Enabling Author-Centric Ranking of Web Content

Muneeb Ali and Andrea S. LaPaugh

Princeton University
{muneeb,aslp}@cs.princeton.edu

ABSTRACT

In the past decade there has been a rapid increase in the amount of web content, partially due to wide-spread adoption of blogging and micro-blogging platforms that make it easy for ordinary users to quickly create web content. A single user may create content on many different online platforms or social networks. Current web search systems largely rely on link-structure of the web graph in ranking content. In other words, they care about "who links to a webpage" and are largely agnostic to "who authored the content on this webpage". In this paper, we argue that in addition to link-structure the topic-authority of the original author can provide important signals for ranking and discovery of web content. We present algorithms for transferring topic-authority between webpages based on a same-origin policy. Our preliminary results suggest that such topic-authority transfers can improve the visibility of certain types of content compared to a link-structure only approach.

Keywords

Web Search, Ranking, Topic Models, Algorithms

1. INTRODUCTION

Online information discovery is changing fundamentally for the first time since link-structure based algorithms, like HITS and PageRank [12, 14], were introduced in the late 90s. Humans now actively interact with online data (e.g., by posting links, commenting, or voting up content) instead of passively consuming data. An "online identity" is emerging for people and users typically have multiple online profiles (personal blog, Twitter, Google Scholar etc.), which are sometimes explicitly linked to each other. Online services like Twitter, and Facebook have hundreds of millions of active users and they allow these users to sign into other websites and services using social identities e.g., users can connect their Twitter accounts to question/answer sites like *Quora.com* or blogging sites like *Medium.com* and then produce new content there. Given this authorship information for content, the question we ask in this paper is: *how can authorship information be used to better rank online content?*

We believe that we are standing at a point in the evolution of the Internet where it's now possible to use author-centric information for ranking online content. In fact, in this paper we make the case that author-centric web ranking has the potential to become the dominant ranking method for web content. This can mark the first major shift from link-structure based ranking. Link-structure will not become irrelevant, but rather link-structure can be used alongside authorship information to give better ranks. Further, author-centric ranks can help discover fresh content faster as it's not necessary to get a lot of incoming high quality links before the content gets a high rank in search results.

Authorship information and associated author profiles and quality signals are a new dimension of information that's now available to aid analysis of online data. However, how exactly to use such signals remains a relatively unexplored problem. There are many research challenges that need to be solved before author-centric ranking can see widespread adoption. The first question is how exactly to calculate author-centric ranks for online content. Also, how do such ranks relate to the authority/influence of the underlying author(s), and how do they change over time. Further, how we can verify authorship of content online is not clear.

In this paper, we take a first stab at some of these questions and present the *AuthLink* algorithm that gives a simple method for calculating author-centric ranks while leaving room for further research on variations on our initial design choices. We focus on web graphs and look at authorship of data for web graphs. One key insight is that humans post content on different platforms and domains where as their expertise is the same regardless of the means of publishing data. Given that we can verify the authorship of content (a separate research problem in itself), we can then better rank the content by factoring in the topic authority of the underlying author. We present the design and evaluation of a novel rank-transfer algorithm that looks at "authorship links" across data from multiple web sources and calculates new ranks that include authorship information in addition to other factors. We use *probabilistic topic models* [5] for calculating topic authority, but our algorithm is agnostic to what exact method is used.

To evaluate our core ideas, we compare search rankings, on a corpus that we collected, for link-structure based ranking and our author-centric ranking. Our preliminary results suggest that author-centric ranking, using topic models for transferring topic authority, is a powerful new concept that has the potential to give better search rankings and re-define how people search for information online. Further, author-centric rankings can have applications beyond web search e.g., in digital libraries, crowd sourcing, and in assigning quality scores to human endorsements (e.g., on LinkedIn).

Sixteenth International Workshop on the Web and Databases (WebDB 2013), June 23, 2013 - New York, NY, USA.

2. DESIGN GOALS & SCOPE

In this section we define our high level design goals:

• **Low Complexity:** We are interested in a generic and simple algorithm. We believe that our work is one of the first to explicitly consider author-centric ranking and the initial approach should be simple: later on, we can add more complexity as needed and as guided by experiment results.

• **Extensibility:** Related to the goal of low complexity is the ease of extensibility. The algorithm should be agnostic to specific underlying methods, like choice of topic modeling, and should be easily extensible.

• **Scalability:** The algorithm should scale to many millions to billions of nodes. The rate at which online data, and the respective graph size, is growing requires algorithms to be highly scalable for them to be practical.

3. AUTHLINK ALGORITHM

In this section we describe the *AuthLink* algorithm. We first present the basic idea and then build on the simple case.

3.1 Basic Idea

Assume that we have a graph G with V_D vertices (online documents) and E edges. In this paper we assume G to represent a Web graph, but the algorithm can work on any graph. We are interested in obtaining a graph G' by adding vertices $a' \in V'_A$ and pairs (a', d) where $a' \in V'_A$ and $d \in V_D$. Pairs (a', d) represent *authorship links* E'_A not originally present in graph G i.e., node a' is the author of document d. In other words, we assume that there are some "overlay" vertices and edges that are not explicitly defined in the Web graph. Since they are not explicit hyperlinks but are implicit *authorship links*, they are not considered in calculating ranks by traditional algorithms like PageRank [14] and HITS [12]. For simplicity we separate the *author nodes* from the original graph and represent them as additional nodes, one for each author/person, although it's possible to use a representative page e.g., the homepage of a person as the *author node*.

After constructing graph G', the next task is to calculate the new scores for G'. A trivial way of doing this can be to simply run PageRank or HITS on the new graph. However, we believe that humans are influential in only certain topics and transferring topic-specific author-influence makes more sense. We propose a new algorithm that transfers topic-specific scores over the links E'_A. The run time of our algorithm is independent of the original graph size of G: it only depends on the score transfers concerning V'_A and E'_A.

The AuthLink algorithm is divided into three steps: a) *graph construction*, in which we construct G' by adding author nodes V'_A and respective authorship links to the original graph G, b) *topic authority*, in which we find the underlying topic models for documents V_D and infer topic authority scores for authors V'_A, and c) *score transfers*, in which we transfer topic scores from author nodes to the content they've authored according to certain criteria. These steps are described in more detail in the next sections respectively.

3.2 Graph Construction

The graph construction step takes as input a graph G with V_D vertices and E_D edges, a set of vertices V'_A and pairs (a', d) where $a' \in V'_A$ and $d \in V_D$, and for every $a' \in V'_A$ there is a pair (a', d). Given these conditions are true for the input, the graph construction is fairly simple. We first add vertices in V'_A to G and then add pairs (a', d) to G, and get a new graph G' as described in Algorithm 1. Figure 1a and Figure 1b show an example of such graph construction.

Data: Graph G with vertices V_D, Vertices V'_A, Pairs (a', d) where $a' \in V'_A$ and $d \in V_D$
Result: Graph G'
Initialization: copy G to G', $\forall d \in V_D$ set $d_{author} = 0$;
for *every pair* (a', d) **do**
  if $a' \notin G'$ **then** add a' to G';
  add (a', d) to G';
  $d_{author} = d_{author} + 1$;
end
for *every vertice* $d \in V_D$ **do**
  if $d_{author} \geq 2$ **then**
    remove respective author edges;
    let super author s' represent multiple authors;
    if super author $s' \notin G'$ **then** add s' to G';
    add edge (s', d);
    $d_{author} = 1$;
  end
end

Algorithm 1: Graph Construction

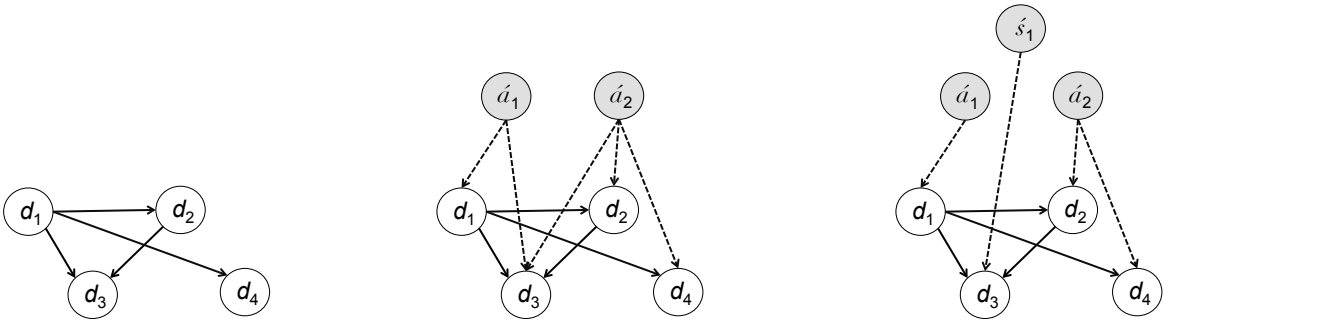
Although the basic graph construction is fairly straightforward, we need to take special care if there are more than one authors of the same content. In Figure 1(b), both a'_1 and a'_2 have *authorship links* to content node d_3. This can be problematic for convergence of score transfers (see Section 3.4) and we don't want such overlap. To solve this problem, we introduce the concept of *super authors*: nodes that represent two or more authors. We construct *super authors* as a superset of co-author nodes (Figure 1c). Inferring topic scores for *super authors* is different from calculating topics scores for author nodes (see Section 3.3.1 for details). Algorithm 1 marks the number of authors for each content node d and if there are more than one authors, it removes the respective author edges and adds a *super author* node s' instead.

3.3 Topic Authority

Our algorithm requires topic-specific scores for each author, but is agnostic to what exact scoring method is used. For our work, we use *topic modeling* [5] to calculate scores. Topic models are algorithms for discovering the main themes underlying a unstructured collection of documents. More specifically, we use a probabilistic hidden topic model called LDA [6]. In probabilistic topic modeling, each document is viewed as a distribution over topics and gets a probability score for each topic. This is not a partition of documents into topics, and the topic probabilities don't add to 1.

$$R_k(d) = \hat{\beta}_k(d) \log \left(\frac{\hat{\beta}_k(d)}{\left(\prod_{j=1}^{K} \hat{\beta}_j(d) \right)^{\frac{1}{K}}} \right) \quad (1)$$

The topic probabilities can be ordered to get the top-k topics. The simplest way to get topic scores is to use the per-topic probabilities $\hat{\beta}_k(d)$ for topic k in document d as given by LDA. However, we use the modified score $R_k(d)$ given by Equation 1, which re-scales the scores based on



a) A given graph G with webpages as nodes and hyperlinks as edges

b) Addition of author nodes $a'_1, a'_2 \in V'_A$ and respective authorship edges

c) Construction of super author s'_1 from author nodes a'_1, a'_2 to get graph G'

Figure 1: AuthLink Pre-processing Step: Constructing $G^{'}$ from G

how large the probability of topic k is relative to all topics. More details are in [5]. In Equation 1, K denotes the number of topics in the model. We rescale $R_k(d)$ to (0-100) to get topic scores for documents.

$$R_k(a') \; for \; a' \in V'_A = \frac{\sum R_k(d) \; for \; d \in C}{|C|} \tag{2}$$

Topic-specific score $R_k(a')$ for author a' is different from topic-specific score $R_k(d)$ for document d as $R_k(a')$ is inferred from topic scores of all documents authored by a' respectively. Given C the set of children of a' (the documents authored by a'), the topic-specific author score $R_k(a')$ for each topic k is calculated by Equation 2 and the general procedure is described in Algorithm 2. The topic scores for authors depend on the respective topic scores of the content they authored, while ignoring topic scores below a threshold λ. The rationale for ignoring scores below a threshold is that an author might be influential about a topic, but the document might not be about that topic. Final scores are calculated by a simple averaged sum, as given in Equation 2. We discuss alternate scoring methods in Section 7.

Data: Graph G' without topic scores, document vertices V_D, author vertices V'_A
Result: Graph G' with initialized topic scores
Initialization: run $LDA(V_D)$;
for *every node* $d \in V_D$ **do**
  calculate $R_k(d)$ using Equation 1;
  re-scale $R_k(d)$ between (0-100);
end
for *every node* $a' \in V'_A$ **do**
  let C be the set of children of a';
  for *every topic* k **do**
    ignore $R_k(d) \in C$ values $\leq$ threshold λ;
    calculate $R_k(a')$ using Equation 2;
    re-scale $R_k(a')$ between (0-100);
  end
end

Algorithm 2: Topic-specific Scores

3.3.1 Super Authors

Scoring topics for *super authors* is different from scoring topics for author nodes $a' \in V'$. The topic-specific scores of *super authors* are inferred from the topic scores of the respective co-authors that are combined to construct the *super author*. Given topic-specific scores $R_k(a')$ for topic $1 \ldots K$ and $a' \in V'$, the respective topic-specific scores for a *super author* s' can be calculated as:

$$R_k(s') = \max[R_k(a')] \; for \; a' \in sub(s') \tag{3}$$

where $sub(s')$ is the set of authors that the *super author* is constructed from i.e., the actual co-authors. We are using a *max* function because we believe that when authors collaborate they complement the expertise of each other and the resulting content will have the respective topic influence of both. This is obviously an assumption and $R_k(s')$ can be calculated using other methods (see Section 7).

3.4 Score Transfers

Both the graph construction and topic scoring can be considered as pre-processing steps for the score transfer method, described in Algorithm 3.

Data: Graph G' with scores, author vertices V'_A, content vertices V_D
Result: Graph G' with new scores
Initialization: $V' = V'_A \cup \{\text{all super authors } s'\}$
for *every node* $a' \in V'$ **do**
  let C be the set of children of a';
  for *every node* $d \in C$ **do**
    if $R_k(d) \leq$ threshold λ **then** skip;
    else calculate new $\hat{R}_k(d)$ using Equation 4
  end
end

Algorithm 3: Score Transfers

$$\hat{R}_k(d) = \{\omega_d \times R_k(d)\} + \{\omega_a \times R_k(a')\} \tag{4}$$

The score transfer algorithm basically calculates new topic scores for documents d by transferring topic authority from author nodes for specific topics. The algorithm ignores scores below a certain threshold λ and leaves them unchanged. The rationale is that an author might be influential about a certain topic, but if the document is not about that topic then we don't want to transfer authority to the document. Weights ω_d and ω_a in Equation 4 determine how new topic

scores $\hat{R}_k(d)$ are calculated. Different variations for choosing weights are possible, but for our preliminary evaluations we use a simple averaged sum approach while ensuring that $\hat{R}_k(d) \not< R_k(d)$. The reason for this choice is mostly simplicity: the topic scores can only go up because of author influence and there cannot be any negative effects of authorship information. We discuss other variations in Section 7.

4. EVALUATION

In this section we describe our preliminary evaluation.

4.1 Data Collection

For our evaluation we needed a dataset of people and the respective content they have authored online. In other words, we needed a dataset of $a' \in V'_A$ vertices (authors) and pairs (a', d) (authorship links) between authors $a' \in V'_A$ and documents $d \in V_D$ that we can then combine with a web graph G to construct G'. Verifying authorship of content is an open research problem (see Section 7) and collecting authorship pairs (a', d) from the web graph is non-trivial.

We got around the problem of verifying authorship of content by collecting data from an online service *About.me* [2] that provides users with tools to connect their various online identities, like webpages, social networks, and blogs etc., into a single profile. For most connected sources *About.me* users have to explicitly verify ownership of content or explicitly authenticate their social network profiles. This enabled us to collect a dataset with verified authorship pairs (a', d). For profiles/links that can be connected without any authentication, we assume that the users behaved in good faith. We believe that better methods for verification of authorship are needed (see Section 7), but for our preliminary study the *About.me* dataset is a reasonable start. Explicitly verifying authorship links is beyond the scope of this paper.

We collected 110,532 profiles (authors) from *About.me* which had 462,353 outgoing links to various web content. We collected all content nodes pointed to by the 460K links in our dataset. We performed experiments with both unfiltered and filtered/processed content nodes. For the unfiltered dataset we saved the respective HTML files without looking at what those files contain. For the processed dataset we a) replaced HTML files from *Twitter.com* by the tweets obtained from the Twitter API [16] for the respective users, b) replaced *LinkedIn.com* profiles by the "Full Profile" of the respective users, c) replaced links to Pinterest "login required" error pages with profiles obtained from our custom Pinterest crawler (that performs login before fetching content), and d) ignored all links to Facebook, which had very little information from public profiles. We also ignored links to certain image-heavy sites like *Behance.com* that did not have enough text content for topic modeling. We plan to make this dataset available online.

4.2 Score Calculations:

For topic modeling, we used LDA [6] and more specifically the *gensim* [15] distributed implementation of LDA. A single run of our implementation of Algorithm 2 can give topic-specific scores for author (and *super author*) nodes. However, we believe that in practical scenarios frequent updates will be made on such topic scores as new content or links are discovered by web crawlers. To test how our algorithm performs with incremental addition of new content/links we incrementally added content nodes from our dataset to our working set and studied the change in topic-specific scores for authors. Our preliminary results show that scores changes make logical sense to human observers given the topics/content of each additional content node.

4.3 Search Experiments

To evaluate the effectiveness of *AuthLink*, we compared a default search engine using link-structure based ranks to a search engine using author-centric ranks. For our experiments, queries are generated synthetically (and randomly) and 3-4 related queries are combined into individual sets e.g., queries 'CDN', 'content distribution network', and 'content distribution networks' will belong to one query set and we report the average results for each query set. We used 10 such *query sets*. For ranking search results, topics are extracted from the query and then a variation of *cosine similarity* between query topics and document topics is used for scoring.

Comparison Sets: Our base ranking method is a variation of the default scoring method of Lucene [1]. For author-centric ranks, we used topic-specific scores from *AuthLink*. To get a more complete picture, we added a third ranking method that uses purely LDA-based topic scores. For our experiments the data corpus was the *About.me* processed dataset described earlier. Lucene by default does not use PageRank in scoring documents. We provided PageRanks of indexed pages as an additional scoring factor to Lucene at index time and adjusted the relative importance of PageRanks in the final ranks by using weights as given below:

$$rank(q, d) = \{\omega_p \times PageRank(d)\} + \{\omega_s \times score(q, d)\} \quad (5)$$

where $PageRank(d)$ is the actual PageRank as reported by Google for the webpage. We obtained PageRanks from Google for our dataset in January 2013 using Google's toolbar service for reporting PageRanks. The $score(q, d)$ is the only variable factor in the three ranking methods. In Lucene, the $score(q, d)$ of document d for query q is given by a variant of *cosine similarity* along with other factors like document length, query/document boost factor, and adjustments for the relative importance of multiple terms in queries. Details on Lucene's scoring are at [1]. For *AuthLink* and LDA, the $score(q, d)$ is the *cosine similarity* between the topics in query q and topic scores given by *AuthLink* and LDA for documents d respectively. We perform some processing on the topics for query q to adjust the vector size and include hidden/related terms as otherwise the query topic vector had very sparse data in it.

Using topic models in search rankings is a relatively new concept, but our preliminary results show that *cosine similarity* of $topics(q)$ and $topics(d)$ can give comparable results to other methods. Further, some experiments by Search Engine Optimization (SEO) companies suggest that LDA *cosine similarity* of $topics(q)$ and $topics(d)$ and Google's proprietary search rankings are remarkably correlated [10].

Gold Standard: Evaluating search rankings has this inherent limitation that there is no "gold standard" to compare the results to. For our experiments we constructed two human-judgement gold standards, one from student volunteers (g_s) and the other using Amazon *Mechanical Turk* [3] (g_m). For g_s we asked 10 student volunteers to hand score 40 results per query set. To reduce hand scoring workload, we did not ask them to score individual queries but provided

	Gold standard g_s				Gold standard g_m			
	Generic τ	LDA τ	AuthLink τ	Difference from (Generic, LDA)	Generic τ	LDA τ	AuthLink τ	Difference from (Generic, LDA)
Queryset S_a	0.21	0.12	0.37	(0.16, 0.25)	0.43	0.22	0.35	(-0.08, 0.13)
Queryset S_b	0.76	0.72	0.74	(-0.02, 0.02)	0.86	0.74	0.81	(-0.05, 0.07)
Queryset S_c	0.42	0.61	0.67	(0.25, 0.06)	0.64	0.49	0.62	(-0.02, 0.13)
Queryset S_d	-0.14	0.37	0.32	(0.46, -0.05)	0.25	0.34	0.29	(0.04, -0.05)
Queryset S_e	0.17	-0.02	0.29	(0.12, 0.31)	0.32	0.09	0.37	(0.05, 0.28)
Queryset S_f	0.35	0.25	0.42	(0.07, 0.17)	0.39	0.31	0.44	(0.05, 0.13)
Queryset S_g	0.12	-0.03	-0.11	(-0.23, -0.08)	0.38	0.08	0.06	(-0.32, -0.02)
Queryset S_h	-0.21	0.24	0.27	(0.48, 0.03)	-0.03	0.15	0.13	(0.10, -0.02)
Queryset S_i	0.89	0.81	0.78	(-0.11, -0.03)	0.92	0.76	0.79	(-0.13, 0.03)
Queryset S_j	0.92	0.83	0.88	(-0.04, 0.05)	0.88	0.86	0.90	(0.02, 0.04)

Table 1: τ difference between Generic ranks and AuthLink ranks (compared to gold standards)

them with a merged set of search result (40 documents) per query set. Each student volunteer hand scored all 10 query sets and then we averaged the results. For g_m we asked workers on *Mechanical Turk* to choose between two options for a given query i.e., we presented the *Human Intelligence Task* (HIT) as a question of the form "is document d_1 more relevant than document d_2 for query q? Yes or no". We did not ask *Mechanical Turk* workers to hand score the entire list because experience with *Mechanical Turk* suggests that it is best to keep a HIT as simple as possible and to limit possible answer [3]. We processed the preference pairs from completed HITs into a single ranking order. To limit the no. of rank pair preference questions, we used Lucene's default results as a base order and then produced rank pairs for ± 5 results. We used 30 *Mechanical Turk* workers to score 2000 unique rank pairs with a redundancy of three. The answer with two or more votes was considered the final answer for each rank pair. The cost of the 6000 HITs around $30 USD.

Ranking Results: Given a *gold standard*, search rankings can be compared using the *Kendall tau coefficient* (τ) which is a measure of rank correlation, i.e., the similarity of the orderings of the data. If P is the no. of rank pairs that agree and Q is the no. that disagree, then Kendall's τ is:

$$\tau = \frac{P - Q}{P + Q} \tag{6}$$

τ varies between 1 if all pairs agree, to -1 when they all disagree. For our experiments, positive values of τ mean that there is positive correlation between the *gold standard* and the respective ranking method. Table 1 shows the τ difference between three search ranking methods relative to the respective *gold standard*. The *Generic* ranking uses Lucene (ver 3.6.2) as described above. For our experiments, we kept the weights ω_p and ω_s constant in Equation 5 for any single run. We tried values between 20%-80% for ω_p. The results reported in Table 1 use the default value of 40% for ω_p i.e., 40% of the final rank is determined by PageRank.

Two important observations from Table 1 for *gold standard* g_s are that a) in most cases where *AuthLink* is worse than *Generic*, *Generic* itself has a high value of τ; if we ignore such cases then there is only one case where *AuthLink* is worse and *Generic* itself has a low value of τ, and b) *AuthLink* is never significantly worse than *LDA*, but it is on average better and sometimes significantly better than *LDA*. For the *Mechanical Turk* based *gold standard* g_m, we notice that *Generic* performs much better on this standard. We believe that this is because Lucene's ranking was used as a base for making rank pairs (± 5 results) for *Mechanical Turk* workers. We also notice that the difference of *AuthLink* from *LDA* is in the same order of magnitude. There are very few negative τ values when using g_m suggesting that the student hand scored g_s was more different from both Lucene and others, whereas there is more correlation between g_m and Lucene (for reasons we discussed) and also between g_m and *AuthLink* and *LDA* respectively. Finally, these are preliminary results on a relatively small experiment and hidden properties of the individual query sets and variations in the gold standard values make it hard to make generalizations. However, we believe that these initial results are promising enough to warrant further investigation.

5. RELATED WORK

A detailed discussion of the literature is beyond the scope of this paper. We refer readers to [5] for a detailed discussion on topic models and to [8] for a discussion about the latest advancements in search technologies. There have been some previous attempts to use social signals (from social networks) in search rankings [13] that look at combining social signals with link-structure, but don't look at authorship information of web pages. There are also some search systems that use humans to answer either all queries [11] or a subset of rare queries [4]. The focus of our work is on authorship information of web content and not to use human directly for searching. Researchers have also looked at temporal ranking in social graphs [9]. Some commercial services like *Klout.com* and *PeerIndex.com* give users influence scores and top topics per user, but they currently don't give topic-specific influence scores as we do.

6. OPEN PROBLEMS

This paper opens up more questions than it answers:

• **Authentication:** The biggest roadblock towards author-centric ranking is the current lack of authorship authentication online. Webpages and other online content seldom have verified authorship information and when author information is available e.g., in the case of social networks like Twitter, it is hard to link the same authors from multiple resources. In other words, how do we verify that Jeffrey Ullman is the author of the book "Principles of Database and

Knowledge-Base Systems" and is also the author of the website at `http://infolab.stanford.edu/~ullman/` and the *Google+* profile `http://gplus.to/JeffUllman`. There has been some recent progress towards "online identity" with people using services like Facebook Connect or Google to login to other services. Google's Authorship product enables users to verify their email on a domain and then tell Google's crawlers to link new content posted on that domain to their Google profile. This is an encouraging step towards authentication, but much work remains in this area.

• **Multiple Authors:** Our current algorithm works with web content contributed by multiple authors assuming that the entire content of the document was co-authored (much like an academic research paper). However, having multiple authors contributing web content is not like co-authors of an academic paper at all. Some authors might write parts of the page, while others might write other part independently e.g., the front page of a news website like TechCrunch will have individually contributed content from different authors. Further, it's quite common for online users to leave comments on a webpage. We currently ignore all comments in our algorithm (and removed comments from webpages in our dataset), but these issues need to be explored more.

• **Spam:** Author-centric ranking based on topic-models introduces new opportunities for spam where a spammer can put in a lot of keywords in a document in the hope that the topic-modeling algorithm will give a higher topic score for that keyword. Some standard spam protection techniques may help, but there is a need for further research.

• **Privacy:** Having author-centric ranking implicitly assumes that people will need to link/claim their web content and connect their (real or pseudo) online identity to the content they contribute. This opens up challenges for strong guarantees on privacy and anonymity.

7. CONCLUSION & FUTURE WORK

To the best of our knowledge no other work has previously proposed to explicitly use authorship information of web content for search ranking. We believe that our abstraction of constructing "overlay" author nodes separate from the web graph and then performing score transfers on familiar graph semantics is simple enough to be easy to implement and yet powerful enough to allow complex score transfer methods. Web search today does consider generic *topic models* of content at times but does not factor in the topic-influence of the respective authors as proposed by us. We are currently focusing on the following next steps:

• **Naming System:** For a move towards author-centric ranking, we need a naming system where authors can easily and securely claim the online content that they've authored. This is a hard problem to solve. We're currently taking a first stab at this problem by building a new content based naming system for online web content. Given a valid URL the naming system hashes the content of the web document to give a self-certifying 128-bit flat name of the document. We plan to explore how to keep a mapping between authors and the pages they've authored, and how authors can claim their content using this naming service. We are planning to deploy this service on PlanetLab [7] and are designing custom web crawlers that can lookup authorship information.

• **Authority:** Calculating the "right" authority or topic influence of an author for any given topic is a non-trivial issue because there are no clear right or wrong answers. In this paper we used LDA for calculating topic models and the respective scores; various other variants can be used instead. There are some services like *Klout.com* that try to calculate topic authority of people. Currently, there is no clear way to say that one rank is more appropriate than the other and there is a need for better scientific analysis for comparison of topic authority calculations.

• **Rank Calculations:** Currently, we only increase topic ranks and never decrease them and use a particular function for such "boosting" of topic ranks. It's easy to imagine that many different variants of the rank transfer function can be used instead and it'll be interesting to compare the performance of such variants. Also, looking at explicit endorsements for topic-authority, e.g., endorsements on LinkedIn, in addition to topic-models can be interesting.

• **Scalability:** Our rank transfers are independent of the size of the original web graph. However, we still need to perform graph transformations to construct a new graph and most other calculations are also on the order of some subset nodes and edges of the given graphs. These operations can become costly for graphs containing several millions or billions of nodes and need further work.

• **Applications:** Apart from web search, other applications of author-centric ranking are also possible e.g., in online digital libraries, crowd sourcing, and in assigning quality scores to human endorsements (e.g., on LinkedIn).

8. REFERENCES

[1] Apache Lucene - Scoring. Lucene 3.6.2 Documentation, 2012.

[2] About.me. http://www.about.me, 2013.

[3] Amazon Mechnical Turk. Requester Best Practices Guide, 2012.

[4] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct Answers for Search Queries in the Long Tail. In *SIGCHI'12*, pages 237–246, 2012.

[5] D. Blei and J. Lafferty. Topic Models. In A. Srivastava and M. Sahami, editors, *Text Mining: Classification, Clustering, and Applications*. 2009.

[6] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan. 2003.

[7] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: An Overlay Testbed for Broad-Coverage Services. *ACM SIGCOMM Computer Communication Review*, 33(3), July 2003.

[8] A. Das and A. Jain. Indexing The World Wide Web: The Journey So Far. *Next Generation Search Engines: Advanced Models for Information Retrieval*.

[9] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the Essence: Improving Recency Ranking using Twitter Data. In *WWW'2010*, pages 331–340. ACM, 2010.

[10] R. Fishkin. Latent Dirichlet Allocation (LDA) and Google's Rankings are Remarkably Well Correlated, Sept. 2010.

[11] D. Horowitz and S. D. Kamvar. The Anatomy of a Large-scale Social Search Engine. In *WWW'2010*, 2010.

[12] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *JACM: Journal of the ACM*, 46, 1999.

[13] A. Mislove, K. P. Gummadi, and P. Druschel. Exploiting Social Networks for Internet Search. In *HotNets'06*, November 2006.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report SIDL-WP-1999-0120, Stanford University, Nov. 1999.

[15] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *LREC 2010 Workshop on NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010.

[16] Twitter API. http://dev.twitter.com/, 2013.