# Topic Similarity Networks: Visual Analytics for Large Document Sets

Arun S. Maiya
Institute for Defense Analyses
Alexandria, VA 22311
Email: amaiya@ida.org

Robert M. Rolfe
Institute for Defense Analyses
Alexandria, VA 22311
Email: rolfe@ida.org

***Abstract*—We investigate ways in which to improve the interpretability of LDA topic models by better analyzing and visualizing their outputs. We focus on examining what we refer to as *topic similarity networks*: graphs in which nodes represent latent topics in text collections and links represent similarity among topics. We describe efficient and effective approaches to both building and labeling such networks. Visualizations of topic models based on these networks are shown to be a powerful means of exploring, characterizing, and summarizing large collections of unstructured text documents. They help to "tease out" non-obvious connections among different sets of documents and provide insights into how topics form larger themes. We demonstrate the efficacy and practicality of these approaches through two case studies: 1) NSF grants for basic research spanning a 14 year period and 2) the entire English portion of Wikipedia.**

## I. Introduction and Motivation

In this paper, we study network visualizations as a means of enhancing the interpretability of probabilistic topic models for insight discovery. We focus on what is perhaps the most popular and prevalently-used topic model: *latent Dirichlet allocation* or LDA [5]. Topic modeling algorithms like LDA discover latent themes (*i.e.,* topics) in document collections and represent documents as a combination of these themes. Thus, they are critical tools for exploring text data across many domains. Indeed, it is often the case that users must *discover* the subject matter buried within large and unfamiliar document sets (*e.g.,* sensemaking in text data). Keyword searches are inadequate here, as it is unclear on where to even begin searching. Topic discovery techniques such as LDA are a boon to users in such scenarios, as they reveal the content in an unsupervised and automated fashion. Automated topic organization can potentially facilitate the comprehension of unfamiliar document data on even a massive scale.

However, it is often quite challenging to obtain a "big picture" view of the larger trends in a document collection from only the raw output of an LDA model. LDA is fundamentally a statistical tool that returns a probability distribution for each document showing the relative presence (or absence) of various discovered topics. These topics, in turn, are represented as probability distributions over words (typically unigrams). Words with the highest estimated probabilities for a discovered topic are used as a *label* for the topic. Exploring text corpora using only these raw outputs is considerably challenging. In order to derive insights and identify larger trends within the document collection, one is left to inspect these numerical distributions, which can be difficult, non-trivial, and far from straightforward. The problem is exacerbated as document collections under consideration grow. For instance, with the existence of scalable, MapReduce implementations of LDA (*e.g.,* [27], [29]), it is now possible to train an LDA model on massive text corpora with many latent topics (*i.e.,* big data). The inferred topics discovered by these LDA implementations, can themselves pose their own unique data challenge. It is often unclear on how best to effectively browse these topics to discover information of interest. This, in fact, tends to be a significant challenge even for large data (as opposed to "big data") — *e.g.,* document collections on the order of tens of thousands or hundreds of thousands. In the present work, we investigate the use of what we refer to as *topic similarity networks* to address these challenges. *Topic similarity networks* are graphs in which nodes represent latent topics in text collections and links represent similarity among topics. We describe efficient and effective methods to both building and labeling such networks.

**Summary of Contributions.** Our contributions in both the areas of topic visualization and topic labeling are summarized below.

1) *Constructing Topic Similarity Networks:* In Section IV, we describe the construction of *topic similarity networks*, our approach to big data visualization. We exploit these networks to discover how topics form larger themes. We employ the use of community detection in network visualizations to discover such macro-level themes including the sometimes subtle connections among these themes.
2) *Labeling Topic Similarity Networks:* In Section V, we describe an approach to expressively labeling discovered topics. Our method, based on keyphrase extraction, is purely unsupervised, extractive, and demonstrably efficient. These labels are, then, employed as node labels in our *topic similarity networks* to enable better characterization of large document sets. It is surprising to note that, to the best of our knowledge, few of the existing works on topic visualization (discussed in the next section) make use of automated topic labeling methods. Our work, then, represents one of the first examinations of the efficacy of automated topic labeling in actual topic visualizations of large, real-world data.

There has been a wave of recent work to address challenges in both visualizing topics and labeling topics – each of which we discuss separately in light of our work.

## II. Background and Related Work

### A. Visualizing Topics

A number of both graphical and text-based visualizations and user interfaces have been proposed in the existing literature to browse topics (*e.g.,* [7], [9], [10], [14], [15], [19], [28]). Several, like *TopicViz* by Eisenstein et al. [14] and *TopicNets* by Gretarsson et al. [15], are quite innovative and make significant strides towards improving the interpretability of learned topic models. However, most of these existing methods focus on shedding light on the relationships between topics and documents (or attributes of documents). Although some (*e.g.,* [15]) support the inference of pair-wise similarity between topics, they do not provide insights into how topics come together to form larger themes or the subtle connections among seemingly disparate groups of topics. Such insights are important in obtaining a "big picture" view of ill-understood document collections. An exception to this rule is work on *correlated topic models* (or CTMs) and its variants (*e.g.,* [2], [8], [17]). CTMs model and infer associations among topics. These associations can be further mined to produce clusters of topics that represent larger themes for incorporation into visualizations. These models, however, reveal certain challenges when applied in real-world scenarios. First, existing visualizations based on CTM and its variants do not appear to easily lend themselves to extracting the kinds of insights mentioned above. This is due both to the way in which the topic relations are constructed and depicted and also the way in which the topic nodes are labeled (topic labeling is discussed in the next section). One may refer to [3], [8], [17] for examples of these existing visualizations and for comparison to our visualizations shown later. Second, some approaches, such as [17], artificially constrict the topic relation structure with specification of what are referred to as supertopics, which can hinder a view of the subtle connections among different and seemingly disparate groups of topics and subtopics. A third issue is related to practical scalability. Chen et al. [8] showed that CTM is unable to process a corpus of 285K documents in any reasonable time frame (*i.e.,* it will not finish within a week). Similarly, an approach to infer topic hierarchies proposed by [25] is limited to short texts only. ScaCTM, a parallelized extension to CTM, was shown to be substantially more scalable given a cluster of 40 machines [8]. But, for certain domains, such machine clusters may not be available at sites of deployment. In fact, it is often the case that only a single multi-core machine is available to process millions of documents, as the storage capacity of today's machines often outstrips their processing capacity. Even in scenarios where one has access to a large machine cluster, LDA is *significantly* more scalable and efficient because it does not learn the correlation structure among topics. (See [8] for a time complexity analysis of CTM, ScaCTM, and LDA). Given these aforementioned issues and the clear scalability, efficiency, and also prevalence of LDA, our objective in this work is to infer these topic associations in an organic fashion from the raw output of the *original* LDA model. As we will describe in Section IV, we do so by constructing *topic similarity networks*: networks depicting the similarity (represented as links) among topics (represented as nodes). Next, we discuss existing work on the labeling of topics.

### B. Labeling Topics

A topic similarity network is only useful as a visualization tool if the identity of network nodes are easily discernible. Several visualization schemes label topics by simply using the most probable word (or words) from the topic model (*e.g.,* [8], [15], [17]). However, LDA-derived labels have been observed to not always be adequately expressive of the topic (*e.g.,* see [22], [25], [28]). As a result, a number of methods have been proposed to better label topics in an automated fashion (*e.g.,* [4], [16], [19], [22], [25], [26]). Unfortunately, for a variety of reasons, most of these existing techniques are unable to handle the large text corpora we consider in this work. In Section V, we describe our own method to label topics to address gaps in this existing literature on topic labeling. To better motivate the use of our own labeling method, we describe several goals that must be met by any labeling scheme for a topic similarity network in light of existing work on topic labeling.

**Unsupervised.** The labeling method must be unsupervised, as obtaining a training set for a supervised labeling method can be prohibitively expensive and time-consuming.

**Extractive.** The labeling method must be extractive. That is, labels must be generated directly from the terms within the corpus under consideration, as opposed to an external reference corpus such as Wikipedia. This is especially important for the government and corporate domains, which often deal with document collections describing sensitive or proprietary information, state-of-the-art "bleeding edge" technology, or otherwise esoteric subject matter. Such information may not reside in publicly available reference corpora like Wikipedia. This requirement prevents us from utilizing methods such as [16], which employs the use of reference corpora when labeling topics.

**Supportive of User Interactivity.** Topic similarity networks are intended for use with *interactive* systems utilizing full-text search and faceted navigation of documents (*e.g.,* Solr search engine[1]). Under these scenarios, the documents comprising topics may be filtered in various ways *after* creation of the topic model. For instance, in the government domain, only those documents containing certain markings might be deemed of interest and selected in a visualization. Labels heavily associated with documents that have been filtered out may no longer adequately describe the remaining documents pertaining to important sub-topics. Labeling methods that are tightly coupled with the topic model (*e.g.,* [4], [22], [25], [26]) cannot cope well with such dynamic scenarios. Moreover, it is prohibitively expensive to re-generate the topic model on the filtered document collection. For these reasons, our labeling method, described in Section V, is purposefully de-coupled from the output of LDA. Hence, it can re-label topics in a filtered document collection without having to re-generate the topic model. Our labeling method, then, can best be characterized as a cluster labeling approach to topic labeling.

**Efficient.** Dynamic filtering of document collections, as

[1] https://lucene.apache.org/solr

described above, also necessitates a need for efficiency in the labeling approach. As the document collection is filtered in various ways, the labeling method might be repeatedly executed on a large document collection, which can be problematic for some existing labeling methods. For instance, we were unsuccessful in executing the approach by [4] on the document sets of interest in this work. The approaches by [25] and [22] also do not appear to scale as easily or as well to larger collections of longer documents. The method from [25], for example, was designed only for very short texts (*e.g.,* titles only).

These aforementioned issues motivate our development of a custom labeling method for use with topic similarity networks — a method that can scale to even massive collections of documents. We begin a discussion of our work with a brief overview of LDA and the notation and symbols used throughout this paper.

## III. Preliminaries

Let $D = \{d_1, d_2, \ldots, d_N\}$ represent a document collection of interest and let $K$ be the number of topics or themes in $D$. Each document is composed of a sequence of words: $d_i = \langle w_{i1}, w_{i2}, \ldots, w_{iN_i} \rangle$, where $N_i$ is the number of words in $d_i$ and $i \in \{1 \ldots N\}$. Let $W = \bigcup_{i=1}^{N} f(d_i)$ be the vocabulary of $D$, where $f(\cdot)$ takes a sequence of elements and returns a set. Probabilistic topic models like LDA take $D$ and $K$ as input and produce two matrices as output. The matrix $\theta \in \mathbb{R}^{N \times K}$ is the document-topic distribution matrix and shows the distribution of topics within each document. The matrix $\beta \in \mathbb{R}^{K \times |W|}$ is the topic-word distribution matrix and shows the distribution of words in each topic. Each row of these matrices represents a probability distribution. For any topic $i \in \{1, \ldots, K\}$, the $L$ terms with the highest probability in distribution $\beta_i$ are typically used as thematic labels for the topic. We use these LDA-derived labels as a baseline for comparison in our work. But first, we describe construction of the topic similarity network.

## IV. Constructing the Network

LDA captures the degree to which both documents and words are topically related. However, relations among the topics themselves are *not* explicitly captured. As we will show shortly, such topic-level relations can be used to construct network representations of text corpora. These representations, in turn, can be used to better understand, characterize, and visualize the themes in a document collection. In the present work, we define these relations based on topic similarity.

**Measuring Topic Similarity.** Recall that topics are represented as probability distributions over vocabulary $W$ and captured by the matrix $\beta$. Thus, the similarity for any two topics can be directly computed by comparing the word distributions from $\beta$. The Kullback-Leibler (KL) divergence, a distance measure of two probability distributions, is often used to make such comparisons (*e.g.,* [15], [28]). However, KL divergence satisfies neither the triangle inequality nor symmetry and is, therefore, not a metric. As such, it is less appropriate for defining network links based on similarity (the complement of distance). Although symmetric versions of KL divergence exist, we instead employ the Hellinger distance metric to compute topic similarity. Specifically, for any two topics $x, y \in \{1 \ldots K\}$, the Hellinger similarity is measured as:

$$H_S(\beta_x, \beta_y) = 1 - \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{|W|} (\sqrt{\beta_{xi}} - \sqrt{\beta_{yi}})^2}. \qquad (1)$$

A topic similarity network $G = (V, E)$ can be constructed where $V = \{v_1 \ldots v_K\}$ is the set of nodes representing discovered topics and $E$ is the set of edges representing similarities among topics. For any two topics $x, y \in \{1 \ldots K\}$, an edge $\{v_x, v_y\} \in V$ exists if and only if $H_S(\beta_x, \beta_y)$ is greater than some pre-defined threshold, $\xi$.

**Measuring Topic Similarity in MapReduce.** Note that, when constructing a topic similarity network as just described, the number of computed similarities scales quadratically with $K$. However, since $K \ll |D|$, the method remains computationally viable even for very large document collections. Moreover, with some well-placed substitutions, $\beta$ can be represented using a sparse matrix format for efficient in-memory processing of massive document sets. (We currently employ a compressed sparse row format for storing and manipulating $\beta$.) Nevertheless, for scenarios when even sparse representations of $\beta$ are unwieldy and a high degree of parallelization is desired, we propose a MapReduce implementation of the topic similarity computation. When breaking down problems into distributable units of work under the MapReduce model for parallelization, key-value pairs are employed as the core data structure [12]. In our case, each cell in the matrix $\beta$ can be represented as a key-value pair of the form ($i$ : ($j$, $\beta_{ji}$)), where $i \in \{1 \ldots |W|\}$ is the index of a word (*i.e.,* column), $j \in \{1 \ldots K\}$ is the index of the topic (*i.e.,* row), and $\beta_{ji}$ is the probability of word $i$ appearing in topic $j$. If grouping by key, we obtain a key-value representation of each column in $\beta$. That is, the values list for any key $i \in \{1 \ldots |W|\}$ comprises the set of tuples $\{(j, \beta_{ji}) \mid j \in \{1 \ldots K\}\}$). The *map* operation accepts these key-value pairs as input and outputs key-value pairs of the form ($x, y$ : $e_i$), where the new key $x, y \in \{1 \ldots K\}$ are pairs of topics appearing in the aforementioned values list and the value $e_i = (\sqrt{\beta_{xi}} - \sqrt{\beta_{yi}})^2$, for each word $i \in \{1 \ldots |W|\}$. Thus, the *map* operation completes the inner expression for Hellinger similarity (shown in Equation 1) for every word represented in $\beta$. The *reduce* operation simply sums these values for every pair of topics and completes the Hellinger similarity computation by taking the square root of this sum, multiplying by $\frac{1}{\sqrt{2}}$, and subtracting from one. The resultant network, constructed as described above, can be exploited to discover insights, trends, and patterns among the topics in $D$. For the present work, we employ the use of a community detection algorithm to discover insights into how topics are related to each other and form larger themes.

**Discovering Larger Themes.** A *community* can be loosely defined as a set of nodes more densely connected among themselves than to other nodes in the network [6]. Within the context of a topic similarity network, such communities should represent groups of highly-related topics, which we refer to as *topic groups*. To detect these communities (or topic groups), we employ the use of the Louvain algorithm, a heuristic method based on modularity optimization [6]. Modularity measures the fraction of links falling within communities as compared to the expected fraction if links were distributed evenly in the network [23]. The algorithm initially assigns each topic node to its own community. At each iteration, in a local and greedy fashion, topic nodes are re-assigned to communities with which it achieves the highest modularity. As a greedy optimization method, the Louvain algorithm is exceptionally efficient and fast, even with a large number of topics. As the authors of [6] note, the computational complexity of the method is unknown, but it experimentally appears to run in $O(n \log n)$ time. When the nodes in these constructed topic similarity networks are marked by their inferred community affiliation and labeled to express the topics they represent, the networks become powerful tools for exploration and discovery in large and heterogeneous text corpora. We discuss labeling of topic nodes next.

## V. Labeling the Network

An algorithm capable of generating expressive thematic labels for any subset of documents in a corpus can greatly facilitate both characterization and navigation of document collections. Here, we employ such an algorithm to label nodes in a topic similarity network, as each node is a topic comprising a subset of documents in the corpus. Our approach, referred to as DocSetLabeler, is a purely unsupervised, extractive method and shown in Algorithm 1.[2] DocSetLabeler takes $D_S$, a subset of corpus $D$, as input, where $D_S$ consists of all documents associated with some LDA-discovered topic $t \in \{1 \dots K\}$. This subset can be constructed in one of two ways. The first is to populate $D_S$ with all documents $d_i$ (where $i \in \{1 \dots N\}$) for which the topic proportion $\theta_{it}$ is greater than some pre-defined threshold (*e.g.,* 0.3 was used in [28]). The second is to construct $D_S$ by transforming topics into mutually-exclusive clusters, where the topic cluster for document $d_i$ is $\text{argmax}_x\, \theta_{ix}$. We employ the latter approach, as it better eliminates noise contributed by foreign topics (*i.e.,* $\{1 \dots K\} - \{t\}$). Labels for topic $t$ are, then, extracted by DocSetLabeler directly from the text constituting the documents in $D_S$.

DocSetLabeler is essentially a *descriptive* model of topic labeling that follows naturally from four observed characteristics of high-quality, topic-representative labels: *Expressivity*, *Prominence*, *Prevalence*, and *Discriminability*.

**Algorithm 1** DocSetLabeler algorithm

```
Require: $D_S \subset D$, a subset of corpus $D$
Require: $C$, the number of candidate terms to consider
Require: $L$, the number of labels to return for document set ($L \leq C$)
Require: stopwords, list of terms to filter out
 1: pos = a hash table
 2: neg = a hash table
 3: for all $d \in D$ do
 4:     $terms1 = \text{extractSignificantPhrases}(d, \text{stopwords})$
 5:     $terms2 = \text{extractNounPhrases}(d, \text{stopwords})$
 6:     $terms3 = \text{extractProperNounUnigrams}(d, \text{stopwords})$
 7:     $candidates = (terms1 \cap terms2) \cup terms3$
 8:     for all $c \in candidates$ do
 9:         $x =$ normalized frequency of term c in $d$
10:         $y = 1 - \frac{\text{index of first occurrence of } c \text{ in } d}{\text{num. of words in d}}$
11:         (weight of term $c$) $= \frac{2 \cdot x \cdot y}{x+y}$
12:     end for
13:     if $d \in D_S$ then
14:         pos[d] = top $C$ terms based on weight
15:     else
16:         neg[d] = top $C$ terms based on weight
17:     end if
18: end for
19: for all $\ell \in \bigcup_{x \in \text{pos.values()}} x$ do
20:     # compute information gain for each label $\ell$
21:     (score of label $\ell$) $= \text{calcScore}(\ell, \text{pos}, \text{neg})$
22: end for
23: $top\_candidates =$ top $C$ labels based on information gain
24: # optionally re-sort final top candidates
25: $top\_candidates = \text{re\_sort}(top\_candidates)$
26: return top $L$ labels from $top\_candidates$
```

**Expressivity.** *Expressivity* captures the extent to which labels express and represent themes. Previous works have noted that human-assigned labels tend towards multi-word noun phrases, as they are more expressive than unigrams (*e.g.,* see [24]). The term "information retrieval," for instance, is more expressive than just "information" or "retrieval" alone. Unigrams tend to most often be expressive when denoting uniqueness (*i.e.,* a proper noun). This is especially true of research reports, our domain of interest, as proper noun unigrams denote important concepts, systems, techniques, or programs (*e.g.,* "LinearSVM," "F-22"). Lines 4-6 in Algorithm 1 explicitly extract terms conforming to the above principles. Noun phrases[3] and proper nouns are extracted using *hunpos*, an open-source, HMM-based, part-of-speech tagger.[4] The $\text{extractSignificantPhrases}(\cdot)$ function uses likelihood ratio tests to extract phrases of multiple words that occur together more often than chance.[5] For a bigram of words $w_1$ and $w_2$, this association, $assoc(\cdot, \cdot)$, is measured as:

$$assoc(w_1, w_2) = 2 \sum_{ij} n_{ij} \log \frac{n_{ij}}{m_{ij}}, \tag{2}$$

where $n_{ij}$ are the observed frequencies of the bigram from the contingency table for $w_1$ and $w_2$ and $m_{ij}$ are the expected frequencies assuming that the bigram is independent [13]. Only phrases with a p-value less than $0.001$ are extracted. These tests can also be used to measure associations of words within n-grams where $n \geq 3$ (*e.g.,* trigrams). However, we limit phrases to the $n < 3$ cases to save space in the visualizations.

**Prominence.** *Prominence* captures the degree to which labels are featured prominently within individual documents. Intuitively, prominent terms tend to make their first appearance earlier and also appear more frequently. Thus, we weight candidate labels by both frequency and position using the harmonic mean, as shown in Line 11 of Algorithm 1.

[2] Lines 4–11 of Algorithm 1 are a variation of the KERA algorithm described in [19].

[3] We use the POS pattern: (ADJECTIVE)*(NOUN)+.

[4] http://code.google.com/p/hunpos/

[5] This is known as *collocation extraction* [20].

| **Actual Topic** | **Labels from LDA** | **Labels from** DocSetLabeler |
|---|---|---|
| **Fluid Mechanics and Fluid Dynamics** | flow,fluid,flows,fluids,dynamics,transports | fluid dynamics, fluid mechanics, multiphase flow |
| **Game Theory** | agents,theory,game,agent,games,equilibrium | game theory, economic agents, repeated games |
| **Graph Theory** | discrete,graph,combinatorial,theory,combinatorics,graphs | graph theory, algebraic combinatorics, ramsey theory |
| **Human Evolution** | modern,fossil,early,years,human,age | modern humans, human evolution, hominid evolution |
| **Hydrology** | water,river,hydrologic,watershed,balance,surface | hydrologic controls, watershed scale, alpine basins |
| **Modal Analysis in Structural Engineering** | mode,modes,research,vibration,direction,coupling | normal modes, vibration control, modal analysis |
| **Object Recognition** | object, objects,features,recognition, oriented,feature | object recognition, curved objects, cluttered scenes |
| **Protein Function/Mechanisms** | protein,proteins,function,role,biochemical,phosphorylation | protein kinases, protein phosphorylation, protein import |
| **Protein Structure** | protein,proteins,binding,structure,amino,acid | protein structure, protein folding, amino acid |
| **Social Psychology** | social,people,research,individuals,attitudes,status | social psychology, social influence, social perception |

TABLE I: **[NSF Grants.]** Ten discovered NSF topics and the highest-ranked labels assigned to each by both LDA and DocSetLabeler.

**Prevalence and Discriminability.** Good labels for a particular topic appear in many documents pertaining to that topic (*Prevalence*) and appear rarely in other un-related topics (*Discriminability*). This was also recently observed by [11] and [28]. The concept of *information gain* from the field of information theory simultaneously captures both prevalence and discriminability. Consider a document collection $D$ where documents belong to either a positive or negative category. The *entropy* $\mathrm{H}$ of $D$ measures impurity as follows: $\mathrm{H}(D) = -p^{+} \log_2(p^{+}) - p^{-} \log_2(p^{-})$, where $p^{+}$ and $p^{-}$ are the proportions of positive and negative documents in $D$, respectively.[6] For instance, if all documents are positive (or negative), $\mathrm{H}(D) = 0$, while a perfectly even split of positive and negative documents has entropy of 1. In Algorithm 1, we assign $D_S$ as positive and $\overline{D_S}$ as negative. The information gain IG of a candidate label $\ell$ in $D$, then, is the expected entropy reduction due to segmenting on $\ell$: $\mathrm{IG}(\ell, D) = \mathrm{H}(D) - (\frac{|D^{\ell}|}{|D|}\mathrm{H}(D^{\ell}) + \frac{|\overline{D^{\ell}}|}{|D|}\mathrm{H}(\overline{D^{\ell}}))$, where $D^{\ell}$ is the set of documents in $D$ from which label $\ell$ was extracted. Thus, labels with the highest information gain for $D_S$ are expected to be simultaneously common in $D_S$ (prevalence) and rare in $\overline{D_S}$ (discriminability). Information gain is computed by the $\mathrm{calcScore}(\cdot)$ function in Algorithm 1.

**Final Sorting.** At the end of the previous step, we are left with a small number of candidate labels (*e.g.,* $C = 5$) for each topic. There are several options for choosing the final label for the topic node. For instance, one could simply select the label with the highest information gain (*i.e.,* the existing sorting). One might also select the label most frequently extracted from the documents pertaining to the topic. Yet another option is to include word probabilities from $\beta$ into the final weighting. All three approaches generally yield good (albeit slightly different) results. For the present work, based on some preliminary testing, we choose to sort labels based on a combination of the latter two approaches, as indicated in Line 25 of Algorithm 1. More specifically, we sort labels based on the mean of the normalized frequency and the combined $\beta$ probabilities for each word comprising the label.

To conclude, we briefly comment on the efficiency and scalability of our current DocSetLabeler implementation. Note that, in Algorithm 1, Lines 1–12 process documents in an online fashion and can be easily parallelized. Computing information gain also scales well to larger collections of longer documents, as it is a simple computation of different combinations of independent and dependent variables. Moreover, it deals with a substantially reduced representation of the data (*i.e.,* generally, $C \ll N_i$ for all $i \in \{1 \ldots N\}$). For these reasons, it is fairly straightforward to implement DocSetLabeler in a variety of different parallel processing models (*e.g.,* MapReduce, multi-core processing). Lines 1–12, for instance, can be implemented as a map-only job with either zero reducers or an identity reducer. On the other hand, for execution on single-node, multi-core, shared-memory systems (as opposed to clusters), documents can be processed in an online fashion and passed to as many processors available on the system.

## VI. Case Study 1: NSF Research Grants

As a realistic and informative case study, we utilize our methods to characterize and visualize basic research funded by the National Science Foundation (NSF). The corpus considered in this case study consists of 132,372 titles and abstracts describing NSF awards for basic research between the years 1990 and 2003 [1]. We executed the MALLET implementation of LDA [21] on this corpus using $K = 400$ as the number of topics and 200 as the number of iterations. All other parameters were left as defaults. For topic similarity, we experimentally set $\xi$ as 0.15 to yield a graph density of approximately 0.01. For the labeling of topic nodes in the network using DocSetLabeler, we set $C = 5$ and $L = 1$. We did not find the choice of $C$ to affect results significantly. This is possibly due to the fact that, as described previously, we prune out candidates with no statistical significance, as measured by a likelihood ratio test.

**Topic Labeling of NSF Grants.** Table I shows the labels generated for a sample of ten discovered topics by both DocSetLabeler and LDA. As can be seen, labels produced by DocSetLabeler are more expressive and representative of the true themes of each topic. We assigned two judges to evaluate labels for all topics. For a fair comparison, we showed six unigram labels from LDA but only three labels (mostly bigrams) from DocSetLabeler for each topic. As shown in Table III, both judged the labels by DocSetLabeler to be generally superior ($\chi^2$=145.73, $P$<0.0001) with an inter-judge agreement of 0.62, as measured by Cohen's kappa coefficient.

|  | DocSetLabeler | LDA |
|---|---|---|
| DocSetLabeler | 313 | 6 |
| LDA | 23 | 29 |

TABLE III: Evaluation of labels for each method on NSF grants. Overall, both judges chose labels from DocSetLabeler to be most on-point. (Poor quality topics thrown out.)

[6] Note that $\log_2(0)$ is taken to be 0.

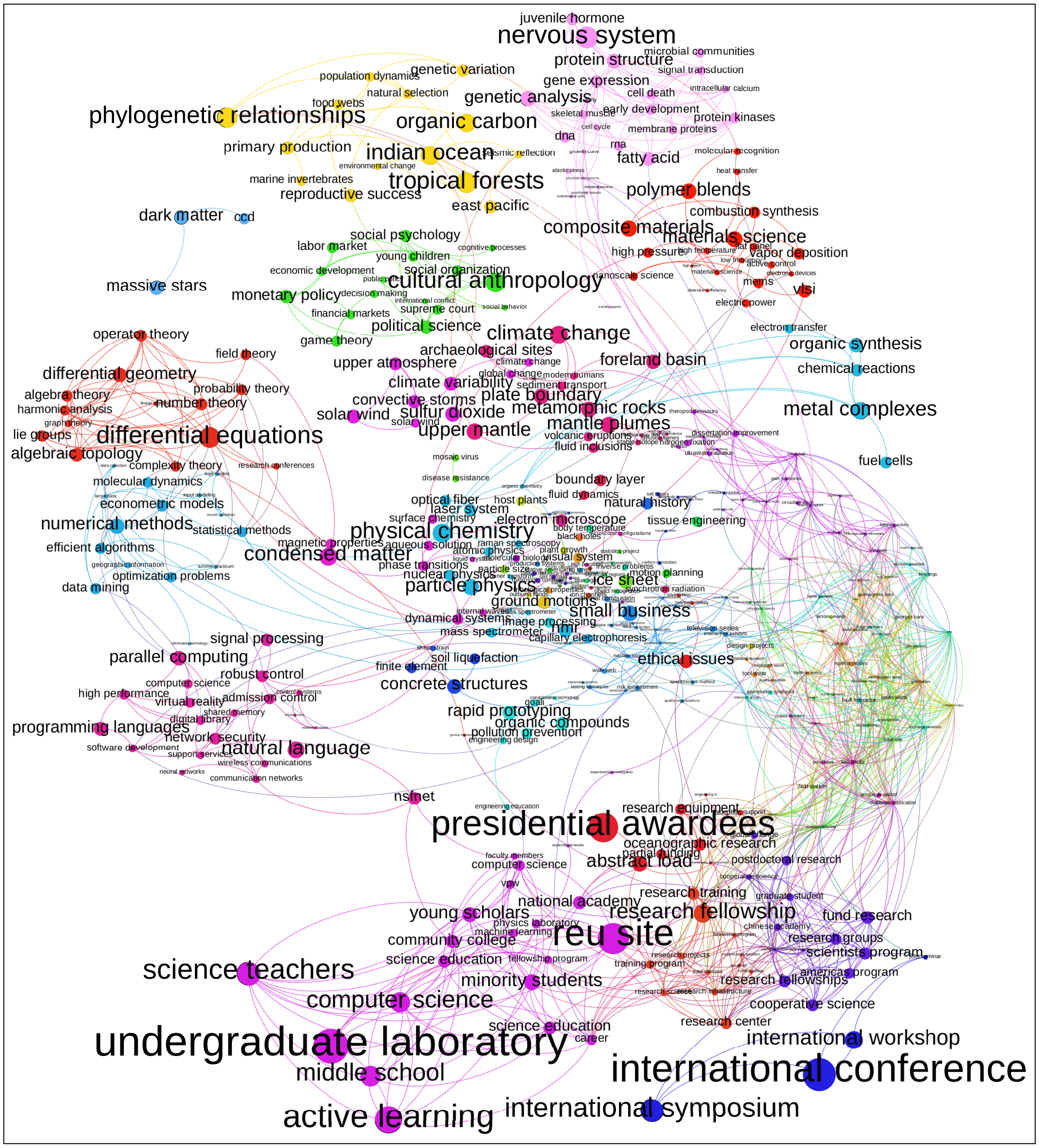


Fig. 1: **[NSF Grants.]** The *Topic Similarity Network* of 14 years of NSF research and support (*i.e.,* a total of 132, 372 research grants). Major research topics are shown including their subtle connections to each other. Also displayed (towards the bottom of network) are major funding efforts for education support and conference support. Node sizes indicate the number of grant abstracts pertaining to the topic. Node colors indicate the community (or topic group) affiliation, which illustrate how research topics form larger themes.

**Visualizing NSF Grants.** A topic similarity network was constructed, with each node representing a topic and labeled using the highest ranked term returned by DocSetLabeler. The network concisely presents a comprehensive and holistic view of 14 years of NSF-funded research and can be navigated and explored using any available network visualization software (*e.g.,* Gephi, Cytoscape ). The entire network is shown in Figure 1, where both expected and unexpected trends are revealed. As can be seen, the visualization encapsulates the major research funding efforts for scientific research in addition to the subtle connections among them. Major funding efforts for education and conference support are also displayed (towards the bottom). In this network and all networks shown in this paper, node sizes indicate the number of documents pertaining to the topic represented by the node.[7] Node colors indicate the community (or topic group)

[7] Although we could have sized nodes based on funding amount of the grant, we instead size nodes based on the number of documents for the sake of consistency.

| Actual Topic | Labels from LDA | Labels from DocSetLabeler |
|---|---|---|
| **BBC** | bbc,british,series,television,london,uk | bbc, british television, bbc radio, british actor |
| **Boxing** | fight,title,boxing,champion,round,boxer | professional boxer, professional career, amateur boxer |
| **Computers** | system,computer,systems,control,computers,electronic | computer science, operating system, control system |
| **Electronic Dance Music** | music,dj,label,dance,artists,records | electronic music, record label, dance music |
| **Probability Theory** | data,analysis,method,methods,distribution | probability distribution, random variables, random variable |
| **Manufacturing** | company,production,factory,manufacturing,plant,industry | manufacturing company, motor company, manufacturing plant |
| **Motorcycles** | motorcycle,racing,cc,race,davidson,bike | speedway rider, cc race, british motorcycle |
| **Summer Olympics** | olympics,summer,medal,won,olympic,world | summer olympics, gold medal, bronze medal |
| **Tropics** | species,family,tropical,habitat,natural,subtropical | tropical moist, habitat loss, natural habitats |
| **Winter Olympics** | winter,world,event,olympics,won,competed | winter olympics, world championships, ski championship |

TABLE II: **[Wikipedia.]** Ten discovered Wikipedia topics and the highest-ranked labels assigned to each by both LDA and DocSetLabeler.

affiliation. Using this network, one can better understand how topics form larger themes, discover and characterize information of interest, and derive insights into how best to search and explore the corpus further. It is difficult to quantitatively evaluate visualization schemes such as this. Thus, we present illustrative examples of the patterns and trends discovered using our topic similarity network. Figure 2 shows one small corner of the "topic universe" — a "social clique" of math topics discovered by community detection within the larger network of all topics. Note that each node in the network represents hundreds of documents (or more). Thus, this visualization of math topics clearly and concisely summarizes over $10,000$ documents. Such visualizations also provide insights into relations between topic groups. For instance, Figure 3 shows a community of biology-related topics (shown in pink). Here, we see peripheral connections to another life science theme (shown in yellow) containing topics such as *genetic variation*, *population dynamics*, and *food webs*. We also see a peripheral connection to a material science theme (shown in red), illuminating research areas dedicated to developing materials based on biological and organic components and also the mutual interest in molecular recognition. As a final example, Figure 4 shows a connected component of astronomical research topics that appears separate from the larger network. This last example illustrates one possible way to use these visualizations to identify outliers (*i.e.,* topics that are comparatively more different than the larger corpus based on their set of similarity scores).[8]

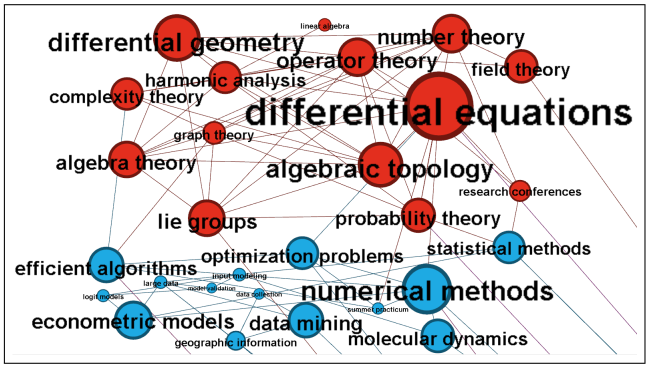


Fig. 2: **[NSF Grants.]** Two discovered topic groups (or communities) pertaining to math-oriented research. The red covers pure math, while the blue is more applied. Each are separate communities but tightly-coupled, as shown. Together, they represent over $10,000$ documents covering a range of math subfields.

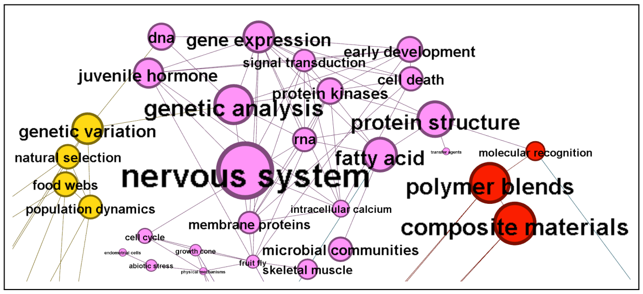


Fig. 3: **[NSF Grants.]** A discovered topic group related to biology (shown in pink). Also shown are topic nodes from other related communities (*e.g., polymer blends*, *population dynamics*) and their peripheral connections to this biology-related topic group.

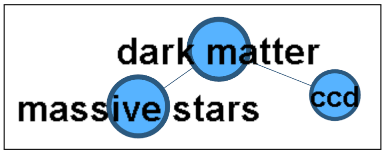


Fig. 4: **[NSF Grants.]** A connected component of astronomical research topics separated from the larger network.

## VII. Case Study 2: Wikipedia

For our second case study, we apply our method to visualize Wikipedia topics. The corpus considered here was obtained from the University of Alberta and comprises the entire English portion of Wikipedia.[9] It contains over 3.3 million documents spanning a range of different topics. We executed the MALLET implementation of LDA [21] on this corpus using $K = 1000$ as the number of topics and 200 as the number of iterations. All other parameters were left as defaults. For topic similarity, we experimentally set $\xi$ as 0.2 to yield a graph density of approximately 0.01. For the labeling of topic nodes in the network using DocSetLabeler, we again set $C = 5$ and $L = 1$.

**Labeling Wikipedia Topics.** Table II shows a sample of ten Wikipedia topics and the labels generated for each by both LDA and DocSetLabeler. As we did with the NSF grants, we conducted a user evaluation of the labels generated for all Wikipedia topics by both LDA and our method. From the results shown in Table IV, we again see that DocSetLabeler outperforms LDA ($\chi^2$=426.68, $P$<0.0001) with an inter-judge

[8] While it is possible to re-connect singleton nodes to whichever node it is most similar, we have not done so in any of the presented visualizations.

[9] Shaoul, C. & Westbury C. (2010) The Westbury Lab Wikipedia Corpus, Edmonton, AB: University of Alberta (downloaded from http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html)

agreement of 0.71, as measured by Cohen's kappa coefficient.. However, we also see that LDA performs significantly better here than on the NSF grants. We elaborate on this observation further in Section VIII.

| | DocSetLabeler | LDA |
|---|---|---|
| DocSetLabeler | 545 | 74 |
| LDA | 30 | 199 |

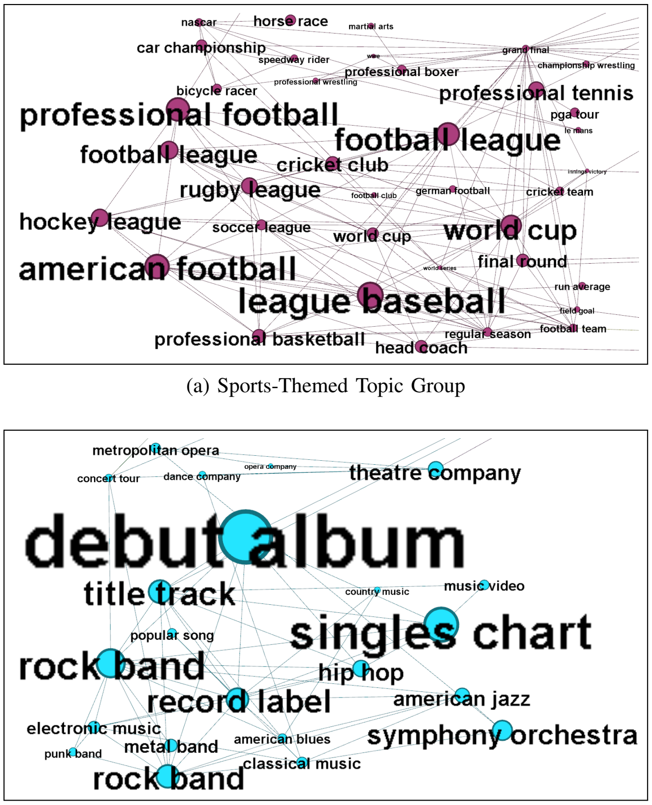
TABLE IV: **[Wikipedia.]** Evaluation of labels for each method on Wikipedia. Overall, both judges chose labels from DocSetLabeler to be most on-point. (Poor quality topics thrown out.)

**Visualizing Wikipedia.** A topic similarity network was constructed for Wikipedia, with nodes labeled using the highest ranked label generated from DocSetLabeler for each topic. Due to space constraints, we do not present the entire Wikipedia topic similarity network in this paper. Rather, we provide illustrative examples of some of the major trends discovered by our method. Two of the most salient and well-defined topic groups (*i.e.,* macro-level themes) emerging from our visualization are *sports* and *music/dance*, shown in Figures 5a and 5b, respectively. We posit that this is due to the fact that authorship and editing of Wikipedia articles are crowd-sourced and the subjects of *sports* and *music/dance* both have enormous fan bases. It should follow that television and film should also appear as salient topic groups, and this is precisely what we see in Figure 6. Also shown in Figure 6 are the peripheral connections to topic nodes from other related communities (*e.g., plot summary* and *love story* from a writing theme in green, *daily newspaper* and *monthly magazine* from a news media theme in yellow).

## VIII. Limitations

In both our two case studies, DocSetLabeler was observed to outperform LDA on topic labeling tasks. However, comparing the two case studies, we see the performance differential was less for Wikipedia topics and greater for the highly technical and scientific topics present in the NSF grants corpus. We attribute this to the fact that Wikipedia is an encyclopedia with many topics that are very general and broad in nature. On those topics that are so broad and general that they are best summarized with a single word (*e.g., songs*, *tennis*, *BBC*), LDA performs quite well – albeit sometimes less well than DocSetLabeler. In cases where there is not an equivalently expressive bigram (*i.e.,* two-word phrase) or proper unigram, LDA will perform better than our method, since DocSetLabeler currently focuses only on bigrams and *proper* unigrams. One example of the latter case is the *motorcycle* topic in Wikipedia shown in Table II. The top-ranked labels generated by DocSetLabeler are simply not as expressive as the simple label "motorcycle" produced by LDA. Addressing such cases is an area for future work. However, we find these cases to be in the minority – especially with respect to mining content from scientific and technical documents, which is our current and primary area of interest.

A second limitation is related to short texts. Both LDA and DocSetLabeler are optimized for articles, summaries, and reports, such as the corpora considered in this work. Shorter documents such as abstracts are also handled well by both algorithms, as evidenced by performance on the NSF grant abstracts. However, extremely short texts can cause difficulties.



(a) Sports-Themed Topic Group

(b) Music/Dance-Themed Topic Group

Fig. 5: **[Wikipedia.]** Discovered Wikipedia topic groups for: (a) *Sports* and (b) *Music/Dance*.

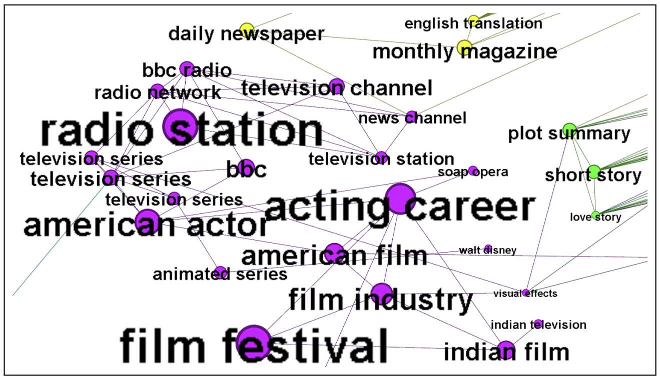


Fig. 6: **[Wikipedia.]** A discovered topic group pertaining to *Television/Film/Radio* (shown in purple). Also shown are the peripheral connections to topic nodes from other related communities (*e.g., plot summary* and *love story* from a writing theme in green, *daily newspaper* and *monthly magazine* from a news media theme in yellow).

This was observed to a certain degree in some Wikipedia topics containing many so-called "stub" articles of only a single sentence[10] (*e.g.,* one-sentence descriptions of minor fictional characters, small towns, or persons of minor notability). One solution might be to replace LDA and DocSetLabeler with algorithms specifically designed to handle short texts such as Twitter-LDA [30] and keyword extraction algorithms designed

[10] It appears that Wikipedia now recommends a minimum of three sentences for an article. See http://en.wikipedia.org/wiki/Wikipedia_talk:One_sentence_does_not_an_article_make

for short snippets of text [18]. We leave an investigation of this for future work.

## IX. Conclusion

We have investigated the use of *topic similarity networks* as a practical approach to improving the interpretability of LDA topic models. We described both how to construct such networks and an approach to labeling nodes in the network. These methods were combined and employed to effectively characterize and explore 14 years of NSF-funded basic research and the English portion of Wikipedia using network analysis. For future work, we plan on incorporating these visualizations into a larger, facet-based, text analytic system previously developed for the U.S. Department of Defense (see [19] for more details on this system).

## References

[1] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[2] David M. Blei and John D. Lafferty. Correlated Topic Models. In *NIPS*, 2005.

[3] David M. Blei and John D. Lafferty. A correlated topic model of Science. *Annals of Applied Statistics*, 1(1):17–35, August 2007.

[4] David M. Blei and John D. Lafferty. Visualizing Topics with Multi-Word Expressions, July 2009.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3(4-5):993–1022, March 2003.

[6] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, July 2008.

[7] Allison Chaney and David M. Blei. Visualizing Topic Models. In *ICWSM '12*, 2012.

[8] Jianfei Chen, June Zhu, Zi Wang, Xun Zheng, and Bo Zhang. Scalable Inference for Logistic-Normal Topic Models. In *NIPS 2013: Neural Information Processing Systems Conference*, December 2013.

[9] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. Interpretation and Trust: Designing Model-driven Visualizations for Text Analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 443–452, New York, NY, USA, 2012. ACM.

[10] P. J. Crossno, A. T. Wilson, T. M. Shead, and D. M. Dunlavy. TopicView: Visually Comparing Topic Models of Text Collections. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 936–943. IEEE, November 2011.

[11] Marina Danilevsky, Chi Wang, Nihit Desai, Jingyi Guo, and Jiawei Han. KERT: Automatic Extraction and Ranking of Topical Keyphrases from Content-Representative Document Titles, June 2013.

[12] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[13] Ted Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Comput. Linguist.*, 19(1):61–74, March 1993.

[14] Jacob Eisenstein, Duen H. Chau, Aniket Kittur, and Eric Xing. TopicViz: Interactive Topic Exploration in Document Collections. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2177–2182, New York, NY, USA, 2012. ACM.

[15] Brynjar Gretarsson, John O'donovan, Svetlin Bostandjiev, Tobias H. Llerer, Arthur Asuncion, David Newman, and Padhraic Smyth. TopicNets: Visual Analysis of Large Text Corpora with Topic Modeling. *Journal ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2), February 2012.

[16] Jey H. Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic Labelling of Topic Models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1536–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[17] Wei Li and Andrew McCallum. Pachinko Allocation: DAG-structured Mixture Models of Topic Correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 577–584, New York, NY, USA, 2006. ACM.

[18] Zhenhui Li, Ding Zhou, Yun F. Juan, and Jiawei Han. Keyword Extraction for Social Snippets. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1143–1144, New York, NY, USA, 2010. ACM.

[19] Arun S. Maiya, John P. Thompson, Francisco L. Lemos, and Robert M. Rolfe. Exploratory Analysis of Highly Heterogeneous Document Collections. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1375–1383, New York, NY, USA, 2013. ACM.

[20] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[21] Andrew K. McCallum. MALLET: A Machine Learning for Language Toolkit, 2002.

[22] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic Labeling of Multinomial Topic Models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 490–499, New York, NY, USA, 2007. ACM.

[23] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June 2006.

[24] PeterD Turney. Learning Algorithms for Keyphrase Extraction. 2(4):303–336, 2000.

[25] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. A Phrase Mining Framework for Recursive Construction of a Topical Hierarchy. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 437–445, New York, NY, USA, 2013. ACM.

[26] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 697–702, Washington, DC, USA, 2007. IEEE Computer Society.

[27] Yi Wang, Hongjie Bai, Matt Stanton, Wen Y. Chen, and Edward Y. Chang. PLDA: Parallel Latent Dirichlet Allocation for Large-Scale Applications. In *Proceedings of the 5th International Conference on Algorithmic Aspects in Information and Management*, volume 5564 of *AAIM '09*, pages 301–314, Berlin, Heidelberg, 2009. Springer-Verlag.

[28] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X. Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. TIARA: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 153–162, New York, NY, USA, 2010. ACM.

[29] K. Zhai, J. Boyd-Graber, and N. Asadi. Using Variational Inference and MapReduce to Scale Topic Modeling. *ArXiv e-prints: arXiv:1107.3765 [cs.AI]*, July 2011.

[30] WayneXin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and Traditional Media Using Topic Models. In Paul Clough, Colum Foley, Cathal Gurrin, GarethJ Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, chapter 34, pages 338–349. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.