Information Genealogy: Uncovering the Flow of Ideas in Non-Hyperlinked Document Databases

Benyah Shaparenko
Department of Computer Science
Cornell University
Ithaca, NY 14853
benyah@cs.cornell.edu

Thorsten Joachims
Department of Computer Science
Cornell University
Ithaca, NY 14853
tj@cs.cornell.edu

ABSTRACT

We now have incrementally-grown databases of text documents ranging back for over a decade in areas ranging from personal email, to news-articles and conference proceedings. While accessing individual documents is easy, methods for overviewing and understanding these collections as a whole are lacking in number and in scope. In this paper, we address one such global analysis task, namely the problem of automatically uncovering how ideas spread through the collection over time. We refer to this problem as *Information Genealogy*. In contrast to bibliometric methods that are limited to collections with explicit citation structure, we investigate content-based methods requiring only the text and timestamps of the documents. In particular, we propose a language-modeling approach and a likelihood ratio test to detect influence between documents in a statistically well-founded way. Furthermore, we show how this method can be used to infer citation graphs and to identify the most influential documents in the collection. Experiments on the NIPS conference proceedings and the Physics ArXiv show that our method is more effective than methods based on document similarity.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Measurement, Performance

Keywords

Information Genealogy, Flow of Ideas, Language Models, Citation Inference, Text Mining, Temporal Data

1. INTRODUCTION

In many domains, complete electronic records of documents now reach back for over a decade, including computer science research papers, US news articles, and most people's personal email. These databases incrementally grow through an "evolutionary" process, where new documents are influenced by the content of already existing documents. For example, scientific documents extend existing ideas, newsstories refine and comment on other articles, and emails aggregate or respond to other emails.

While keyword-based retrieval systems allow efficient access to individual documents in such corpora, we yet lack methods to understand the corpus as a whole. To remedy this shortcoming, this paper investigates whether it is possible to uncover the temporal dependency structure of a corpus. Which documents influenced each other? How did ideas spread through the corpus over time? Which documents (or authors) were most influential? While many of these question have been addressed for hyperlinked data with explicit citation structure, explicit citations are not available in most domains. We therefore aim to address these questions based on the (textual) content of the documents alone.

The premise for this research is that ideas manifest themselves in statistical properties of a document (e.g. the distribution of words), and that these properties can act as a signature for an idea which can be traced through the database. Following this premise, we present a probabilistic model of influence between documents and design a content-based significance test to detect whether one document was influenced by an idea first presented in another document. The test takes the form of a Likelihood Ratio Test (LRT) and leads to a convex programming problem that can be solved efficiently. Our goal is to use this test for inferring an influence graph derived from the text of the documents alone. Analogous to detecting inheritance from genes, we refer to this text-mining problem as *Information Genealogy*.

Using corpora of scientific literature, we show that it is indeed possible to infer meaningful influence graphs from the text of the documents. Evaluating against the explicit citation graphs for these corpora, we find that the automatically-computed influence graphs are similar to the citation graphs. The ablility to automatically generate an influence graph for a collection enables a range of applications, from browsing, to visualizing and mining the structure of the network. As a simple example, we demonstrate that the in-degree of the influence graph provides an interesting measure of document impact, similar to the in-degree of the citation graph.

2. MEASURING INFLUENCE

In this paper, we investigate and operationalize the notion of influence between documents. Influence is an interesting relationship between documents in historically grown

KDD'07, August 12–15, 2007, San Jose, California, USA.

databases, since such corpora have grown through a self-referential process: documents are influenced by the content of prior documents, but also contribute new ideas which in turn influence later documents. Our goal is to uncover and mine how ideas introduced in some document spread through the corpus over time.

At first glance, one might think that similarity, as captured by information retrieval metrics like TFIDF cosine similarity (see e.g. [32]), provides the full picture of influence. However, this is not the case.

On the one hand, similarity can occur without influence. First, if a document $d^{(1)}$ introduces an idea that is picked up in documents $d^{(2)}$ and $d^{(3)}$, then $d^{(2)}$ and $d^{(3)}$ will likely be similar but do not necessarily influence each other. Second, two documents might concurrently propose the same idea. Again, neither document influences the other although the documents likely are similar.

On the other hand, influence can occur with very little similarity. In the scientific literature, for example, a large textbook might devote a section to an idea introduced in an earlier research paper. Clearly, the paper had influence on the textbook. However, the overall similarity between the book and the paper is small, since the book covers many other ideas as well.

As we will briefly review in the following, most prior work on analyzing temporal corpora has focused on identifying relatedness between documents, not influence. We will then develop a probabilistic model and a statistical test for detecting influence, and show that it captures influence better than similarity and provides a more complete understanding and model of influence.

2.1 Topic Detection and Tracking

Topic Detection and Tracking (TDT) [5, 6] has the goal of grouping documents by topic. Unlike influence, which is a directed relationship, TDT aims to group documents into equivalence classes. While TDT approaches have relied heavily on finding similarity measures that capture closeness in topic, this approach is not necessarily detecting influence, as we have argued above. Methods that model influence not only can detect and track topics and ideas, but also can provide reference points for *why* a document collection developed as it did. Another minor difference is that the TDT studies were performed in an online setting, while we assume access to the full corpus at any time.

Similar work on detecting and visualizing topic development includes visualization methods such as Temporal Cluster Histograms [34] and ThemeRiver [15], EM-based corpus evolution detection [29], temporal clustering methods [7, 37], continuous time clustering models [37], Thread Decomposition [14], Independent Component Analysis [22], topic-intensity tracking [23], and Topical Precedence [27].

2.2 Real-World Influence on Documents

Research on Burst Detection [21] and TimeMines [36] aims to identify hidden causes based on changes in the word distribution over time. However, their notion of influence is different from ours. These approaches determine influence from real-world events on topics (e.g., events influencing US State of the Union Addresses). Instead, we model the influence of documents on each other.

2.3 Citation and Hyperlink Analysis

In bibliometrics, a document's influence is measured through properties of the citation graph [30, 31, 20, 12]. Our work differs from citation analysis because our method is based on document content, not on citations. We assume that influence is inherently reflected in the statistical properties of documents. In particular, we conjecture that when one document influences another, the influenced document shows traces of the word distribution of the original document[1]. Besides bibliometrics' consideration of citation analysis on research papers, other methods work on general hyperlink structure. One of the most well-known such methods is PageRank [31], which uses hyperlink structure to find influential Web pages.

2.4 Automatic Hypertext

There is related work on automatically adding hyperlinks in information retrieval and related fields. Most prominently, Link Detection was a key task in the TDT evaluations [5]. Several proposals and methods exist for introducing hyperlinks between similar documents or passages of documents [11, 10, 33, 26, 2, 4, 3, 24, 25]. Good surveys are given in [38] and the 1997 special issue of Information Processing and Management [1]. The work we propose is different in several respects. First, our goal is to detect influence between documents, not just their "relatedness." This will allow a causal interpretation of the resulting citation graph. Second, we take a statistical testing approach to the problem of identifying influence links, which can be seen as synonymous to citations. This will give a formal semantic to the predictions of the methods, give theoretical guidance on how to apply the methods, and expose underlying assumptions.

2.5 Language and Topic Models

We take a probabilistic language modeling approach in the development of our methods. While we rely on a rather basic language model for the sake of simplicity, more detailed language models exist and can possibly be employed as well. Previous work by Steyvers et al. [35] looks at how document text can be generated by a two-step model of generating topics probabilistically from authors, and then words probabilistically from topics. There has also been language modeling work done in the natural language processing and machine learning [28, 16, 8], speech recognition [19], and information retrieval communities [39, 24, 25].

3. METHODS

In constructing an influence graph for a database of documents, the core problem is to determine when and where ideas flow from one document to another document. In the following, we propose a probabilistic model of influence in a language-modeling framework, and develop a Likelihood Ratio Test (LRT) [9] for detecting whether one document has significantly influenced another document.

3.1 Probabilistic Model and Motivation

To make the method widely applicable, we have only two basic requirements for our corpus of documents — first, the documents contain text and, second, the documents have

[1]Note that our goal is not plagiarism detection, where authors would try to disguise their choice of words.

timestamps. Formally, the corpus $\mathcal{D}$ is a collection of n documents $\{D^{(1)} \cdots D^{(n)}\}$, where each document $D^{(i)} \in \mathcal{D}$ has an associated timestamp $time(D^{(i)})$. There are m different terms (i.e. words) across the entire corpus, which are denoted by $\{t_1 \cdots t_m\}$.

We assume that the document is a vector-valued random variable $D = (W_1 \cdots W_{|D|})$, which describes a document as a sequence of random variables W_i, one for each word in the document. A particular observed document is denoted as $d = (w_1 \cdots w_{|d|})$. In the following, we assume that each document $D^{(i)} \in \mathcal{D}$ was generated by a unigram language model $P(D^{(i)} = d^{(i)} \mid \theta^{(i)})$ with parameters $\theta^{(i)}$ specific to that document.

Model 1. (Document Language Model)
A document $D^{(i)} \in \mathcal{D}$ is assumed to be generated by independently drawing $|D^{(i)}|$ words from a document specific distribution with individual word probabilities $\theta^{(i)} = (\theta^{(i)}_{t_1}, ..., \theta^{(i)}_{t_m})$, i.e.

$$\begin{aligned} P(D^{(i)} = d^{(i)} \mid \theta^{(i)}) &= P(D^{(i)} = (w_1^{(i)} \cdots w^{(i)}_{|D^{(i)}|}) \mid \theta^{(i)}) \\ &= \prod_{j=1}^{|D^{(i)}|} P(W^{(i)} = w_j^{(i)} \mid \theta^{(i)}) \\ &= \prod_{j=1}^{|D^{(i)}|} \theta^{(i)}_{w_j} \end{aligned}$$

Note that we do not explicitly model document length. We chose this basic language model for mathematical and computational convenience. However, our approach can be extended to more complex language models as well (e.g. n-gram models).

Since we wish to detect the flow of ideas and influence between documents, we also need a model of inter-document relationship. We formalize this as a question of how the language model $\theta^{(new)}$ of a new document $D^{(new)}$ depends on the language models $\{\theta^{(1)} \cdots\}$ of the documents that precede $\theta^{(new)}$ in time. In particular, we assume that the language model of a new document can be (approximately) expressed as a mixture distribution over the language models of previous documents.

Model 2. (Inter-Document Influence Model)
A new document $D^{(new)}$ is generated by a mixture distribution of the already existing documents $D^{(i)}$ with $i \in \mathcal{P}$ for $\mathcal{P} = \{i : time(D^{(i)}) < t_0\}$, in particular

$$P(D^{(new)}{=}d^{(new)} \mid \pi) = \prod_{j=1}^{|D^{(new)}|} \sum_{p \in \mathcal{P}} \pi_p P(W^{(p)}{=}w_j \mid \theta^{(p)}) \quad (1)$$

with mixing weights π satisfying $0 \le \pi_i$ and $\sum_i \pi_i = 1$.

In this dependency model, a new document is composed of parts generated by the word distributions of old documents, where the mixing coefficient π_p indicates the fraction of $D^{(new)}$ that is generated from $D^{(p)}$. Clearly, there is direct influence of a document $D^{(p)}$ on $D^{(new)}$, if the respective mixing coefficient is non-zero. Note that the resulting language model for $D^{(new)}$ is again a unigram model, so that $P(D^{(new)}{=}d^{(new)} \mid \pi) = P(D^{(new)}{=}d^{(new)} \mid \theta^{(new)})$ with

$$\theta^{(new)} = \sum_{p \in \mathcal{P}} \pi_p \theta^{(p)}. \quad (2)$$

In actual document collections, documents typically contain some original part that does not come from previous documents. To account for the original portion of a document in our model, we include a distribution $\theta^{(o)}$ with weight π_o in the mixture. It models the distribution of words that is original to the document and that cannot be explained by previous documents. (In practice, we will assume that π_o is fixed, but that we have no knowledge of $\theta^{(o)}$.

Model 3. (Inter-Document Influence Model with Original Content)
A new document $D^{(new)}$ is generated by a mixture distribution of the already existing documents $D^{(i)}$ with $i \in \mathcal{P}$ for $\mathcal{P} = \{i : time(D^{(i)}) < t_0\}$, and a document specific mixture component $\theta^{(o)}$ with weight π_o, in particular

$$P(D^{(new)}{=}d^{(new)} \mid \pi) = \prod_{j=1}^{|D^{(new)}|} \sum_{p \in \mathcal{P} \cup \{o\}} \pi_p P(W^{(p)}{=}w_j \mid \theta^{(p)}) \quad (3)$$

with mixing weights π s.t. $0 \le \pi_i, \pi_o$ and $\pi_o + \sum_i \pi_i = 1$.

In the case when the documents have no original content, setting $\pi_o = 0$ in the Inter-Document Influence Model with Original Content results in Model 2. Vice versa, Model 2 also subsumes Model 3 by simply introducing an artificial single-word document for each term in the corpus and constraining their mixture weights to sum to π_o. We will therefore focus our further derivations on Model 2 for the sake of simplicity.

We will now show how this probabilistic setup can be used in a significance test for detecting whether a particular mixing weight π_p is non-zero in a given document collection.

3.2 A Statistical Test for Detecting Influence

How can one decide whether a candidate influential document $d^{(can)}$ had a significant influence on $d^{(new)}$ given the other documents in the collection? First, $d^{(can)}$ can only have had an influence on $d^{(new)}$ if it had been published before $d^{(new)}$ (i.e. $time(d^{(can)}) < time(d^{(new)})$). Note that this is already encoded in the Inter-Document Influence Models defined above. Second, influence should be attributed to the first publication that introduced an idea through an original section or portion, not to other documents that later copied an idea. To illustrate this in the context of research papers, this means that influence should be credited to the original article, not a tutorial that reproduced the original idea.

Under these conditions, the decision of whether document $d^{(new)}$ shows significant influence from $d^{(can)}$ can be phrased as a Likelihood Ratio Test [9]. In general, a Likelihood Ratio Test decides between two families of densities described by sets of parameters Π and Π_0 that are nested, i.e. $\Pi_0 \subset \Pi$. Applied to our case, Π will be all mixture models of $D^{(new)}$ as in Eq. (1) with parameters π_i for all documents $\mathcal{P}$ published prior to $t_0 = time(d^{(can)})$ (and therefore prior to $d^{(new)}$), as well as a parameter π_{can} for $d^{(can)}$.

$$\Pi = \left\{ \pi : \sum_{i \in \mathcal{P} \cup \{can\}} \pi_i = 1 \ \wedge \ \pi_i \ge 0 \right\}$$

The subset Π_0 of the mixture models in Π will be the models where $d^{(can)}$ has zero mixture weight (i.e. $\pi_{can} = 0$).

$$\Pi_0 = \left\{ \pi : \sum_{i \in \mathcal{P} \cup \{can\}} \pi_i = 1 \ \wedge \ \pi_i \geq 0 \ \wedge \ \pi_{can} = 0 \right\}$$

Note that the set of prior documents $\mathcal{P} = \{i : time(d^{(i)}) < time(d^{(can)})\}$ serves as a "background model" of what was already known when $d^{(can)}$ was published. Against this background, we can then measure how much the new ideas in document $d^{(can)}$ influenced $d^{(new)}$.

The null hypothesis of the Likelihood Ratio test is that the data comes from a model in Π_0 (i.e. document $d^{(new)}$ was not influenced by $d^{(can)}$ given the documents published before $d^{(can)}$). To reject this null hypothesis, a likelihood ratio test considers the following test statistic

$$\Lambda_{d^{(can)}}(d^{(new)}) = \frac{\sup_{\pi \in \Pi_0}\{P(D^{(new)} = d^{(new)}|\pi)\}}{\sup_{\pi' \in \Pi}\{P(D^{(new)} = d^{(new)}|\pi')\}}$$

Note that $P(D^{(new)} = d^{(new)}|\pi)$ is convex over Π and Π_0, so that the suprema can be computed efficiently. We will elaborate on the computational aspects below. Intuitively, the value of $\Lambda_{d^{(can)}}(d^{(new)})$ measures whether using $d^{(can)}$ in the mixture model better explains the content of $d^{(new)}$ than just using previously published documents. More formally, $\Lambda_{d^{(can)}}(d^{(new)})$ compares the likelihood $\sup_{\pi' \in \Pi}\{P(D^{(new)} = d^{(new)}|\pi')\}$ of the best mixture model containing $d^{(can)}$ with the likelihood $\sup_{\pi \in \Pi_0}\{P(D^{(new)} = d^{(new)}|\pi)\}$ of the best mixture model that does not use $d^{(can)}$ (i.e. $\pi_{can} = 0$). The test then decides whether there is significant evidence that a non-empty part of $d^{(new)}$ was generated from $d^{(can)}$, in comparison to using a mixture only over the other language models.

If the null hypothesis is true, then the distribution of the LRT statistic $-2\log(\Lambda_{d^{(can)}}(d^{(new)}))$ is asymptotically (in the document length under the unigram model) χ^2 with one degree of freedom.

$$-2\log(\Lambda_{d^{(can)}}(d^{(new)})) \sim \chi^2_1$$

The null hypothesis H_0 should be rejected, if

$$-2\log(\Lambda_{d^{(can)}}(d^{(new)})) > c$$

for some c selected dependent on the desired significance level. For a significance level of 95%, c should be 3.84. This captures the intuition that we can reject the null hypothesis and conclude that $d^{(can)}$ had a significant influence on $d^{(new)}$, if the best model that does not use $d^{(can)}$ has a much worse likelihood than the best model that considers $d^{(can)}$. Specifically, if $-2\log(\Lambda_{d^{(can)}}(d^{(new)}))$ is large, then $d^{(can)}$ significantly influenced $d^{(new)}$ given all other documents published at that time.

To estimate the language models $\theta^{(i)}$ of the documents entering into the mixture model of $d^{(new)}$, we use the maximum-likelihood estimate. We denote with $tf^{(i)}$ the term frequency (TF) vector of document $d^{(i)}$, where each entry $tf^{(i)}_j$ is the number of times that term t_j appears in the document $d^{(i)}$. The estimator is

$$\theta^{(i)}_{w_j} = \frac{tf^{(i)}_{w_j}}{|d^{(i)}|},$$

which is simply the fraction of times the particular word occurs in the observed document $d^{(i)}$. Using a more advanced estimator instead is straightforward, but we will not discuss this for the sake of simplicity.

3.3 Relating the LRT to Detecting Influence

What does it mean for the LRT to significantly reject the null hypothesis? A good intuition is to think of this method in the context of trying to explain the ideas and content found in $d^{(new)}$. There are two choices. First, explain $d^{(new)}$ using only other documents preceding $d^{(can)}$ as well as some original component. Second, explain $d^{(new)}$ with these plus an additional $d^{(can)}$. If the first case already provides a wonderful model for $d^{(new)}$, then adding $d^{(can)}$ will not explain $d^{(new)}$ any more accurately. Thus, $d^{(can)}$ really does not contribute to $d^{(new)}$. On the other hand, if $d^{(can)}$ introduced some new ideas and terminology that then flowed to $d^{(new)}$, using $d^{(can)}$ will provide a better explanation than only using $\mathcal{P}$. Consequently, the likelihood of $d^{(new)}$ using $d^{(can)}$ will be significantly higher than without it, and we can reject the null hypothesis. To summarize, rejecting the null hypothesis means that $d^{(can)}$ significantly exerted influence on $d^{(new)}$.

3.4 Computing the LRT

Computing the value of $\Lambda_{d^{(can)}}(d^{(new)})$ requires solving two optimization problems.

$$L_0 = \sup_{\pi \in \Pi_0}\{P(D^{(new)} = d^{(new)}|\pi)\} \text{ and} \quad (4)$$

$$L = \sup_{\pi \in \Pi}\{P(D^{(new)} = d^{(new)}|\pi)\}. \quad (5)$$

Given our model, these problems can be solved efficiently. Note that we can write the log-likelihood $L(\pi \mid d^{(new)}, \mathcal{S})$ of the document $d^{(new)}$ w.r.t. a fixed π as

$$\begin{aligned} \log L(\pi \mid d^{(new)}) &= \log P(d^{(new)} \mid \pi) \\ &= \sum_{j=1}^{m} tf^{(new)}_j \log(\sum_{i \in \mathcal{S}} \pi_i \theta^{(i)}_j). \end{aligned}$$

With $\mathcal{S}$ we denote the set of documents considered in the model. This gives $\mathcal{S} = \mathcal{P} \cup \{can\}$ for Π and $\mathcal{S} = \mathcal{P}$ for Π_0. In this notation, each of the optimization problems in Eq. (4) and (5) takes the form

$$\begin{aligned} \max_{\pi \in \Re^{|\mathcal{S}|}} \quad & \log L(\pi \mid d^{(new)}) \\ \text{subject to} \quad & \sum_{i \in \mathcal{S}} \pi_i = 1 \\ & \forall i \in \mathcal{S} : \pi_i \geq 0. \end{aligned}$$

For Model 3 an additional linear constraint is introduced to limit the amount of original content π_o to not be more than a user-specified parameter σ. This constraint is necessary, since otherwise the $\theta^{(o)}$ mixture component could always perfectly explain $d^{(new)}$.

It is easy to see that these optimization problems are convex, which means that they have no local optima and that there are efficient methods for computing the solution. We currently use the separable convex implementation for the general-purpose solver Mosek [18] to solve the optimization problems. However, more specialized code is likely to be substantially more efficient.

While solving each optimization problem is efficient, analyzing a collection requires a quadratic number of LRTs, each with on the order of n documents in the background

model. In particular, for each document $d^{(new)}$, we need to test all prior documents

$$\mathcal{C} = \left\{ d^{(i)} : time(d^{(i)}) < time(d^{(new)}) \right\} \quad (6)$$

in the collection, since all of these are candidates for having influenced $d^{(new)}$. For each document $d^{(can)}$ in the candidate candidate set $\mathcal{C}$ of $d^{(new)}$, we then have a background model

$$\mathcal{P}_{d^{(can)}} = \left\{ d^{(i)} : time(d^{(i)}) < time(d^{(can)}) \right\}. \quad (7)$$

Computing all tests exhaustively for a large corpus can be expensive. We therefore use the following approximations.

Both approximations are based on the insight that some similarity is necessary for influence. The potentially influential document $d^{(can)}$ must have some similarity with $d^{(new)}$. Therefore, we first approximate the candidate set to contain the $k_{\mathcal{C}}$ nearest neighbors of $d^{(new)}$ from $\mathcal{C}$. We use cosine distance between TF and TFIDF vectors for document similarity. Second, an analogous argument applies to the background models $\mathcal{P}_{d^{(can)}}$. We therefore approximate the background model, using only the $k_{\mathcal{P}}$ most similar documents from $\mathcal{P}$. Since selecting $\mathcal{P}$ combines document vectors by addition, we use cosine distance between document TF vectors to select $\mathcal{P}$. In the experiments we set $k_{\mathcal{C}} = k_{\mathcal{P}}$ and refer to this parameter as k. We will empirically evaluate the effect of these approximations depending on k.

4. EXPERIMENTS

We wish to measure how well these models' assumptions match real data. First, how does an influence graph inferred by the LRT method compare against a citation graph? Second, can the influence graph identify top influential papers?

4.1 Experiment Setup and Corpora

The concept of influence and idea flow between documents corresponds well with the notion of a citation. Consequently, we focus on research papers to provide a quantitative evaluation of the LRT method by comparing with citations.

The first corpus is the full-text proceedings of the Neural Information Processing Systems (NIPS) conference [17] from 1987-2000, with a timestamp of the publication year. NIPS has 1955 documents, with 74731 terms (features). We manually constructed the graph of 1512 intra-corpus citations, but only compare to citations of previous documents in time. We ignore citations of first-year documents since the LRT requires a background model.

The second corpus is the theoretical high-energy physics (HEPTH) section of the Physics ArXiv [13] from Aug. 1991 to Apr. 2006. We aggregate the full-text papers by year. HEPTH has 39008 documents, 229194 terms, and 557582 citations. SLAC-SPIRES compiled these citations.

4.2 Inferring Influence Graphs

This set of experiments analyzes how well the LRT recovers the influence graph. After an illustrative example, we explore the LRT's sensitivity on synthetic data under controlled experiment conditions, and then evaluate on two real-world datasets.

4.2.1 Qualitative Evaluation

We first discuss a simple example to illustrate the LRT method's behavior and how it compares to citations. Figure 2 shows those documents that NIPS document 1541

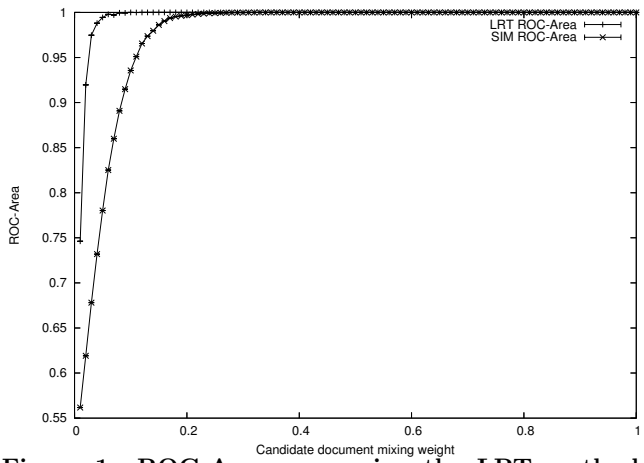


Figure 1: ROC-Area comparing the LRT method against a cosine similarity baseline. The x-axis is π_{can}. At a π_{can} level, the ROC-Area measures the quality of influence prediction in documents with the specified π_{can} as compared against documents with $\pi_{can} = 0$.

(Schoelkopf et al. on "Shrinking the Tube: a New Support Vector Regression Algorithm") most significantly influenced according to the LRT statistic. Three of the top five papers actually cite document 1541 (or a document with equivalent content from another venue). Furthermore, the top document could arguably have cited 1541 as well, since it relies on the ν-parameterization of SVMs that document 1541 introduced to NIPS. In fact, all papers (except "Fast Training of Support Vector Classifiers") consider this new parameterization. Note that the paper "ν-arc: Ensemble Learning in the Presence of Outliers" is not about SVMs, but uses the ν-parametrization in the context of boosting.

The LRT appears to accurately focus on the paper's original contribution, the ν-parameterization. General SVM papers do not score highly, since they are already modeled by earlier papers, e.g. paper 1217 "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing" of V. Vapnik et al., which was one of the first SVM papers in NIPS. When considering influencers of "A Support Vector Method for Clustering" by A. Ben-Hur et al. (using the conventional parameterization), the method correctly recognizes that paper 1541's influence is low ($-2\log(\Lambda_{d^{(1541)}}(d^{(new)})) = 67.0$) even though the documents are similar. Paper 1217 already "explains" the SVM content ($-2\log(\Lambda_{d^{(1217)}}(d^{(new)})) = 535.0$).

4.2.2 Quantitative Evaluation on Synthetic Data

Beyond this qualitative example, how accurately can the LRT discover influence? How much must $d^{(new)}$ copy from $d^{(can)}$ before the LRT can detect it?

To explore these questions, we constructed artificial documents $d^{(new)}$ from the NIPS corpus as follows. A candidate document $d^{(can)}$ and a set $\mathcal{P}$ of $k = 100$ previous documents are chosen at random form the NIPS corpus so that the documents in $\mathcal{P}$ preceed $d^{(can)}$ in time. Then, 101 artificial new documents are generated according to Eq. 1, where each new document has been influenced by $d^{(can)}$ at the fractional levels of $\pi_{can} \in \{0.00, 0.01, 0.02, \cdots, 1.00\}$. The remaining mixing weights π_i are selected by generating random num-

$-2\log(\Lambda_{d^{(1541)}}(d^{(new)}))$	Cite?	Title and Author(s) of d'
321.2455	no	"Support Vector Method for Novelty Detection", B. Schoelkopf, Robert C. Williamson, Alex Smola, John Shawe-Taylor, John C. Platt.
221.8297	yes	"An Improved Decomposition Algorithm for Regression Support Vector Machines", Pavel Laskov.
219.8769	yes	"ν-arc: Ensemble Learning in the Presence of Outliers, Gunnar Raetsch", B. Scholkopf, Alex Smola, Kenneth D. Miller, Takashi Onoda, Steve Mims.
184.5493	no	"Fast Training of Support Vector Classifiers", Fernando Perez-Cruz, Pedro Alarcon-Diana, Angel Navia-Vazquez, Antonio Artes-Rodriguez.
168.8972	yes	"Uniqueness of the SVM Solution", Christopher J. C. Burges, David J. Crisp.

Figure 2: Papers that are influenced by NIPS paper 1541, "Shrinking the Tube: a New Support Vector Regression Algorithm" written by B. Schoelkopf, P. Bartlett, A. Smola, and R. Williamson. The leftmost column shows the LRT statistic value. (Larger LRT statistic values represent greater influence.)

bers uniformly on the interval $[0, 1]$, and then normalizing them so that they sum to $1 - \pi_{can}$. The LRTs are run on each new document. Additionally, TF document vector cosine similarity is measured between $d^{(can)}$ and each $d^{(new)}$. The entire process is repeated for 1000 random selections of $\mathcal{P}$ and $d^{(can)}$.

We computed ROC-Area in the following manner. First, we select a particular $\pi_{can} \in \{0.01 \cdots 1.00\}$. The generated documents at the π_{can} level are marked as positive examples. The negative examples are documents with $\pi_{can} = 0$. Finally, a ranking, either LRT statistic scores or cosine distance similarity, is used to compute ROC-Area.

Figure 1 shows that even if only a small portion (i.e. a few percent) of $d^{(new)}$ is drawn from $d^{(can)}$, the LRT accurately detects the influence. The similarity baseline needs a much larger signal. This example illustrates that similarity and influence are in fact different, and that the well-founded statistical approach can be more accurate and sensitive than an ad-hoc heuristic.

4.2.3 Quantitative Evaluation on Real Data

Moving to real data, we use the LRTs to discover the influence graph for NIPS and HEPTH. For each document $d^{(new)}$, we first compute a set of candidate documents $\mathcal{C}$ based on similarity. The elements of $\mathcal{C}$ are then ranked according to the LRT statistic (i.e. whether $d^{(can)}$ was significant in explaining $d^{(new)}$). The higher $d^{(can)}$ is ranked, the more likely that it influenced $d^{(new)}$, and we can derive the influence graph by thresholding (discussed below).

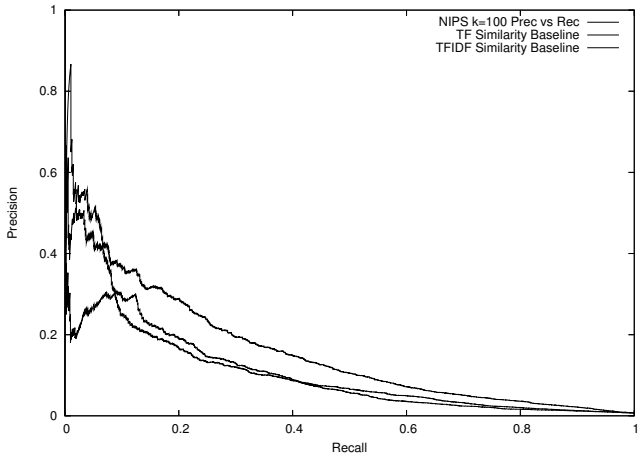
We evaluate the influence graph by a graph-based mean-average-precision (G-MAP) metric. For a document d, average the precision of the ranked predicted list of influencers at the positions corresponding to documents that d actually cites. Citations not in the list are averaged as 0, i.e. ranked at infinity. (As an information retrieval analogy, the influence list is the search result page, with citations being relevant results.) G-MAP is the mean of the per-document average precision scores. We exclude documents from the first two years due to edge effects (the LRT cannot predict citations for the first years since $\mathcal{C}$ or $\mathcal{P}$ are empty).

We compare G-MAP for the LRT method against G-MAP of a similarity-based heuristic, which serves as a baseline. This baseline method ranks the elements of $\mathcal{C}$ not by LRT score, but by similarity. We explored several similarity measures. The best similarity measures in our experiments are TF cosine and TFIDF cosine. We report their performance.

Note that citations are not necessarily a perfect gold standard for influence, since they reflect idiosycracies of how scientific communities cite prior work. For example, in Figure 2 authors sometimes cited a journal paper or book instead of the NIPS paper. Therefore, a G-MAP of 1 is not achievable.

	TF		TFIDF	
G-MAP	LRT	SIM	LRT	SIM
NIPS	0.4489	0.3948	0.4531	0.4412
HEPTH	0.2432	0.2216	0.2543	0.2167

Table 1: G-MAP scores comparing the LRT against the similarity baseline. The similarity measure to select $\mathcal{P}$ is the TF cosine and to select/rank $\mathcal{C}$ is either the TF cosine or the TFIDF cosine. Results are reported for $k = 100$ and $\sigma = 0.05$.



TFIDF $\mathcal{C}$ for LRT with $k = 100$ and $S = .05$

Figure 3: Precision vs. Recall on NIPS. The three lines are (from top to bottom) the LRT method's precision at a recall level with TFIDF cosine used to select $\mathcal{C}$, the TFIDF distance $\mathcal{C}$ similarity baseline, and the TF distance $\mathcal{C}$ similarity baseline.

LRTs are more accurate than similarities.

Table 1 shows that the LRT achieves higher G-MAP scores than the similarity baselines on both NIPS and HEPTH. Among the two heuristic baselines, TFIDF cosine performs better then TF cosine. TFIDF cosine also appears to select better sets $\mathcal{C}$ for the LRT. The HEPTH results are reported for a random sample of 1600 documents.

G-MAP	TF		TFIDF	
	LRT	SIM	LRT	SIM
$\sigma = .001$	0.4575		0.4597	
$\sigma = .01$	0.4620		0.4649	
$\sigma = .05$	0.4489	0.3948	0.4531	0.4412
$\sigma = .1$	0.4475		0.4535	
$\sigma = .2$	0.4373		0.4447	

Table 2: G-MAP scores comparing the LRT for a range of $d^{(can)}$ influence mixing weights σ against the similarity baseline. The similarity measure to select $\mathcal{C}$ is either TF or TFIDF cosine. Results are reported on NIPS for $k = 100$.

G-MAP	TF		TFIDF	
	LRT	SIM	LRT	SIM
NIPS ($k = 100$)	0.4489	0.3948	0.4531	0.4412
NIPS ($k = 10$)	0.4067	0.3754	0.4580	0.4226
HEPTH ($k = 100$)	0.2432	0.2216	0.2543	0.2167
HEPTH ($k = 20$)	0.2227	0.2037	0.2264	0.1943

Table 3: G-MAP scores comparing the LRT against the similarity baseline for two k-NN approximation levels. The similarity measure for selecting $\mathcal{C}$ is either TF or TFIDF cosine. Results are reported on NIPS and HEPTH for $\sigma = .05$.

LRT scores are more comparable than similarities.

Table 1 showed that the LRT can find the most influential papers for one particular document. Figure 3 measures how well it can find the strongest edges in the whole influence graph. This precision-recall graph uses the ranking of all LRT statistic scores of all documents, with actual citations marked as positive examples. Figure 3 also shows the scores for using lists of TF and TFIDF cosine similarities. The LRT graph dominates the similarity baselines over the whole range and the difference in performance is larger than in the per-document evaluation. We conclude from this that LRT scores are more comparable between documents than similarity scores. This is to be expected because the LRT values have a clear probabilistic semantic. However, the similarity scores have no such guarantees.

Effects of the σ parameter.

Table 2 shows that the LRT is robust over a large range σ values. The LRT's G-MAP dominates the similarity baselines. However, $\sigma = 0.01$ seems to perform better than our initial guess of 0.05 used above.

Effect of k parameter in LRT approximations.

Table 3 shows G-MAP scores at differing levels of the k-NN approximation. Recall from Table 1 that G-MAP scores for HEPTH are substantially lower than for NIPS. We conjecture that this is due to the size of the corpus in relation to k. With a large corpus, $k = 100$ is likely to exclude too many relevant documents from consideration. We further analyze the role of k, in its two roles in controlling the sizes of $\mathcal{C}$ and $\mathcal{P}$.

First, k controls the size of $\mathcal{C}$. If k is too small, truly influential documents will not be tested by the LRT. E.g., in HEPTH, each document has 14 citations on average. With $k = 10$, it would be simply impossible to recover the entire citation graph. Therefore we conclude that k must be large enough to include all documents that make contributions to $d^{(new)}$. On HEPTH, $k = 100$ is better than $k = 20$ for TF and TFIDF cosine, and for LRT and similarity baseline. We believe this is because $k = 20$ is too restrictive. NIPS with TF cosine shows the same behavior.

Dataset ($\mathcal{C}$)	GMAP	GMAP (perfect $\mathcal{C}$)
NIPS (TFIDF)	0.4531	0.4556
NIPS (TF)	0.4489	0.4590
HEPTH (TFIDF)	0.2543	0.3803
HEPTH (TF)	0.2432	0.3906

Table 4: How close is the approximation to the optimal? G-MAP scores are reported for $S = .05$.

Optimal $\mathcal{C}$.

To better understand how much loss in performance is due to the k-NN approximation of $\mathcal{C}$, the following experiment explores the G-MAP scores of the LRT for a "perfect" $\mathcal{C}$. In particular, we construct $\mathcal{C}$ so that it includes all documents that $d^{(new)}$ actually cites, and then fill the remaining places in $\mathcal{C}$ with the most similar documents. Table 4 shows that for $k = 100$ the loss in performance due to an approximate $\mathcal{C}$ is fairly small on NIPS. For HEPTH, on the other hand, $k = 100$ shows a much greater loss, with G-MAP scores only about 60-65% of the optimal. We believe this loss occurs because $\mathcal{C}$ is too small to accomodate all the influential documents.

4.3 Identifying Influential Documents

What are the influential documents that have the most effect on the document collection's development? Which documents should one read to best grasp this development? We have already shown that LRTs can be used to infer an influence graph that is similar to a citation graph. We now investigate whether this influence graph can be used to identify the documents with the overall largest influence on the collection. In analogy to citation counts (i.e. the in-degree in the citation graph), we propose the in-degree in the influence graph as a measure of impact. If not noted otherwise, we form the influence graph by connecting each document $d^{(new)}$ with the l other nodes that receive the highest LRT value. We typically use $l = 10$, although we also explore this parameter's effect on performance.

4.3.1 Qualitative evaluation

For each year in NIPS, Table 5 lists the paper with the highest in-degree in the influence graph computed by the LRT method with $k = 100$ and $l = 10$. We expect these to have high citation counts, which we test by showing the paper's citation counts both from within the NIPS corpus (as of 2000) and from Google Scholar (as of 2007). For most documents, the citation count is indeed high when compared to the average NIPS document citation count of 0.7734 other NIPS papers. An interesting example is "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing" from 1996. While this is one of the papers that introduced SVMs to NIPS, it has only 3 citations within NIPS and only 44 citations in Google Scholar. Nevertheless, SVMs had a huge impact on NIPS. In this sense our LRT method is correct and is not influenced by citation habits. In this example, most authors cite Vapnik's later book (with 5144 citations) instead of this paper. The LRT method is unaffected and correctly identifies the SVM idea as highly influential on NIPS.

Document		Citation Counts	
Year	Document Title and Author(s)	NIPS	Google Scholar
1988	"An Optimality Principle for Unsupervised Learning" by Terence D. Sanger	4	61
1989	"Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters" by John S. Bridle	11	113
1990	"Learning Theory and Experiments with Competitive Networks" by Gniff L. Bilbro, David E. van den Bout	0	0
1991	"The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems" by John Moody	12	234
1992	"Reinforcement Learning Applied to Linear Quadratic Regulation" by Steven J. Bradtke	6	56
1993	"Supervised Learning from Incomplete Data via an EM approach" by Zoubin Ghahramani, Michael I. Jordan	12	163
1994	"Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems" by Tommi Jakkola, Sizarad Singhal, Michael I. Jordan	10	142
1995	"EM Optimization of Latent-Variable Density Models" by Chris M. Bishop, M. Svensen, Chistopher K.I. Williams	1	27
1996	"Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing" by V. Vapnik, Steven E. Golowich, Alex Smola	3	44 (5144)
1997	"EM Algorithms for PCA and SPCA" by Sam Roweis	1	177

Table 5: The most influential paper per year in NIPS, as measured by influence graph in-degree, with $k = 100$, $\sigma = .05$, and TFIDF cosine for $\mathcal{C}$. We exclude years with edge effects and the last 3 years, since they do not have statistically significant counts. Comparison is against the within-NIPS citation counts, and Google-scholar citation counts (on Feb. 28, 2007).

Corpus	TF						TFIDF					
	LRT			SIM			LRT			SIM		
	τ	RMap@3	@12	τ	RMap@3	@12	τ	RMap@3	@12	τ	RMap@3	@12
NIPS	0.4216	0.2771	0.3126	0.3379	0.1475	0.2561	0.4163	0.2751	0.3022	0.3686	0.1959	0.2585
HEPTH	0.3887	0.2558	0.2376	0.3497	0.1421	0.1594	0.3549	0.1456	0.1582	0.3190	0.1139	0.1138

Table 6: Rank metrics comparing the LRT against similarity on NIPS ($k = 100$) and HEPTH ($k = 20$), using $\sigma = .05$ and TF or TFIDF cosine for $\mathcal{C}$. We ignore the first two and last two years because of edge effects.

4.3.2 Quantitative Evaluation

We compare the ranking of documents by in-degree in the influence graph to the ranking by citation count. As similarity measures, we use Kendall's τ and a ranking version of MAP, which we term R-MAP.

Kendall's τ.

Kendall's τ measures how many pairs two rankings rank in the same order. It ranges between -1 and 1, with higher numbers indicating greater similarity. Formally,

$$\tau = \frac{2 \cdot \text{ number of concordant pairs}}{\text{total number of pairs } - \text{ number of tied pairs}}$$

R-MAP@k.

R-MAP@k measures the average precision of a ranking. With the k top-ranked documents as positive examples, average the ranking's precision at the positions of these documents. We calculate R-MAP@3 and R-MAP@12.

There is one caveat with rank-based metrics. Edge effects (e.g., older papers have more citations, papers from the last year have no citations) make it difficult to present one unified ranking of all documents. Therefore, we calculate each metric per-year and average the year-by-year values to get a single score for the entire corpus. Additionally, because of edge effects, the first two and the last two years are not used, since they do not contain meaningful results.

The TF and TFIDF baselines use the most similar documents instead of the LRT predictions.

LRTs are better than similarity.

Table 6 shows that the LRT gives substantially better rankings than the similarity baseline for all metrics on both NIPS and HEPTH with both TF and TFIDF cosine $\mathcal{C}$.

Effect of the parameter l.

The left plot of Figure 4 explores whether selecting influencers is sensitive to the parameter l. For the influence graph, we considered each document's l predicted influencers with highest LRT scores. Figure 4 shows how varying l affects τ for both LRT and the similarity baseline. Since NIPS documents do not have many citations, we explore $l = 1$ to 15. The upper line is LRT performance with 95% confidence interval error bars. (The confidence interval is computed using the multiple τ values per data point, because each graphed τ is the average of multiple (here, 10) years of τ metric scores.) The lower line depicts τ on the similarity baseline. For the TFIDF cosine $\mathcal{C}$, when l is small, the method computes a count over only the few top influential documents selected by the LRTs for $d^{(new)}$. It turns out that small l seem to perform better than our initial guess of $l = 10$. As l increases, more non-influential documents are counted and τ correspondingly falls. When l approaches 100 (not shown), the LRT and the baseline are identical as expected by construction.

Thresholding on the LRT score.

The right plot of Figure 4 depicts how τ varies if we do not select a fixed number of l neighbors per document, but instead use a threshold on the LRT statistic. The LRT is set up to reject the null hypothesis and declare that $d^{(can)}$ influences $d^{(new)}$ if the LRT statistic is sufficiently large. Varying this threshold controls the level of confidence in the LRT, so we use the threshold level as the x-axis and examine how it affects τ. Thresholding the LRT values actually gives better performance than using the l parameter, since we are not forcing a certain number of influence links for each document. There are four different regions in this graph. First,

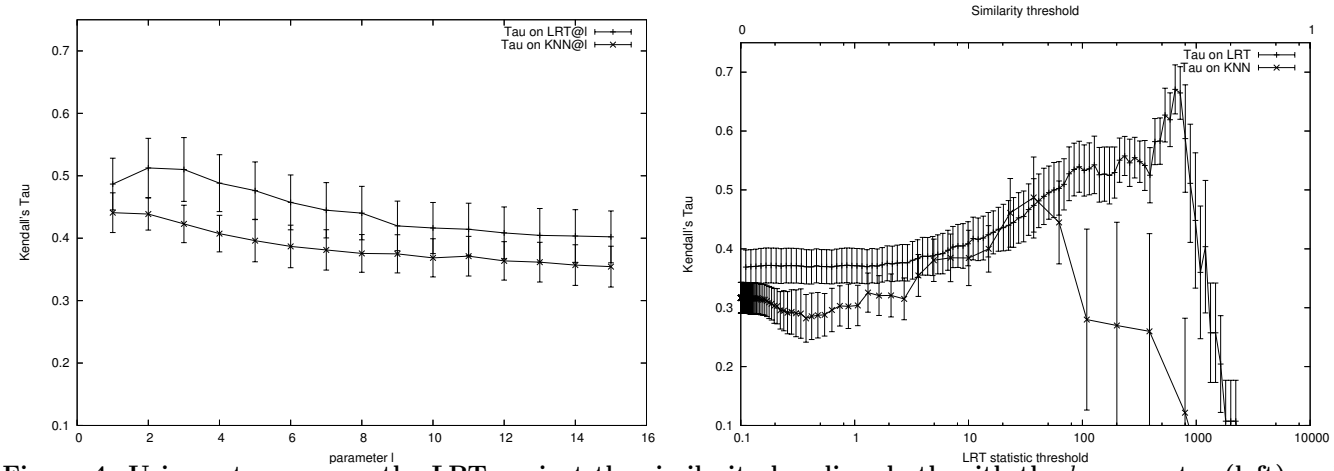


Figure 4: Using τ to compare the LRT against the similarity baseline, both with the l parameter (left) and by thresholding the LRT statistic values (right). Results are for NIPS with TFIDF cosine $\mathcal{C}$ and $k = 100$. The TF plot looks similar, except that the baseline is smoother.

if the threshold is too low, performance suffers because the null hypothesis is being accepted erroneously. Second, performance increases as the threshold approaches reasonable confidence levels. Third, a large range of threshold values (approximately 100-2000) give good and similar τ scores, showing that the LRT method is robust. Fourth, when the threshold is too high, many influential documents are no longer detected, and performance subsequently falls.

Note that a confidence level of 95% per test (i.e. a threshold of 3.84) performs quite poorly. This level means that 5% of the influence links are erroneous. NIPS, with 2000 papers, would have an expected 100,050 false links (and only 1512 real citations). Therefore, we need a much higher confidence level to account for the multiple-testing bias. Using Bonferroni adjustment, each test's level is the overall level divided by the number of tests.

5. DISCUSSION AND FUTURE WORK

One obvious limitation of the current model is the simplicity of the language model. The assumption that each document is a sequence of independent words is, in reality, likely violated. This observation motivates more expressive language models such as n-gram language models.

There is also the question of whether these methods can generalize to other domains. LRTs do not use citation data, so many domains should be applicable. However, we have only conducted experiments on research publications.

Finally, there is scalability and efficiency. Much of the computing time is spent solving convex optimization problems. While $\mathcal{C}$ and $\mathcal{P}$ prune this space, there may be other criteria to provably eliminate certain LRTs without affecting the results. Furthermore, the optimization problems have a special structure, which can probably be exploited by specialized methods to solve the optimization problems.

6. CONCLUSIONS

We presented a probabilistic model of influence between documents for corpora that have grown over time. In this model, we derived a Likelihood Ratio Test to detect influence based on the content of documents and showed how the test can be computed efficiently. We found that the influence graphs derived from the content resemble the structure of explicit citation graphs for corpora of scientific literature. Furthermore, we showed that in-degree in the influence graph is an effective indicator of a document's impact. The ability to create influence graphs based on document content alone has the potential to open databases without explicit citation structure to the large repertoire of graph mining algorithms.

7. ACKNOWLEDGMENTS

We thank Johannes Gehrke and Rich Caruana for the discussions that lead to this work. This work was funded in part by NSF Career Award IIS-0237381, NSF Award OISE-0611783, and the KD-D grant.

8. REFERENCES

[1] M. Agosti and J. Allan. Introduction to the special issue on methods and tools for the automatic construction of hypertext. *Inf. Process. Manage*, 33(2):129–131, 1997.

[2] M. Agosti and F. Crestani. A methodology for the automatic construction of a hypertext for information retrieval. In *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing*, pages 745–753, Indianapolis IN, Feb. 1993.

[3] M. Agosti, F. Crestani, and M. Melucci. On the use of information retrieval techniques for the automatic construction of hypertext. *Information Processing and Management*, 33:133–144, 1997.

[4] J. Allan. *Automatic Hypertext Construction*. PhD thesis, Cornell University, 1995.

[5] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic Detection and Tracking Pilot Study: Final Report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop-1998*, 1998.

[6] J. Allan, R. Papka, and V. Lavrenko. On-Line New Event Detection and Tracking. In *Research and Development in Information Retrieval*, pages 37–45, 1998.

[7] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Info. Processing Systems*, 2005.

[8] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[9] G. Casella and R. L. Berger. *Statistical Inference*, chapter 10.3.1 Asymptotic Distribution of LRTs, pages 488–492. Duxbury, 2002.

[10] J. H. Coombs. Hypertext, full text, and automatic linking. In *Proc. Thirteenth Int'l. Conf. on Res. and Development in Information Retrieval*, Hypertext and Image Retrieval, page 83, 1990.

[11] R. Furuta, C. Plaisant, and B. Shneiderman. A spectrum of automatic hypertext constructions. *Hypermedia*, 1(2):179–195, 1989.

[12] E. Garfield. The Meaning of the Impact Factor. *International Journal of Clinical and Health Psychology*, 3(2):363–369, 2003.

[13] P. Ginsparg. The physics e-print arxiv. http://www.arxiv.org.

[14] R. Guha, D. Sivakumar, R. Kumar, and R. Sundaram. Unweaving a Web of Documents. In *Proceedings of KDD-2005*, Chicago, Illinois, 2005.

[15] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: In Search of Trends, Patterns, and Relationships. *IEEE Transactions on Visualization and Computer Graphics*, 2002.

[16] T. Hofmann. Probabilistic latent semantic analysis. In *Fifteenth Conference on Uncertainty in Artificial Inteligence (UAI)*, 1999.

[17] http://nips.djvuzone.org/txt.html. NIPS Online: The Text Repository.

[18] http://www.mosek.com/index.html. Mosek.

[19] F. Jelinek. *Statistical Methods for Speech Recognition*, chapter Basic Language Modeling, pages 57–78. MIT Press, 1998.

[20] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.

[21] J. Kleinberg. Bursty and Hierarchical Structure in Streams. In *Proceedings of KDD-2002*, Edmonton, Alberta, Canada, 2002.

[22] T. Kolenda, L. K. Hansen, and J. Larsen. Signal Detection using ICA: Application to Chat Room Topic Spotting. In Lee, Jung, Makeig, and Sejnowski, editors, *Proc. of the Third International Conference on Independent Component Analysis and Signal Separation (ICA2001)*, pages 540–545, San Diego, CA, USA, 2001.

[23] A. Krause, J. Leskovec, and C. Guestrin. Data association for topic intensity tracking. In *International Conference on Machine Learning (ICML)*, 2006.

[24] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Special Interest Group on Information Retrieval (SIGIR)*, pages 194–201, 2004.

[25] O. Kurland and L. Lee. Respect my authority! hits without hyperlinks, utilizing cluster-based language models. In *Special Interest Group on Information Retrieval (SIGIR)*, pages 83–90, 2006.

[26] A. Lelu. Automatic generation of hypertext links in information retrieval systems: A stochastic and an incremental algorithm. In V. V. Bookstein, A.; Chiaramella, Y.; Salton, G.; Raghavan, editor, *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 326–336, Chicago, Ill., USA, Oct. 1991. ACM Press.

[27] G. Mann, D. Mimno, and A. McCallum. Bibliometric impact measures leveraging topic analysis. In *Joint Conference on Digital Libraries (JCDL)*, 2006.

[28] C. D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[29] Q. Mei and C. Zhai. Discovering Evolutionary Theme Patterns from Text - An Exploration of Temporal Text Mining. In *Proceedings of KDD-2005*, Chicago, Illinois, 2005.

[30] F. Osareh. Bibliometrics, Citation Analysis and Co-citation Analysis: A Review of Literature I. *Libri*, 46:149–158, 1996.

[31] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Computer Science Department, Stanford University, 1998.

[32] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[33] G. Salton and C. Buckley. Automatic text structuring and retrieval – experiments in automatic encyclopedia searching. In *Proc. Fourteenth Int'l. Conf. on Res. and Development in Information Retrieval*, Document Structure, page 21, 1991.

[34] B. Shaparenko, R. Caruana, J. Gehrke, and T. Joachims. Identifying temporal patterns and key players in document collections. In *IEEE ICDM Workshop on Temporal Data Mining: Algorithms, Theory and Applications (TDM-05)*, pages 165–174, 2005.

[35] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Knowledge Discovery and Data-Mining (KDD)*, 2004.

[36] R. Swan and D. Jensen. TimeMines: Constructing Timelines with Statistical Models of Word Usage. In *Proceedings of KDD-2000*, pages 73–80, Boston, MA, 2000.

[37] X. Wang and A. McCallum. Topics over time: A nonmarkov continuoustime model of topical trends. In *Knowledge Discovery and Data Mining (KDD)*, pages 424–433, 2006.

[38] R. Wilkinson and A. F. Smeaton. Automatic link generation. *ACM Computing Surveys (CSUR)*, 31(4es), 1999.

[39] C. Zhai. *Risk Minimization and Language Modeling in Information Retrieval*. PhD thesis, Carnegie Mellon University, 2002.