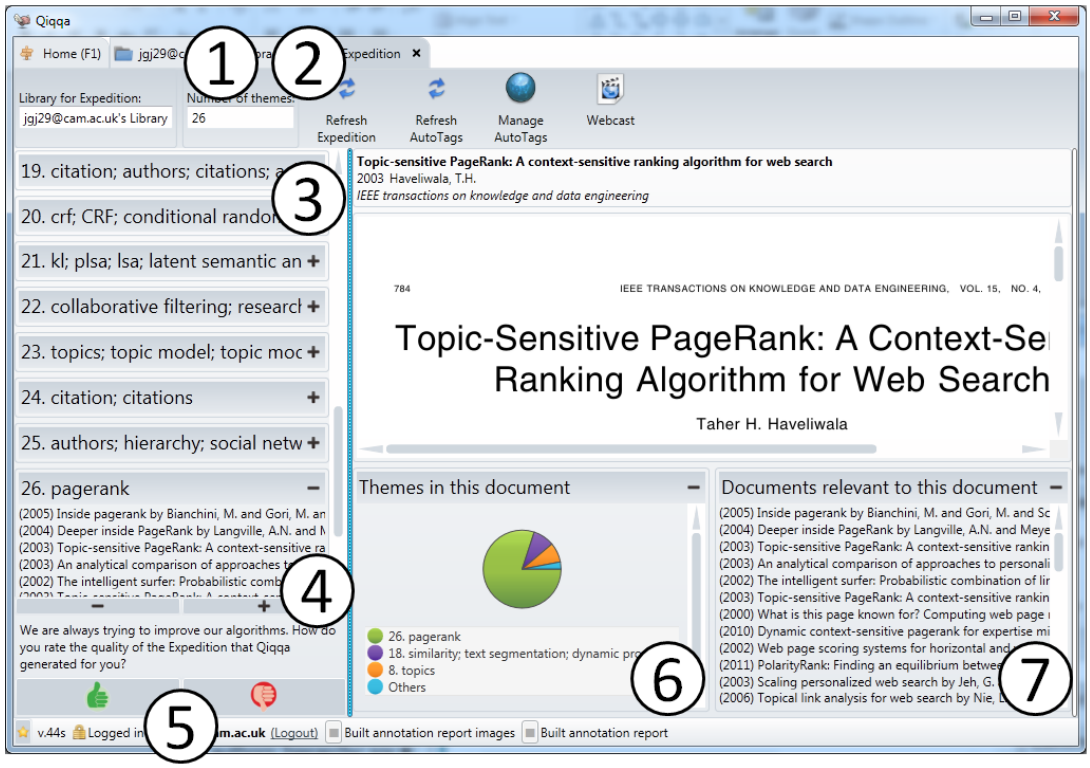


Figure 28. Screenshot of Qiqqa's Recommended Topics.

5.6 Summary

This chapter presented the evaluations performed in this thesis. Section 5.2 evaluated TPR in the task of automatically generating reading lists. This evaluation directly addressed the third research question of this thesis: *does lexical and social information contribute towards the task of automatically generating reading lists?* TPR significantly outperformed two state-of-the-art commercial search systems at this task. Section 5.3 evaluated TPR in the task of reference list reconstruction. TPR significantly outperformed a state-of-the-art system designed specifically for this task. In both experiments, TPR was also compared against the ablation system and baseline systems summarised in Section 5.1. The evaluation in Section 5.4 investigated the second research question of this thesis: *does the exposition of relationships between papers and their technical terms improves the performance of a novice in exploratory scientific search?* A task-based evaluation was used to compare how well novices were able to produce reading lists using TTLDA, a system that embeds the relationships between papers and their technical terms in its search results. Finally, Section 5.5 presented two simple user satisfaction evaluations that evaluated the quality of the technical terms and topics automatically generated from collections of PDFs.

Chapter 6.

Conclusion

This thesis addresses the task of automatically generating reading lists and the resulting primary output of this research is the ThemedPageRank (TPR) algorithm. My intuition behind TPR is that reading lists are different to generic information retrieval search results in two important ways. Reading lists need to be tailored to specific niches in science and they need to contain recommended papers that have authority in those niches.

TPR automatically finds niches in science using Automatic Term Recognition and Latent Topic Modelling and finds authority in these niches using a modified version of Personalised PageRank with Age Adjustment. This novel combination significantly improves on state-of-the-art at two different tasks.

The first task directly evaluates the automatic generation of reading lists for particular fields in science. System-generated reading lists are compared against gold-standard reading lists collected from experts specifically for this task. TPR significantly outperforms two state-of-the-art systems, Google Scholar and Google Index Search specialised to the ACL Anthology Network (AAN). It is also significantly outperforms a Lucene TFIDF baseline. While the results are statistically significant, an immediate avenue of further research would be to assemble a much larger corpus of gold-standard reading lists to strengthen these results. It would be useful to have additional reading lists from a wider range of topics, thus allowing the generalisation capabilities of the recommendation algorithms to be tested, and additional reading lists covering the same topics, thus reducing the potential subjectivity bias in the experts' paper choices. The task should be straightforward using the same instructions to the experts provided in this work.

The second task, Reference List Reconstruction, indirectly evaluates the automatic generation of reading lists in a much larger experiment. The proxy task is to reconstruct the bibliography section of thousands of scientific papers once all references have been redacted from the papers' text. TPR, which is unsupervised and uses only textual information, significantly outperforms a state-of-the-art system, purpose-built for this task, which relies on supervised machine learning and human-annotated data. While this task focusses only on regenerating the bibliography section, some research in the literature has tackled the harder task of regenerating the citations themselves in the running text. There are a variety of ways that TPR could be brought to bear on this harder task, and it would be an interesting line of further research to compare TPR to the current state-of-the-art systems.

Both tasks make use of the ACL Anthology Network (AAN) as the underlying corpus of scientific papers. It would make for important further research to examine the generality of TPR by performing these tasks on different corpora, especially those from other scientific domains such as chemistry or medicine (e.g., the TREC Genomics Corpus (Hersh et al. 2006)).

In both the reading list generation and reference list reconstruction tasks, a variety of ablation tests are performed to investigate the performance of the components of TPR. The tests show that on their own, traditional information retrieval mechanisms, such as TFIDF, Personalised PageRank and Latent Topic Modelling, do not perform particularly well at either task. The tests also highlight the fact that Citation Count, a bibliometric measurement widely relied upon by the academic community, does not perform well either. It is only their combination, such as is used in TPR, that produces results that better state-of-the-art. The field of bibliometrics might benefit from a deeper investigation of the applicability of TPR (rather than simple Citation Count or Impact Factor) to the attribution of authority and its influence on funding and grant allocation.

Another contribution of this thesis is the Citation Substitution Coefficient (CSC), an evaluation metric for evaluating the quality of reading lists. CSC is better suited to evaluating reading lists than standard IR metrics because it gives partial credit to recommended papers that are close to gold-standard papers in the citation graph, allowing the subtle differences in the performance of recommendation algorithms to be detected. Ideally, a new metric should undergo a series of calibration tests, comparing the ordering of CSC scores with those of more traditional IR metrics and potentially even calibrating CSC scores on a wide range of IR tasks against user acceptance scores. A detailed analysis was beyond the scope of my research, but does deserve deeper investigation.

From the performance of TPR in general, and from the ablation test results in particular, it is clear that topic modelling and technical terms are useful artefacts for the generation of reading lists. Two user satisfaction evaluations is Section 5.5 also confirm that automatically generated technical terms and topics are useful in general to two-thirds of the thousands of experiment participants.

The task-based evaluation in Section 5.4 investigates the notion that topic modelling and technical terms are useful artefacts specifically for exploratory scientific search. TTLDA, an early precursor to TPR that makes use of technical terms and topic modelling, performs similarly to a state-of-the-art scientific search system, Google Scholar (GS), in the task of novices performing exploratory scientific search. More interestingly, the results substantiate my claim that relationships induced by topic modelling over technical terms helps novices not only search with more non-obvious search queries, but also find more non-obvious papers relevant to their information need. An obvious avenue of further research would be to repeat the task based evaluation comparing GS to full-blown TPR, rather than TTLDA.

A final contribution of this thesis is a light-weight algorithm for Automatic Term Recognition (ATR). This light-weight algorithm extracts technical terms from the titles of documents without the need for the apparatus required by most state-of-the-art ATR algorithms. It is also capable of extracting very long technical terms, unlike many other ATR algorithms. While my ATR algorithm was adequate for the purposes of generating technical terms for TPR's document representation, comparison of my ATR algorithm to state-of-the-art ATR systems was outside the score of my research. In future, it would be worthwhile to investigate their relative performances. From a more technical perspective, it is clear from the results of all the experiments that documents modelled using topic distributions over technical terms is comparable to, if not better than, documents modelled using their full-text. This opens several potential lines of research into making Information Retrieval systems more scalable by using the more compact and efficient model of topic distributions over technical terms.

Bibliography

Agirre, E. & Soroa, A. (2009), Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. pp. 33–41.

Ananiadou, S. (1994), A methodology for automatic term recognition. *Proceedings of the 15th conference on Computational linguistics-Volume 2*, p.1038.

Andrzejewski, D. & Zhu, X. (2009), Latent Dirichlet Allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*. pp. 43–48.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T. & others (2000), Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1), p.25.

Asuncion, A., Smyth, P. & Welling, M. (2008), Asynchronous distributed learning of topic models. *Advances in Neural Information Processing Systems*, 21, pp.81–88.

Asuncion, A., Welling, M., Smyth, P. & Teh, Y.W. (2009), On smoothing and inference for topic models. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp.27–34.

Balog, K., Fang, Y., de Rijke, M., Serdyukov, P. & Si, L. (2012), Expertise Retrieval. *Foundations and Trends in Information Retrieval*, 6(2-3), pp.127–256.

Barrington, L., Oda, R. & Lanckriet, G.R. (2009), Smarter than Genius? Human Evaluation of Music Recommender Systems. In *ISMIR*. pp. 357–362.

Bazerman, C. (1985), Physicists reading physics. *Written Communication*, 2(1), p.3.

Bernstein, Y. & Zobel, J. (2005), Redundant documents and search effectiveness. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. pp. 736–743.

Bethard, S. & Jurafsky, D. (2010), Who should I cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. pp. 609–618.

Bharat, K. & Henzinger, M.R. (1998), Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 104–111.

Bharat, K. & Mihaila, G.A. (2001), When experts agree: Using non-affiliated experts to rank popular topics. In *Proceedings of the 10th international conference on World*

Wide Web. pp. 597–602.

Bird, S., Dale, R., Dorr, B.J., Gibson, B., Joseph, M.T., Kan, M.Y., Lee, D., Powley, B., Radev, D.R. & Tan, Y.F. (2008), The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. *Proc. of the 6th International Conference on Language Resources and Evaluation Conference (LREC08)*, pp.1755–1759.

Blei, D., Griffiths, T.L., Jordan, M.I. & Tenenbaum, J.B. (2004), Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems*, 16, p.106.

Blei, D.M. (2004), Probabilistic models of text and images. *PhD thesis, University of California at Berkeley*.

Blei, D.M. & Lafferty, J.D. (2007), A correlated topic model of science. *Annals*, 1(1), pp.17–35.

Blei, D.M. & Lafferty, J.D. (2006), Correlated Topic Models. *Advances in Neural Information Processing Systems 18: Proceedings of the 2005 Conference*, p.147.

Blei, D.M., Ng, A.Y. & Jordan, M.I. (2003), Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3, pp.993–1022.

Bogers, T. & Van Den Bosch, A. (2008), Recommending scientific articles using citeulike. *Proceedings of the 2008 ACM conference on Recommender systems*, pp.287–290.

Bonnin, G. & Jannach, D. (2013), A Comparison of Playlist Generation Strategies for Music Recommendation and a New Baseline Scheme. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Boole, G. (1848), The calculus of logic. *Cambridge and Dublin Mathematical Journal*, 3(1848), pp.183–198.

Boyack, K.W., Börner, K. & Klavans, R. (2009), Mapping the structure and evolution of chemistry research. *Scientometrics*, 79(1), pp.45–60.

Boyd-Graber, J., Blei, D. & Zhu, X. (2007), A topic model for word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. pp. 1024–1033.

Brin, S. & Page, L. (1998), The anatomy of a large-scale hypertextual Web search engine* 1. *Computer networks and ISDN systems*, 30(1-7), pp.107–117.

Buckley, C. & Voorhees, E.M. (2000), Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 33–40.

Buckley, C. & Voorhees, E.M. (2004), Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 25–32.

Burgin, R. (1992), Variations in relevance judgments and the evaluation of retrieval performance. *Information Processing \& Management*, 28(5), pp.619–627.

Castellvi, M., Bagot, R.E. & Palatresi, J.V. (2001), Automatic term detection: A review of current systems. *Recent Adv. in Comp. Terminology*.

Chambers, N. & Jurafsky, D. (2011), Template-Based Information Extraction without the Templates.

Chang, J. & Blei, D.M. (2009), Relational topic models for document networks. *Dvan Dyk, MWelling (eds.), AISTATS*, 9, pp.81–88.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C. & Blei, D. (2009a), Reading tea leaves: How humans interpret topic models. *Neural Information Processing Systems*.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C. & Blei, D. (2009b), Reading tea leaves: How humans interpret topic models. *Neural Information Processing Systems*.

Chemudugunta, C., Smyth, P. & Steyvers, M. (2007), Modeling general and specific aspects of documents with a probabilistic topic model. *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, p.241.

Chen, C. (2004), Searching for intellectual turning points: Progressive knowledge domain visualization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1), p.5303.

Chen, C. (1999), Visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing and Management*, 35(3), pp.401–420.

Chen, C.C., Yang, K.H., Kao, H.Y. & Ho, J.M. (2008), BibPro: A Citation parser based on sequence alignment techniques. *22nd International Conference on Advanced Information Networking and Applications-Workshops*, pp.1175–1180.

Chen, P., Xie, H., Maslov, S. & Redner, S. (2007), Finding scientific gems with Google's PageRank algorithm. *Journal of Infometrics*, 1(1), pp.8–15.

Christoffersen, M. (2004), Identifying core documents with a multiple evidence relevance filter. *Scientometrics*, 61(3), pp.385–394.

Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S. & MacKinnon, I. (2008), Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 659–666.

Cleverdon, C.W. (1960), *Report on the first stage of an investigation into the comparative efficiency of indexing systems*, College of Aeronautics.

Cleverdon, C.W., Mills, J. & Keen, M. (1966), Factors determining the performance of indexing systems.

Cohn, D. & Chang, H. (2000), Learning to probabilistically identify authoritative documents. *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pp.167–174.

Cohn, D. & Hofmann, T. (2001), The missing link-a probabilistic model of document content and hypertext connectivity. *Advances in neural information processing systems*, pp.430–436.

Councill, I.G., Giles, C.L. & Kan, M.Y. (2008), ParsCit: An open-source CRF reference string parsing package. *Proceedings of LREC*, 2008.

Van de Cruys, T., Poibeau, T. & Korhonen, A. (2011), Latent vector weighting for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pp. 1012–1022.

Daille, B. (1995), Combined approach for terminology extraction: lexical statistics and linguistic filtering.

Daille, B., Gaussier, É. & Langé, J.-M. (1994), Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*. pp. 515–521.

Daud, A. (2008), Scientific Recommendation through Topic Modeling.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. & Harshman, R. (1990), Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), pp.391–407.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977), Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp.1–38.

Dietz, L., Bickel, S. & Scheffer, T. (2007), Unsupervised prediction of citation influences. *Proceedings of the 24th international conference on Machine learning*, p.240.

Ding, Y. (2011), Topic-based PageRank on author cocitation networks. *Journal of the American Society for Information Science and Technology*, 62(3), pp.449–466.

Eales, J., Pinney, J., Stevens, R. & Robertson, D. (2008), Methodology capture: discriminating between the "best" and the rest of community practice. *BMC bioinformatics*, 9(1), p.359.

Eck, N., Waltman, L., Noyons, E. & Buter, R. (2008), Automatic Term Identification for Bibliometric Mapping. *Research Paper*.

Eisenstein, J., Chau, D.H., Kittur, A. & Xing, E.P. (2011), TopicScape: Semantic Navigation of Document Collections. *Arxiv preprint arXiv:1110.6200*.

Ekstrand, M.D., Kannan, P., Stemper, J.A., Butler, J.T., Konstan, J.A. & Riedl, J.T. (2010), Automatically building research reading lists. In *Proceedings of the fourth ACM conference on Recommender systems*. pp. 159–166.

El-Arini, K. & Guestrin, C. (2011), Beyond keyword search: discovering relevant scientific literature. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 439–447.

Ellis, S.R. & Hitchcock, R.J. (1986), The emergence of Zipf's law: Spontaneous encoding optimization by users of a command language. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(3), pp.423–427.

Elmqvist, N. & Tsigas, P. (2007), CiteWiz: a tool for the visualization of scientific citation networks. *Information Visualization*, 6(3), pp.215–232.

Erosheva, E., Fienberg, S. & Lafferty, J. (2004), Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1), p.5220.

Fairthorne, R. (2007), The patterns of retrieval. *American Documentation*, 7(2), pp.65–70.

Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. & Nevill-Manning, C.G. (1999), Domain-specific keyphrase extraction. *International Joint Conference on Artificial Intelligence*, 16, pp.668–673.

Frantzi, K.T. & Ananiadou, S. (1997), Automatic term recognition using contextual cues. *Proceedings of Mulsaic*, 97.

Fujii, A. (2007), Enhancing patent retrieval by citation analysis. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 793–794.

Garfield, E. (1972), Citation analysis as a tool in journal evaluation. *American Association for the Advancement of Science*.

Garfield, E. (1955), Citation indexes to science: a new dimension in documentation through association of ideas. *Science*, 122, pp.108–111.

Garfield, E. (1964), Science Citation Index: a new dimension in indexing. *Science*, 144(3619), pp.649–654.

Garfield, E. (2006), The history and meaning of the journal impact factor. *JAMA: the journal of the American Medical Association*, 295(1), p.90.

Gaussier, E. & Goutte, C. (2005), Relation between PLSA and NMF and implications. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.601–602.

Giles, C.L., Bollacker, K.D. & Lawrence, S. (1998), CiteSeer: An automatic citation indexing system. *Proceedings of the third ACM conference on Digital libraries*, pp.89–98.

Gipp, B. & Beel, J. (2009), Identifying Related Documents For Research Paper Recommender By CPA and COA. *International Conference on Education and Information Technology*, 1, pp.636–639.

Goffman, W. (1964), A searching procedure for information retrieval. *Information Storage and Retrieval*, 2(2), pp.73–78.

Goldberg, D., Nichols, D., Oki, B.M. & Terry, D. (1992), Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), pp.61–70.

Gori, M. & Pucci, A. (2006), Research paper recommender systems: A random-walk based approach. *IEEE Computer Society*.

Griffiths, T.L. & Steyvers, M. (2004), Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1), p.5228.

Gruber, A., Rosen-Zvi, M. & Weiss, Y. (2008), Latent topic models for hypertext. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*.

Haveliwala, T.H. (2003), Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, pp.784–796.

Haveliwala, T.H., Kamvar, S. & Jeh, G. (2003), An analytical comparison of approaches to personalizing PageRank. *Preprint, June*.

Havre, S., Hetzler, B. & Nowell, L. (2000), ThemeRiver: Visualizing theme changes over time. *Proceedings of the IEEE Symposium on Information Vizualization 2000*, p.115.

He, Q., Chen, B., Pei, J., Qiu, B., Mitra, P. & Giles, L. (2009), Detecting topic evolution in scientific literature: how can citations help? *Proceeding of the 18th ACM conference on Information and knowledge management*, pp.957–966.

He, Q., Kifer, D., Pei, J., Mitra, P. & Giles, C.L. (2011), Citation recommendation without author supervision. In *Proceedings of the fourth ACM international conference on Web search and data mining*. pp. 755–764.

He, Q., Pei, J., Kifer, D., Mitra, P. & Giles, L. (2010), Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*. pp. 421–430.

Heaps, H.S. (1978), *Information retrieval: Computational and theoretical aspects*, Academic Press, Inc.

Hearst, M.A. (2009), Search User Interfaces.

Hersh, W.R., Cohen, A.M., Roberts, P.M. & Rekapalli, H.K. (2006), TREC 2006 Genomics Track Overview. In *TREC*.

Hirsch, J.E. (2005), An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46), p.16569.

Hofmann, T. (1999), Probabilistic latent semantic analysis. *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, pp.289–296.

Jeh, G. & Widom, J. (2003), Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*. pp. 271–279.

Justeson, J.S. & Katz, S.M. (1995), Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(01), pp.9–27.

Kageura, K. & Umino, B. (1996), Methods of automatic term recognition: A review. *Terminology*, 3(2), pp.259–289.

Kataria, S., Mitra, P. & Bhatia, S. (2010), Utilizing context in generative bayesian models for linked corpus. In *AAAI*.

Kelly, D. (2009), Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1—2), pp.1–224.

Kessler, M.M. (1963), Bibliographic coupling between scientific papers. *American Documentation*, 14(1), pp.10–25.

Kim, S.N., Baldwin, T. & Kan, M.Y. (2009), An Unsupervised Approach to Domain-Specific Term Extraction.

Kim, S.N., Medelyan, O., Kan, M.Y. & Baldwin, T. (2010), SEMEVAL-2010 Task 5: Automatic keyphrase extraction from scientific articles. *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp.21–26.

Kircz, J.G. (1991), Rhetorical structure of scientific articles: the case for argumentational analysis in information retrieval. *Journal of Documentation*, 47(4), pp.354–372.

Klavans, R. & Boyack, K.W. (2006), Identifying a better measure of relatedness for mapping science. *Journal of the American Society for Information Science and*

Technology, 57(2), pp.251–263.

Kleinberg, J.M. (1999), Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5), pp.604–632.

Kohavi, R., Longbotham, R., Sommerfield, D. & Henne, R.M. (2009), Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1), pp.140–181.

Kostoff, R.N. (1998), The use and misuse of citation analysis in research evaluation. *Scientometrics*, 43(1), pp.27–43.

Larsen, P.O. & von Ins, M. (2009), The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, pp.1–29.

Lee, B., Czerwinski, M., Robertson, G. & Bederson, B.B. (2005), Understanding research trends in conferences using PaperLens. *CHI'05 extended abstracts on Human factors in computing systems*, p.1972.

Lee, D.D. & Seung, H.S. (2001), Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

Lee, W.C. & Fox, E.A. (1988), Experimental comparison of schemes for interpreting Boolean queries.

Lempel, R. & Moran, S. (2000), The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1), pp.387–401.

Li, W. & McCallum, A. (2006), Pachinko allocation: DAG-structured mixture models of topic correlations. *Proceedings of the 23rd international conference on Machine learning*, pp.577–584.

Liben-Nowell, D. & Kleinberg, J. (2007), The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7), pp.1019–1031.

Lin, C.J. (2007), On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 18(6), pp.1589–1596.

Lin, J. (2002), Divergence measures based on the Shannon entropy. *Information Theory, IEEE Transactions on*, 37(1), pp.145–151.

Litvak, M. & Last, M. (2008), Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*. pp. 17–24.

Liu, X. & Croft, W.B. (2004), Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research*

and development in information retrieval. pp. 186–193.

Liu, Z., Zhang, Y., Chang, E.Y. & Sun, M. (2011), PLDA+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), p.26.

Lopez, P. & Romary, L. (2010), HUMB: Automatic Key Term Extraction from Scientific Articles in GROBID. *SemEval 2010 Workshop*.

Lu, Y., He, J., Shan, D. & Yan, H. (2011), Recommending citations with translation model. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. pp. 2017–2020.

Ma, N., Guan, J. & Zhao, Y. (2008), Bringing PageRank to the citation analysis. *Information Processing \& Management*, 44(2), pp.800–810.

MacKay, D.J.C. (2003), *Information theory, inference and learning algorithms*, Cambridge university press.

MacRoberts, M.H. & MacRoberts, B.R. (1996), Problems of citation analysis. *Scientometrics*, 36(3), pp.435–444.

Mann, G.S., Mimno, D. & McCallum, A. (2006), Bibliometric impact measures leveraging topic analysis. *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, p.74.

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

Maron, M. & Kuhns, J. (1960), On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3), pp.216–244.

Maslov, S. & Redner, S. (2008), Promise and pitfalls of extending Google's PageRank algorithm to citation networks. *The Journal of Neuroscience*, 28(44), p.11103.

Matsuo, Y. & Ishizuka, M. (2004), Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1), pp.157–170.

McCallum, A., Corrada-Emmanuel, A. & Wang, X. (2005), The Author-Recipient-Topic Model for Topic and Role Discovery in Social Networks: Experiments with Enron and Academic Email.

McCallum, A., Wang, X. & Corrada-Emmanuel, A. (2007), Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1), pp.249–272.

McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A. & Riedl, J. (2002), On the recommending of citations for research papers. *Proceedings of the 2002 ACM conference on Computer supported*

cooperative work, pp.116–125.

Meij, E. & De Rijke, M. (2007), Using prior information derived from citations in literature search. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*. pp. 665–670.

Mihalcea, R. & Tarau, P. (2004), TextRank: Bringing order into texts. In *Proceedings of EMNLP*. pp. 404–411.

Miller, G.A. (1995), WordNet: a lexical database for English. *Communications of the ACM*, 38(11), pp.39–41.

Mimno, D. & Blei, D. (2011), Bayesian checking for topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pp. 227–237.

Minka, T. & Lafferty, J. (2002), Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. pp. 352–359.

Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishan, P., Qazvinian, V., Radev, D. & Zajic, D. (2009a), Generating Surveys of Scientific Paradigms. *Proceedings of HLT-NAACL*.

Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishan, P., Qazvinian, V., Radev, D. & Zajic, D. (2009b), Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. pp. 584–592.

Mooers, C.N. (1950), Information retrieval viewed as temporal signaling. In *Proceedings of the International Congress of Mathematicians*. pp. 572–573.

Moore, J.L., Chen, S., Joachims, T. & Turnbull, D. (2012), Learning to Embed Songs and Tags for Playlist Prediction. In *ISMIR*. pp. 349–354.

Nallapati, R. & Cohen, W. (2008), Link-PLSA-LDA: A new unsupervised model for topics and influence of blogs. In *International Conference for Weblogs and Social Media*.

Nallapati, R., Cohen, W. & Lafferty, J. (2007), Parallelized variational EM for latent Dirichlet allocation: An experimental evaluation of speed and scalability. *icdmw*, pp.349–354.

Nanba, H., Kando, N., Okumura, M. & others (2004), Classification of research papers using citation links and citation types: Towards automatic review article generation. *Proceedings of the 11th SIG Classification Research Workshop*, pp.117–134.

Nanba, H. & Okumura, M. (2005), Automatic detection of survey articles. *Research and Advanced Technology for Digital Libraries*, pp.391–401.

Narayan, B., Murthy, C. & Pal, S.K. (2003), Topic continuity for web document categorization and ranking. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*. pp. 310–315.

Newman, D., Asuncion, A., Smyth, P. & Welling, M. (2007), Distributed inference for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems*, 20(1081-1088), pp.17–24.

Newman, D., Smyth, P. & Steyvers, M. (2006), Scalable Parallel Topic Models. *Journal of Intelligence Community Research and Development*.

Nguyen, Q.V., Huang, M.L. & Simoff, S. (2007), Visualization of relational structure among scientific articles. *Proceedings of the 9th international conference on Advances in visual information systems*, pp.415–425.

Nguyen, T.D. & Luong, M.T. (2010), WINGNUS: Keyphrase Extraction Utilizing Document Logical Structure.

Nie, L., Davison, B.D. & Qi, X. (2006), Topical link analysis for web search. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 91–98.

Noel, S., Chu, C.H. & Raghavan, V. (2002), Visualization of document co-citation counts. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*. pp. 691–696.

Oddy, R.N., Liddy, E.D.R., Balakrishnan, B., Bishop, A., Elewononi, J. & Martin, E. (1992), Towards the use of situational information in information retrieval. *Journal of Documentation*, 48(2), pp.123–171.

Page, L., Brin, S., Motwani, R. & Winograd, T. (1998), The PageRank citation ranking: Bringing order to the web. *Stanford Digital Library Technologies Project*.

Pal, S.K. & Narayan, B. (2005), A web surfer model incorporating topic continuity. *Knowledge and Data Engineering, IEEE Transactions on*, 17(5), pp.726–729.

Park, Y., Byrd, R.J. & Boguraev, B.K. (2002), Automatic glossary extraction: beyond terminology identification. *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp.1–7.

Pauca, V.P., Shahnaz, F., Berry, M.W. & Plemmons, R.J. (2004), Text mining using nonnegative matrix factorizations. *Proc. SIAM Inter. Conf. on Data Mining, Orlando, FL*.

Peeters, G., Urbano, J. & Jones, G.J. (2012), Notes from the ISMIR 2012 late-breaking session on evaluation in music information retrieval.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P. & Welling, M. (2008), Fast collapsed gibbs sampling for latent dirichlet allocation. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.569–577.

Porter, M.F. (1980), An algorithm for suffix stripping. *Program*, 14(3), pp.130–137.

Qazvinian, V. & Radev, D.R. (2008), Scientific paper summarization using citation summary networks. *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp.689–696.

Radev, D.R., Joseph, M.T., Gibson, B. & Muthukrishnan, P. (2009a), A Bibliometric and Network Analysis of the field of Computational Linguistics. *Ann Arbor*, 1001, pp.48109–1092.

Radev, D.R., Muthukrishnan, P. & Qazvinian, V. (2009b), The ACL Anthology Network Corpus. In *Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*. Singapore.

Ramage, D., Hall, D., Nallapati, R. & Manning, C.D. (2009), Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. pp. 248–256.

Renear, A.H. & Palmer, C.L. (2009), Strategic reading, ontologies, and the future of scientific publishing. *Science*, 325(5942), pp.828–832.

Richardson, M. & Domingos, P. (2002), The intelligent surfer: Probabilistic combination of link and content information in pagerank. *Advances in neural information processing systems*, 14, pp.1441–1448.

Van Rijsbergen, C.J. (1979), *Information Retrieval*, Butterworth-Heinemann, Newton, MA.

Van Rijsbergen, C.J. (1975), *Information Retrieval*, Butterworths. Available at: http://books.google.co.uk/books?id=EJ2PQgAACAAJ.

Ritchie, A. (2009), *Citation context analysis for information retrieval*.

Ritchie, A., Teufel, S. & Robertson, S. (2006), How to find better index terms through citations. *Proceedings of the workshop on how can computational linguistics improve information retrieval*, pp.25–32.

Robertson, S.E. (1977), The probability ranking principle in IR. *Journal of documentation*, 33(4), pp.294–304.

Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P. & Steyvers, M. (2010), Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1), pp.1–38.

Rosen-Zvi, M., Griffiths, T., Steyvers, M. & Smyth, P. (2004), The author-topic model for authors and documents. *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp.487–494.

Sager, J.C., Dungworth, D. & McDonald, P.F. (1980), *English special languages: principles and practice in science and technology*, Brandstetter Wiesbaden.

Salton, G. (1992), The state of retrieval system evaluation. *Information processing and management*, 28(4), pp.441–449.

Salton, G. & Buckley, C. (1988), Term-weighting approaches in automatic text retrieval. *Information processing \& management*, 24(5), pp.513–523.

Salton, G., Wong, A. & Yang, C.S. (1975), A vector space model for automatic indexing. *Communications of the ACM*, 18(11), pp.613–620.

Sanderson, M. (2010), Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*, 4(4), pp.247–375.

Schäfer, U. & Kasterka, U. (2010), Scientific authoring support: a tool to navigate in typed citation graphs. *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pp.7–14.

Schneider, J.W. (2005), Naming clusters in visualization studies: parsing and filtering of noun phrases from citation contexts. In *Proceedings of 10th International Conference of the International Society for Scientometrics and Informetrics (ISSI 2005), Sweden*. pp. 406–416.

Schwarz, A.W., Schwarz, S. & Tijssen, R. (1998), Research and research impact of a technical university—A bibliometric study. *Scientometrics*, 41(3), pp.371–388.

Shahaf, D., Guestrin, C. & Horvitz, E. (2012), Metro maps of science. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 1122–1130.

Shahnaz, F., Berry, M.W., Pauca, V.P. & Plemmons, R.J. (2006), Document clustering using nonnegative matrix factorization. *Information Processing \& Management*, 42(2), pp.373–386.

Shaparenko, B. & Joachims, T. (2007), Information genealogy: Uncovering the flow of ideas in non-hyperlinked document databases. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 619–628.

Shum, S.B. (1998), Evolving the Web for Scientific Knowledge: First Steps towards an HCI Knowledge Web. *Interfaces*, pp.16–21.

Small, H. (1973), Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American society for*

information science, 24(4), pp.265–269.

de Solla Price, D.J. (1986), *Little science, big science... and beyond*, Columbia University Press, New York.

Sparck-Jones, K. (1972), A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1), pp.11–21.

Sparck-Jones, K. & Needham, R.M. (1968), Automatic term classifications and retrieval. *Information Storage and Retrieval*, 4(2), pp.91–100.

Steyvers, M. & Griffiths, T. (2007), Probabilistic topic models. *Handbook of latent semantic analysis*, 427.

Steyvers, M., Smyth, P., Rosen-Zvi, M. & Griffiths, T. (2004), Probabilistic author-topic models for information discovery. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.306–315.

Strohman, T., Croft, W.B. & Jensen, D. (2007), Recommending citations for academic papers. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.705–706.

Sun, C., Gao, B., Cao, Z. & Li, H. (2008), HTM: A topic model for hypertexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pp. 514–522.

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

Tang, J., Jin, R. & Zhang, J. (2008a), A topic modeling approach and its integration into the random walk framework for academic search. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. pp. 1055–1060.

Tang, J. & Zhang, J. (2009), A discriminative approach to Topic-Based citation recommendation. *Advances in Knowledge Discovery and Data Mining*, pp.572–579.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L. & Su, Z. (2008b), Arnetminer: Extraction and mining of academic social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 990–998.

Tang, Y. (2008), *The design and study of pedagogical paper recommendation*.

Taube, M. & Wooster, H. (1958), *Information storage and retrieval theory, systems, and devices*, Columbia University Press.

Torres, R., McNee, S.M., Abel, M., Konstan, J.A. & Riedl, J. (2004), Enhancing digital libraries with TechLens+. *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pp.228–236.

Toutanova, K. & Johnson, M. (2007), A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of NIPS*.

Treeratpituk, P., Teregowda, P., Huang, J. & Giles, C.L. (2010), SEERLAB: A system for extracting key phrases from scholarly documents. *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp.182–185.

Voorhees, E.M. (2000), Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36(5), pp.697–716.

Voorhees, E.M., Harman, D.K. & others (2005), *TREC: Experiment and evaluation in information retrieval*, MIT Press Cambridge, MA.

Walker, D., Xie, H., Yan, K.K. & Maslov, S. (2007), Ranking scientific publications using a model of network traffic. *Journal of Statistical Mechanics: Theory and Experiment*, 2007, p.P06010.

Wallach, H., Mimno, D. & McCallum, A. (2009a), Rethinking LDA: Why priors matter. *Proceedings of NIPS-09, Vancouver, BC*.

Wallach, H.M. (2002), Structured topic models for language. *Unpublished doctoral dissertation, University of Cambridge*.

Wallach, H.M. (2006), Topic modeling: beyond bag-of-words. *Proceedings of the 23rd international conference on Machine learning*, pp.977–984.

Wallach, H.M., Murray, I., Salakhutdinov, R. & Mimno, D. (2009b), Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*. pp. 1105–1112.

Wang, A. (2006), The Shazam music recognition service. *Communications of the ACM*, 49(8), pp.44–48.

Wang, C. & Blei, D.M. (2011), Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 448–456.

Wang, D., Zhu, S., Li, T. & Gong, Y. (2009a), Multi-document summarization using sentence-based topic models. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp.297–300.

Wang, X. & McCallum, A. (2005), A note on topical n-grams. *University of Massachusetts Technical Report UM-CS-2005-071*.

Wang, X., McCallum, A. & Wei, X. (2007), Topical n-grams: Phrase and topic discovery, with an application to information retrieval. *Proceedings of the 7th IEEE international conference on data mining*, pp.697–702.

Wang, Y., Bai, H., Stanton, M., Chen, W.Y. & Chang, E. (2009b), Plda: Parallel latent dirichlet allocation for large-scale applications. *Algorithmic Aspects in Information*

and Management, pp.301–314.

Wang, Y., Zhai, E., Hu, J. & Chen, Z. (2010), Claper: Recommend classical papers to beginners. *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, 6, pp.2777–2781.

Wei, X. & Croft, W.B. (2006), LDA-based document models for ad-hoc retrieval. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.178–185.

White, H.D. & McCain, K.W. (1998), Visualizing a discipline: An author co-citation analysis of information science, 1972-1995. *Journal of the American Society for Information Science*, 49(4), pp.327–355.

Wilson, A.T. & Chew, P.A. (2010), Term Weighting Schemes for Latent Dirichlet Allocation. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp.465–473.

Wissner-Gross, A.D. (2006), Preparation of topical reading lists from the link structure of Wikipedia. In *Advanced Learning Technologies, 2006. Sixth International Conference on*. pp. 825–829.

Wong, W., Liu, W. & Bennamoun, M. (2009), A probabilistic framework for automatic term recognition. *Intelligent Data Analysis*, 13(4), pp.499–539.

Woodruff, A., Gossweiler, R., Pitkow, J., Chi, E.H. & Card, S.K. (2000), Enhancing a digital book with a reading recommender. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.153–160.

Wu, B., Goel, V. & Davison, B.D. (2006), Topical TrustRank: Using topicality to combat web spam. In *Proceedings of the 15th international conference on World Wide Web*. pp. 63–72.

Xu, W., Liu, X. & Gong, Y. (2003), Document clustering based on non-negative matrix factorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp.267–273.

Yang, Z., Tang, J., Zhang, J., Li, J. & Gao, B. (2009), Topic-level random walk through probabilistic model. *Advances in Data and Web Management*, pp.162–173.

Zesch, T. & Gurevych, I. (2009), Approximate matching for evaluating keyphrase extraction. In *Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing*. pp. 484–489.

Zhai, C.X., Cohen, W.W. & Lafferty, J. (2003), Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. pp. 10–17.

Zhao, W., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H. & Li, X. (2011), Comparing Twitter and traditional media using topic models. *Advances in Information Retrieval*, pp.338–349.

Zhou, D., Zhu, S., Yu, K., Song, X., Tseng, B.L., Zha, H. & Giles, C.L. (2008), Learning multiple graphs for document recommendations. In *Proceeding of the 17th international conference on World Wide Web*. pp. 141–150.

Zipf, G. (1949), Selective Studies and the Principle of Relative Frequency in Language (Cambridge, Mass, 1932). *Human Behavior and the Principle of Least-Effort (Cambridge, Mass*.

Appendix A.

Gold-Standard Reading Lists

"concept-to-text generation"

H05-1042: Collective Content Selection For Concept-To-Text Generation
Barzilay, Regina; Lapata, Mirella
2005 Human Language Technology Conference And Empirical Methods In Natural Language Processing

D09-1005: First- and Second-Order Expectation Semirings with Applications to Minimum-Risk Training on Translation Forests
Li, Zhifei; Eisner, Jason M.
2009 EMNLP

W98-1411: Experiments Using Stochastic Search For Text Planning
Mellish, Chris S.; Knott, Alistair; Oberlander, Jon; O'Donnell, Michael
1998 Workshop On Natural Language Generation EWNLG

P04-1011: Trainable Sentence Planning For Complex Information Presentations In Spoken Dialog Systems
Stent, Amanda J.; Prassad, Rashmi; Walker, Marilyn A.
2004 Annual Meeting Of The Association For Computational Linguistics

P95-1034: Two-Level, Many-Paths Generation
Knight, Kevin; Hatzivassiloglou, Vasileios
1995 Annual Meeting Of The Association For Computational Linguistics

P09-1011: Learning Semantic Correspondences with Less Supervision
Liang, Percy; Jordan, Michael I.; Klein, Dan
2009 ACL-IJCNLP

W05-1506: Better K-Best Parsing
Huang, Liang; Chiang, David
2005 International Workshop On Parsing Technology

P07-1019: Forest Rescoring: Faster Decoding with Integrated Language Models
Huang, Liang; Chiang, David
2007 45th Annual Meeting of the Association of Computational Linguistics

J07-2003: Hierarchical Phrase-Based Translation
Chiang, David
2007 Computational Linguistics

J93-2003: The Mathematics Of Statistical Machine Translation: Parameter Estimation
Brown, Peter F.; Della Pietra, Vincent J.; Della Pietra, Stephen A.; Mercer, Robert L.
1993 Computational Linguistics

P02-1040: Bleu: A Method For Automatic Evaluation Of Machine Translation
Papineni, Kishore; Roukos, Salim; Ward, Todd; Zhu, Wei-Jing
2002 Annual Meeting Of The Association For Computational Linguistics

N07-1022: Generation by Inverting a Semantic Parser that Uses Statistical Machine Translation
Wong, Yuk Wah; Mooney, Raymond J.
2007 Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference

A00-2026: Trainable Methods For Surface Natural Language Generation
Ratnaparkhi, Adwait
2000 Applied Natural Language Processing Conference And Meeting Of The North American Association For Computational Linguistics

P00-1041: Headline Generation Based On Statistical Translation
Banko, Michele; Mittal, Vibhu O.; Witbrock, Michael J.
2000 Annual Meeting Of The Association For Computational Linguistics

C10-2062: Generative Alignment and Semantic Parsing for Learning from Ambiguous Supervision
Kim, Joohyun; Mooney, Raymond J.
2010 COLING - POSTERS

D08-1082: A Generative Model for Parsing Natural Language to Meaning Representations
Lu, Wei; Ng, Hwee Tou; Lee, Wee Sun; Zettlemoyer, Luke
2008 Conference On Empirical Methods In Natural Language Processing

"distributional semantics"

P93-1024: Distributional Clustering Of English Words
Pereira, Fernando C.N.; Tishby, Naftali; Lee, Lillian
1993 Annual Meeting Of The Association For Computational Linguistics

P99-1004: Measures Of Distributional Similarity
Lee, Lillian
1999 Annual Meeting Of The Association For Computational Linguistics

C04-1146: Characterising Measures Of Lexical Distributional Similarity
Weeds, Julie; Weir, David J.; McCarthy, Diana
2004 International Conference On Computational Linguistics

J07-2002: Dependency-Based Construction of Semantic Space Models
Padó, Sebastian; Lapata, Mirella

2007 Computational Linguistics

P02-1030: Scaling Context Space
Curran, James R.; Moens, Marc
2002 Annual Meeting Of The Association For Computational Linguistics

P98-2127: Automatic Retrieval and Clustering of Similar Words
Lin, Dekang
1998 COLING-ACL

P08-1068: Simple Semi-supervised Dependency Parsing
Koo, Terry; Carreras, Xavier; Collins, Michael John
2008 Annual Meeting Of The Association For Computational Linguistics

D09-1098: Web-Scale Distributional Similarity and Entity Set Expansion
Pantel, Patrick; Crestan, Eric; Borkovsky, Arkady; Popescu, Ana-Maria; Vyas, Vishnu
2009 EMNLP

J01-3003: Automatic Verb Classification Based On Statistical Distributions Of Argument Structure
Merlo, Paola; Stevenson, Suzanne
2001 Computational Linguistics

P04-1036: Finding Predominant Word Senses In Untagged Text
McCarthy, Diana; Koeling, Rob; Weeds, Julie; Carroll, John A.
2004 Annual Meeting Of The Association For Computational Linguistics

P07-1028: A Simple, Similarity-based Model for Selectional Preferences
Erk, Katrin
2007 45th Annual Meeting of the Association of Computational Linguistics

J98-1004: Automatic Word Sense Discrimination
Schütze, Hinrich
1998 Computational Linguistics

D08-1094: A Structured Vector Space Model for Word Meaning in Context
Erk, Katrin; Padó, Sebastian
2008 Conference On Empirical Methods In Natural Language Processing

P08-1028: Vector-based Models of Semantic Composition
Mitchell, Jeff; Lapata, Mirella
2008 Annual Meeting Of The Association For Computational Linguistics

“domain adaptation”

W01-0521: Corpus Variation And Parser Performance
Gildea, Daniel
2001 SIGDAT Conference On Empirical Methods In Natural Language Processing

P07-1033: Frustratingly Easy Domain Adaptation

Daumé III, Hal
2007 45th Annual Meeting of the Association of Computational Linguistics

W06-1615: Domain Adaptation With Structural Correspondence Learning
Blitzer, John; McDonald, Ryan; Pereira, Fernando C.N.
2006 Conference On Empirical Methods In Natural Language Processing

W08-1302: Exploring an Auxiliary Distribution Based Approach to Domain Adaptation of a Syntactic Disambiguation Model
Plank, Barbara; van Noord, Gertjan
2008 Coling 2008: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing

N04-4006: Language Model Adaptation With Map Estimation And The Perceptron Algorithm
Bacchiani, Michiel; Roark, Brian; Saraçlar, Murat
2004 Human Language Technology Conference And Meeting Of The North American Association For Computational Linguistics - Short Papers

W07-2202: Evaluating Impact of Re-training a Lexical Disambiguation Model on Domain Adaptation of an HPSG Parser
Hara, Tadayoshi; Miyao, Yusuke; Tsujii, Jun'ichi
2007 Tenth International Conference on Parsing Technologies

P06-1043: Reranking And Self-Training For Parser Adaptation
McClosky, David; Charniak, Eugene; Johnson, Mark
2006 International Conference On Computational Linguistics And Annual Meeting Of The Association For Computational Linguistics

D08-1050: Adapting a Lexicalized-Grammar Parser to Contrasting Domains
Rimell, Laura; Clark, Stephen
2008 Conference On Empirical Methods In Natural Language Processing

D07-1112: Frustratingly Hard Domain Adaptation for Dependency Parsing
Dredze, Mark; Blitzer, John; Talukdar, Partha Pratim; Ganchev, Kuzman; Graça, João V.; Pereira, Fernando C.N.
2007 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)

P07-1034: Instance Weighting for Domain Adaptation in NLP
Jiang, Jing; Zhai, ChengXiang
2007 45th Annual Meeting of the Association of Computational Linguistics

P07-1056: Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification
Blitzer, John; Dredze, Mark; Pereira, Fernando C.N.
2007 45th Annual Meeting of the Association of Computational Linguistics

"information extraction"

D09-1001: Unsupervised Semantic Parsing
Poon, Hoifung; Domingos, Pedro
2009 EMNLP

P09-1113: Distant supervision for relation extraction without labeled data
Mintz, Mike; Bills, Steven; Snow, Rion; Jurafsky, Daniel
2009 ACL-IJCNLP

P07-1107: Unsupervised Coreference Resolution in a Nonparametric Bayesian Model
Haghighi, Aria; Klein, Dan
2007 45th Annual Meeting of the Association of Computational Linguistics

D09-1120: Simple Coreference Resolution with Rich Syntactic and Semantic Features
Haghighi, Aria; Klein, Dan
2009 EMNLP

D08-1112: An Analysis of Active Learning Strategies for Sequence Labeling Tasks
Settles, Burr; Craven, Mark
2008 Conference On Empirical Methods In Natural Language Processing

N04-4028: Confidence Estimation For Information Extraction
Culotta, Aron; McCallum, Andrew
2004 Human Language Technology Conference And Meeting Of The North American Association For Computational Linguistics - Short Papers

P08-1090: Unsupervised Learning of Narrative Event Chains
Chambers, Nathanael; Jurafsky, Daniel
2008 Annual Meeting Of The Association For Computational Linguistics

N07-4013: TextRunner: Open Information Extraction on the Web
Yates, Alexander; Banko, Michele; Broadhead, Matthew; Cafarella, Michael J.; Etzioni, Oren; Soderland, Stephen
2007 Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)

P04-1054: Dependency Tree Kernels For Relation Extraction
Culotta, Aron; Sorensen, Jeffrey S.
2004 Annual Meeting Of The Association For Computational Linguistics

"lexical semantics"

P98-1013: The Berkeley FrameNet Project
Baker, Collin F.; Fillmore, Charles J.; Lowe, John B.
1998 COLING-ACL

W04-2604: Using Prepositions To Extend A Verb Lexicon
Kipper, Karin Christine; Snyder, Benjamin; Palmer, Martha Stone
2004 Computational Lexical Semantics Workshop

P03-2030: The FrameNet Data And Software
Baker, Collin F.; Sato, Hiroaki
2003 Annual Meeting Of The Association For Computational Linguistics - Interactive Posters And Demonstrations

J05-1004: The Proposition Bank: An Annotated Corpus Of Semantic Roles
Palmer, Martha Stone; Gildea, Daniel; Kingsbury, Paul
2005 Computational Linguistics

N06-2015: OntoNotes: The 90% Solution
Hovy, Eduard H.; Marcus, Mitchell P.; Palmer, Martha Stone; Ramshaw, Lance A.; Weischedel, Ralph M.
2006 Human Language Technology Conference And Meeting Of The North American Association For Computational Linguistics - Short Papers

N07-1071: ISP: Learning Inferential Selectional Preferences
Pantel, Patrick; Bhagat, Rahul; Coppola, Bonaventura; Chklovski, Timothy; Hovy, Eduard H.
2007 Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference

N07-1069: Can Semantic Roles Generalize Across Genres?
Yi, Szuting; Loper, Edward; Palmer, Martha Stone
2007 Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference

W04-2807: Different Sense Granularities For Different Applications
Palmer, Martha Stone; Babko-Malaya, Olga; Dang, Hoa Trang
2004 International Workshop On Scalable Natural Language Understanding ScaNaLU

J02-3001: Automatic Labeling Of Semantic Roles
Gildea, Daniel; Jurafsky, Daniel
2002 Computational Linguistics

W05-0620: Introduction To The CoNLL-2005 Shared Task: Semantic Role Labeling
Carreras, Xavier; Màrquez, Lluís
2005 Conference On Computational Natural Language Learning CoNLL

W05-0625: Generalized Inference With Multiple Semantic Role Labeling Systems
Koomen, Peter; Punyakanok, Vasin; Roth, Dan; Yih, Scott Wen-Tau
2005 Conference On Computational Natural Language Learning CoNLL

W05-0623: A Joint Model For Semantic Role Labeling
Haghighi, Aria; Toutanova, Kristina; Manning, Christopher D.

2005 Conference On Computational Natural Language Learning CoNLL

N06-1017: Unknown Word Sense Detection As Outlier Detection
Erk, Katrin
2006 Human Language Technology Conference And Meeting Of The North American Association For Computational Linguistics

P09-2019: Generalizing over Lexical Features: Selectional Preferences for Semantic Role Classification
Zapirain, Beñat; Agirre, Eneko; Màrquez, Lluís
2009 ACL-IJCNLP: Short Papers

"parser evaluation"

H91-1060: A Procedure For Quantitatively Comparing The Syntactic Coverage Of English Grammars
Black, Ezra W.; Abney, Steven P.; Flickinger, Daniel P.; Gdaniec, Claudia; Grishman, Ralph; Harrison, Philip; Hindle, Donald; Ingria, Robert J. P.; Jelinek, Frederick; Klavans, Judith L.; Liberman, Mark Y.; Marcus, Mitchell P.; Roukos, Salim; Santorini, Beatrice; Strzalkowski, Tomek
1991 Workshop On Speech And Natural Language

P06-2006: Evaluating The Accuracy Of An Unlexicalized Statistical Parser On The PARC DepBank
Briscoe, Ted; Carroll, John A.
2006 International Conference On Computational Linguistics And Annual Meeting Of The Association For Computational Linguistics - Poster Sessions

D09-1085: Unbounded Dependency Recovery for Parser Evaluation
Rimell, Laura; Clark, Stephen; Steedman, Mark
2009 EMNLP

W08-1307: Constructing a Parser Evaluation Scheme
Rimell, Laura; Clark, Stephen
2008 Coling 2008: TextGraphs Workshop On Graph Based Methods For Natural Language Processing

"statistical machine translation models"

P08-1024: A Discriminative Latent Variable Model for Statistical Machine Translation
Blunsom, Philip; Cohn, Trevor; Osborne, Miles
2008 Annual Meeting Of The Association For Computational Linguistics

D08-1023: Probabilistic Inference for Machine Translation
Blunsom, Philip; Osborne, Miles
2008 Conference On Empirical Methods In Natural Language Processing

P05-1033: A Hierarchical Phrase-Based Model For Statistical Machine Translation
Chiang, David

2005 Annual Meeting Of The Association For Computational Linguistics

N04-1035: What's In A Translation Rule?
Galley, Michel; Hopkins, Mark; Knight, Kevin; Marcu, Daniel
2004 Human Language Technology Conference And Meeting Of The North American Association For Computational Linguistics

N09-1025: 11,001 New Features for Statistical Machine Translation
Chiang, David; Knight, Kevin; Wang, Wei
2009 -NAACL

"statistical parsing"

J03-4003: Head-Driven Statistical Models For Natural Language Parsing
Collins, Michael John
2003 Computational Linguistics

J07-4004: Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models
Clark, Stephen; Curran, James R.
2007 Computational Linguistics

P02-1035: Parsing The Wall Street Journal Using A Lexical-Functional Grammar And Discriminative Estimation Techniques
Riezler, Stefan; King, Tracy Holloway; Kaplan, Ronald M.; Crouch, Richard; Maxwell III, John T.; Johnson, Mark
2002 Annual Meeting Of The Association For Computational Linguistics

P95-1037: Statistical Decision-Tree Models For Parsing
Magerman, David M.
1995 Annual Meeting Of The Association For Computational Linguistics

J93-1002: Generalized Probabilistic LR Parsing Of Natural Language (Corpora) With Unification-Based Grammars
Briscoe, Ted; Carroll, John A.
1993 Computational Linguistics

J98-4004: PCFG Models Of Linguistic Tree Representations
Johnson, Mark
1998 Computational Linguistics

P96-1025: A New Statistical Parser Based On Bigram Lexical Dependencies
Collins, Michael John
1996 Annual Meeting Of The Association For Computational Linguistics

W97-0301: A Linear Observed Time Statistical Parser Based On Maximum Entropy Models
Ratnaparkhi, Adwait
1997 Conference On Empirical Methods In Natural Language Processing

P99-1069: Estimators For Stochastic "Unification-Based" Grammars

Johnson, Mark; Geman, Stuart; Canon, Stephen; Chi, Zhiyi; Riezler, Stefan
1999 Annual Meeting Of The Association For Computational Linguistics

P05-1012: Online Large-Margin Training Of Dependency Parsers
McDonald, Ryan; Crammer, Koby; Pereira, Fernando C.N.
2005 Annual Meeting Of The Association For Computational Linguistics

C04-1010: Deterministic Dependency Parsing Of English Text
Nivre, Joakim; Scholz, Mario
2004 International Conference On Computational Linguistics

P08-1108: Integrating Graph-Based and Transition-Based Dependency Parsers
Nivre, Joakim; McDonald, Ryan
2008 Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue

C96-1058: Three New Probabilistic Models For Dependency Parsing: An Exploration
Eisner, Jason M.
1996 International Conference On Computational Linguistics

P02-1043: Generative Models For Statistical Parsing With Combinatory Categorial Grammar
Hockenmaier, Julia; Steedman, Mark
2002 Annual Meeting Of The Association For Computational Linguistics

P01-1010: What Is The Minimal Set Of Fragments That Achieves Maximal Parse Accuracy?
Bod, Rens
2001 Annual Meeting Of The Association For Computational Linguistics

P04-1013: Discriminative Training Of A Neural Network Statistical Parser
Henderson, James B.
2004 Annual Meeting Of The Association For Computational Linguistics

P05-1011: Probabilistic Disambiguation Models For Wide-Coverage HPSG Parsing
Miyao, Yusuke; Tsujii, Jun'ichi
2005 Annual Meeting Of The Association For Computational Linguistics

C92-2065: Probabilistic Tree-Adjoining Grammar As A Framework For Statistical Natural Language Processing
Resnik, Philip
1992 International Conference On Computational Linguistics

C02-1013: High Precision Extraction Of Grammatical Relations
Carroll, John A.; Briscoe, Ted
2002 International Conference On Computational Linguistics

W07-2207: Efficiency in Unification-Based N-Best Parsing
Zhang, Yi; Oepen, Stephan; Carroll, John A.

2007 Tenth International Conference on Parsing Technologies

N07-1051: Improved Inference for Unlexicalized Parsing
Petrov, Slav; Klein, Dan
2007 Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference

P03-1054: Accurate Unlexicalized Parsing
Klein, Dan; Manning, Christopher D.
2003 Annual Meeting Of The Association For Computational Linguistics

Appendix B.

Task-based Evaluation Materials

Instructions to Novice Group A

Imagine the following situation:

Your friend is going to Sydney to present a scientific topic to an MSc class. This topic is generally in her field, but she does not know any of the details of the topic. Because she is in a rush, she has asked you to provide her with the 20 most important papers about the topic, which she will then read on the plane. She would appreciate them in the order she should read them in case she runs out of time during the flight. You have only 20 minutes of search time to find these papers. This will leave enough time for them to be printed before she leaves for the airport.

In this experiment you will do this type of search twice, each time with a different topic and different search system. Both systems allow you to:

- type in search queries to be presented with a list of relevant papers
- click on the paper title to read the paper
- see how many people cite each paper returned by the search and follow links to see the citing papers
- follow links to see papers similar to each paper returned by the search

The systems are slightly different in what they present for each paper:

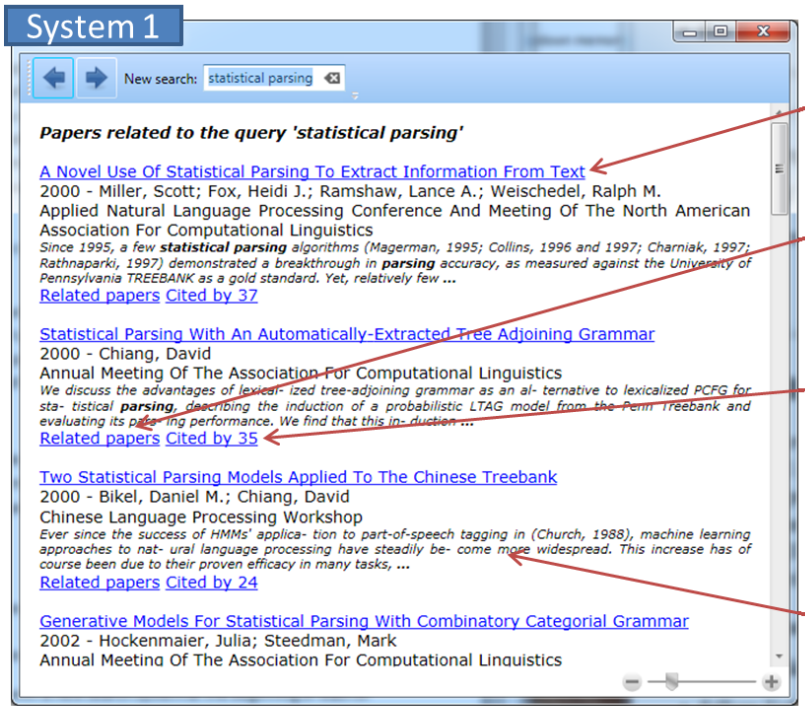
- System 1 provides a short summary-style "snippet" for each paper. This snippet presents part of the paper that is relevant to your query.
- System 2 provides relevant technical terms for each paper. They can be explored by clicking on them: you will be presented with papers relevant to each technical term.

This hour will be split up as follows:

- Introduction : 5 minutes
- System 1 : 5 minutes training & 20 minutes search
- Break : 5 minutes
- System 2 : 5 minutes training & 20 minutes search

You will be given a new search topic and set up with a new search system at the beginning of each of your two search sessions.

At the end of each session, you will be asked to hand in the ranked list of 20 papers that you have produced. To produce this list, you can copy-and-paste the paper details from the search window into a Word document. During the search you can reorder and delete what you have put in that Word document.

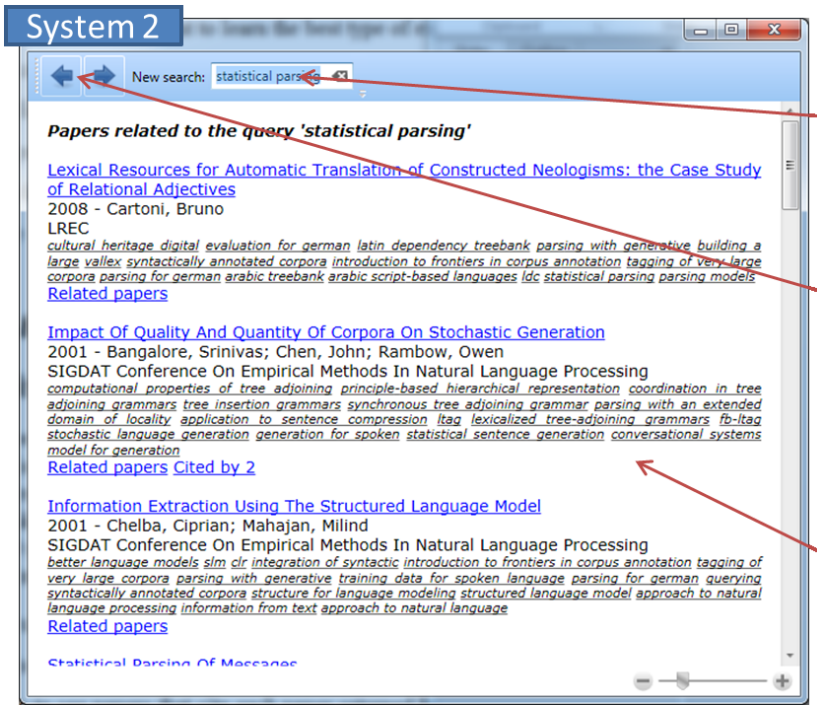


System 1 & 2: Click here to read the paper.

System 1 & 2: Click here for a list of related papers.

System 1 & 2: Click here for a list of papers that cite this paper.

System 1 only: this is a snippet from the paper.



System 1 & 2: Type your search query here and press <ENTER>

System 1 & 2: Navigate back- and forwards through your queries.

System 2 only: these are technical terms from the paper. Click them for a list of papers relevant to the term.

Instructions to Novice Group B

Imagine the following situation:

Your friend is going to Sydney to present a scientific topic to an MSc class. This topic is generally in her field, but she does not know any of the details of the topic. Because she is in a rush, she has asked you to provide her with the 20 most important papers about the topic, which she will then read on the plane. She would appreciate them in the order she should read them in case she runs out of time during the flight. You have only 20 minutes of search time to find these papers. This will leave enough time for them to be printed before she leaves for the airport.

In this experiment you will do this type of search twice, each time with a different topic and different search system. Both systems allow you to:

- type in search queries to be presented with a list of relevant papers
- click on the paper title to read the paper
- see how many people cite each paper returned by the search and follow links to see the citing papers
- follow links to see papers similar to each paper returned by the search

The systems are slightly different in what they present for each paper:

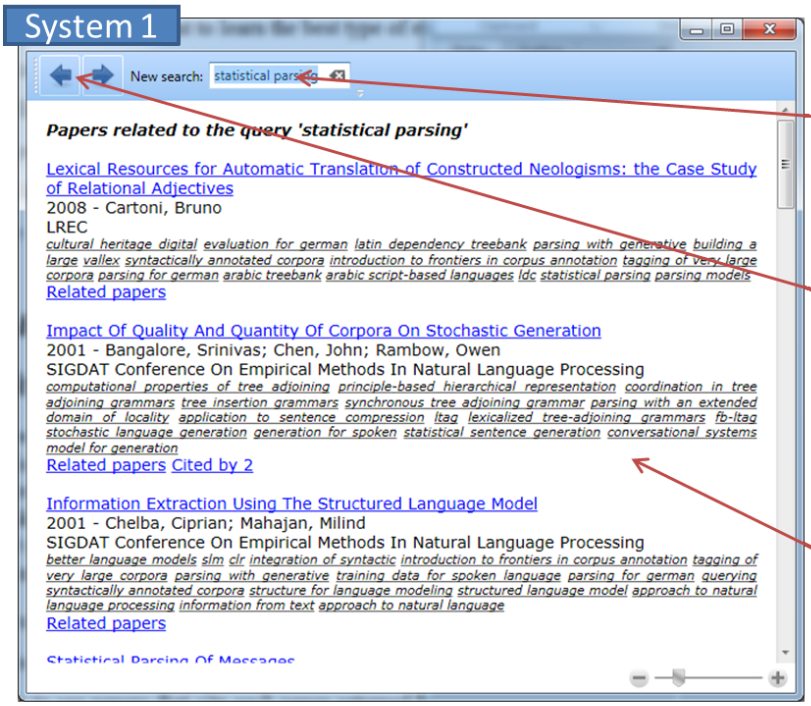
- System 1 provides relevant technical terms for each paper. They can be explored by clicking on them: you will be presented with papers relevant to each technical term.
- System 2 provides a short summary-style "snippet" for each paper. This snippet presents part of the paper that is relevant to your query.

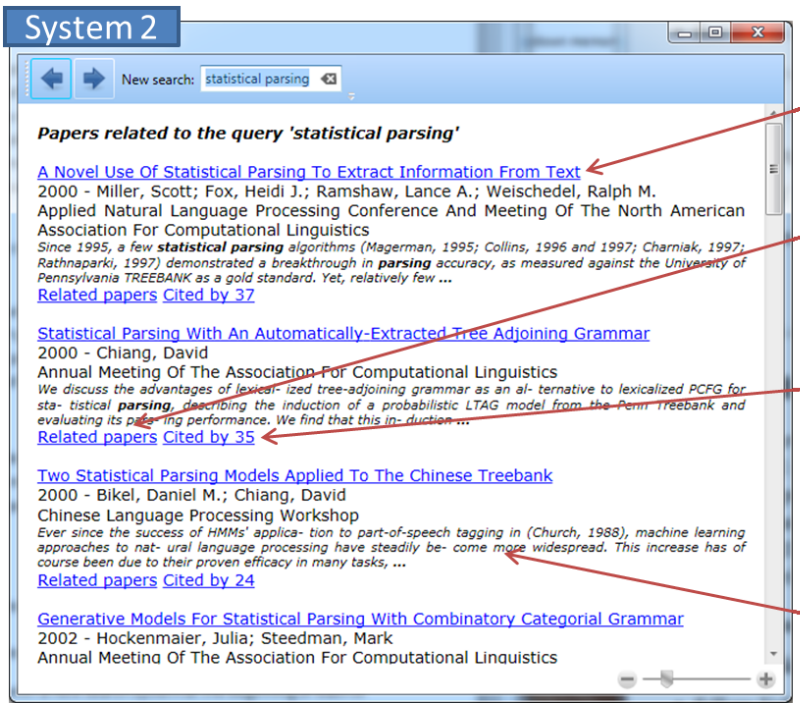
This hour will be split up as follows:

- Introduction : 5 minutes
- System 1 : 5 minutes training & 20 minutes search
- Break : 5 minutes
- System 2 : 5 minutes training & 20 minutes search

You will be given a new search topic and set up with a new search system at the beginning of each of your two search sessions.

At the end of each session, you will be asked to hand in the ranked list of 20 papers that you have produced. To produce this list, you can copy-and-paste the paper details from the search window into a Word document. During the search you can reorder and delete what you have put in that Word document.





System 1 & 2: Type your search query here and press <ENTER>

System 1 & 2: Navigate back- and forwards through your queries.

System 1 only: these are technical terms from the paper. Click them for a list of papers relevant to the term.

System 1 & 2: Click here to read the paper.

System 1 & 2: Click here for a list of related papers.

System 1 & 2: Click here for a list of papers that cite this paper.

System 2 only: this is a snippet from the paper.