Grisham: Topic-Based Search and Exploration

Authors

Abstract

We describe a demonstration of our system *Grisham*, presenting methods for topic-based paper exploration. We use a popular topic modeling algorithm, Latent Dirichlet Allocation, to derive topic distributions for each scientific paper in a DBLP citation data set. We allow users to specify personal topic distribution to contextualize the exploration experience. We demonstrate three types of exploration *keyword-based search*, *topic-based exploration*, and *citation-lineage search*. In each model we shape the results to be specific to a user specification. We describe the components of our web-based system.

Introduction

In a variety of situations, from literature surveys to legal document collections, people try to organize and explore large amounts of documents. Current technology to search on documents are done based on keywords with minor extensions. Sometimes in order to enable keyword search, external effort—What are they? – CPG— has to be put in to tag the documents with relevant keywords. Keyword-based search is useful when the user knows exactly what he or she is looking for. It is not particularly useful when a user wants to explore or learn a new topic. This is especially important when for example, a researcher wants to find out the state of the art in a particular area or a student would like to create a literature survey. In such situations, *topic-based search* can return more relevant results. Topic-based search is a classification of the search space to highlight topical relevance. In order to accomplish this, the topics underlying the document collection need to be extracted, and then they have to be represented in terms of those topics and ranked based on relevance to a particular topic.

In our system called *Grisham* we present various techniques for topic-based exploration and search of peer reviewed scientific papers. Our work allows user to search for scientific papers using three methods: First, users may perform a traditional search *keyword-based search* for papers. Second, users can specify a topic or topics he or she is interested in and the most relevant papers for that topic will be listed in the order of their relevance by our system. Lastly, users are allowed to explore similar or related documents using a visual graph like interface, i.e., given a paper or an article, a user will be shown a set of papers which cite the original paper in the order of their relevance to the topic. In addition, we allow the user to specify a topic distribution that specifies their interest. In all three methods, we adapt search results to personalize results.

The main contributions of our work are the *user-topic ranking function* which ranks the relevance of documents to a set of topics—I think we need to edit this; a similar ranking function is there in George et al. 2012 – CPG, the *similar document explorations* which is performed by computing the similarity of papers to a topic of interest, and *topic-document visualization* which ranks citations for a paper by the interest of the user.

Related Work

There have been previous systems to use topic modeling as a basis of search and exploration. We are the first to our knowledge to incorporate a user modeling into the loop.

A system of note is Yang et al. (Yang, Torget, and Mihalcea 2011) whom apply topic modeling to collections of historical news papers to assist search. They found that the topics generated from topic models are are generally good, however once the sets of topics are generated, an expert opinion is required to name them. In *Grisham*, we allow users to select numbered topics for article-search based on topically relevant words.

Topic Models

Topic models are a set of models for the documents in a collection or corpus. They enable us to represent the properties of a large corpus containing numerous words with a small set of *topics*, by extracting the underlying topical structure of the corpus and representing the documents according to these topics. We can then use these representations for organizing, summarizing, and searching the corpus. Traditionally, topic models assume word occurrences within a document are independent of each other—“bag of words” models. Latent Dirichlet Allocation or LDA (Blei, Ng, and Jordan 2003) is a well known, generative, probabilistic topic model for a corpus. A probabilistic generative model assumes data as *observations* that originate from a generative probabilistic process that includes *hidden* variables. The hidden variable are typically inferred via *posterior inference*, in which one tries to identify the posterior distribution of the hidden variables conditional on the observations. Loosely speaking, one can consider posterior inference as the reverse of a generative process. The generative model of LDA assumes that there exists a set of *latent* (hidden) topics in the corpus. A topic is defined as a distribution—they are assumed to be generated from a *Dirichlet* distribution with a set of parameters—over the corpus vocabulary. For example, the *whales* topic typically will have words related to *whales* and correlated topics, e.g., *blue whales*, *killer whales*, *whaling*, etc., with high probability and words related to other uncorrelated topics, e.g., *sports*, *medicine*, etc., with low probability—assuming the corpus is built from a subset of articles from the topics *whales*, *sports*, and *medicine*. In addition, each document in a corpus is described by a latent topic distribution—another *Dirichlet* distribution with a set of parameters—and the words in a document are generated from the document specific topic distribution.

In real life, we only observe documents and their words. As in any generative probabilistic model, the latent variables in the LDA model are typically identified by posterior inference. However, in most of these generative models, posterior inference is intractable due to the high dimensionality of the latent variable space, and practitioners typically rely on approximate posterior inference alternatives. For the LDA model, people have used different approximate inference methods such as deterministic *optimization methods* (Blei, Ng, and Jordan 2003) and *sampling methods* (Griffiths and Steyvers 2004) for the inference. Blei, Ng, and Jordan employed variational methods to find approximations to the posterior distribution of latent variables, by considering a family of lower bounds on the log likelihood, indexed by a set of variational parameters. The variational parameters are then identified by a deterministic optimization procedure that seeks to find an optimal lower bound. Griffiths and Steyvers's method was based on Gibbs sampling—a Markov chain Monte Carlo method that helps to approximate the intractable posterior integral as an empirical estimate of samples generated from a Markov chain. In Gibbs sampling, one forms the Markov chain by repeatedly sampling each variable conditional on the most recently sampled values of the other variables (Geman and Geman 1984). In this paper, we use the scalable implementation of the online variational inference algorithm for LDA (Hoffman, Blei, and Bach 2010) by (Řehůřek and Sojka 2010), for inference from the LDA model for a corpus.

Due to the fully generative semantics, even at the level of documents, LDA is expected to overcome several drawbacks—e.g., issues such as synonymy and polysemy of words—of earlier models for corpora such as Term-Frequency Inverse Document Frequency (TF-IDF, Salton, Wong, and Yang 1975) and Latent Semantic Analysis (LSA, Dumais et al. 1995). In this paper, we are interested in the LDA model parameters such as the corpus-level latent topic distributions and document-level latent topic distributions. The latent document topic distributions are lower dimensional representations of documents—compared to document term frequency vectors, in which each element of the vector stands for a term in the corpus vocabulary—and very useful for finding and grouping similar documents in a corpus. Now onwards, we denote θ_d^* for the estimate of document d's latent distribution on the topics, identified via the variational inference algorithm (Hoffman, Blei, and Bach 2010). Similarly, the latent topics in a corpus, which are distributions over the vocabulary terms, are helpful in visualizing the prevalent thematic structure of a corpus and exploring documents related to a specific theme of interest. For $j = 1, 2, \ldots, K$, we denote β_j^* for the estimate of the latent topic distributions of a corpus, where K is the number of topics in the corpus—it's a constant in the LDA model inference.

Grisham

STILL NEED TO REARRANGE PARAGRAPH AND REWORD – CEG

Need a better introduction? – CPG

Grisham is a web-based system build on top of the PostgreSQL open source database system. Computation for visualization is performed using HTML, CSS and javascript. Calculations are performed using in-database computation through AJAX calls and client side javascript. In this section we discuss each part of *Grisham*.

Data pre-processing and topic learning

Here we describe the main pre-processing steps we perform on a collection of articles for topic modeling and search. First, we tokenize raw texts of articles with the help of the python Natural Language Toolkit (NLTK)[1] and a set of predefined regular expressions. Second, we standardize tokens by removing noise and stop-words. We use typical standardization techniques for word tokens such as *stemming*—for this project, we use the popular Porter stemming algorithm (Porter 1980) implementation in NLTK. Third, we represent each document in a sparse "bag of words" format, after building a vocabulary of corpus words. Last, we use them as input to the topic learning algorithm (Hoffman, Blei, and Bach 2010) which will in turn learn the latent topic structure of a corpus from the term co-occurrence frequencies of the corresponding documents. Components of a learned topic model includes the estimated corpus-level topics, β_j^*s, and document-level topic mixtures, θ_d^*s. As discussed, β_j^* is useful for our automatic detection of topics among articles. Similarly, θ_d^* give an idea of the topicality of a particular article given a topic. This is quite useful in finding similar articles and grouping them together. In addition, topic modeling is a type of dimensionality reduction technique that enables us to work on the topic-space rather than on the vocabulary-space.

User Model

When performing search, exploration and discovery over academic papers users may bring particular context to their search. Incorporating this information into the search process has been show to be beneficial to users (Dou, Zhicheng

[1]`http://www.nltk.org/`

and Song, Ruihua and Wen, Ji-Rong 2007; Ma, Zhongming and Pant, Gautam and Sheng, Olivia R. Liu 2007). We develop a user model that encapsulates the users personal context and integrates it into their search task.

This model is a distribution of weights for each topic. Formally, given a set of topics T the user model is defined as

$$\mathcal{U} = \{u_0, \ldots, u_{|T|}\}$$

where $u_i \in [0, 1]$ and $\sum_{t \in T} u_t = 1$. We graphically allow the user to select the weights that correspond to each topic. This allows the users to change preferences with each query for more desirable results.

The user model is used in different ways to provide better feedback to the user. After a keyword search, the document results of the search are re-ranked by calculating the KL-divergence of each document and the user model. Formally, given the set of result documents D:

$$KL(\mathcal{U}||d) = \sum_{t \in T} u_t \ln \frac{u_t}{d_t}. \tag{1}$$

where $d \in D$ and d_t is the topic proportion for document d and topic $t \in T$.

In the topic explorer, each topic row is color-coded like a heat based map based on the similarity of the user model to that topic (see Figure 2). The user can look at this heat map to adjust their topic preferences. We use equation 1 on the client side to calculate this preference. In the graph explorer the citations for the current paper is ranked using equation 1. The citations of that paper that are most similar to the user model are ranked the highest.

Topic-Based Search and Exploration

Grisham provides several ranking functions to let the user find the best articles. One way of ranking is to identify the topics of real interest, by looking at the most probable words of the estimated topics β_j^* for the corpus, and then determine relevant articles given the topic of interest. In the next section, we describe how *Grisham* helps users visualizing the estimated topics. We exploit estimated topic distributions θ_d^*s of individual articles, to rank them on relevance given a topic. Let t be the index for the topic of interest, we calculate (George et al. 2012)

$$m(d) = \ln \theta_{dt}^* + \sum_{j \neq t} \ln(1 - \theta_{dj}^*) \tag{2}$$

for all documents $d = 1, 2, \ldots, D$ in the document collection, where $j = 1, 2, \ldots, K$, and sort them to rank them on relevance. Here, we assume that each θ_{dt}^* is normalized, i.e., $\sum_{j=1}^{K} \theta_{dj}^* = 1$. Intuitively, we can see that this equation will give a high value for a document, if the probability of occurring topic t is high in that document. This means a document with a higher value of this score is highly relevant for the topic of interest t, and contains a considerable amount of words from topic t. The next section describes a visualization scheme of the ranked documents given the estimated set of topics.

Topic-Based Exploration

clint stuff – CEG

Lineage search

Demonstration

Note this section will be spread out into the other sections. – CEG

Grisham is demonstrated with the help of a web page which allows the user to perform exploratory search on DBLP conference's scientific papers (Tang et al. 2008). All the searches in the system keep in account the user model. The user model is a profile of preference provided by the user before he makes any search. This is done by specifying weightage to various topics. The *Grisham* website has a list of topics with a slider associated with each of them using which a user can specify the degree of interest in that particular topic. This array of user preference is used as the ranking factor in all the results. The web site has three basic functionality which has been classified under three different tabs of the same name. They are keyword paper search, Topic explore, and Graph explore.

Keyword Paper Explore This is a universal search facility. The user may enter one or more keywords representing topics, or author names etc. The key words are searched in the title, abstract, and the author names of all the papers and are listed. The listing is ranked based on the user model. In the first page only a few of the papers are displayed in two boxes - one containing matching words in the title, and the other in the abstract. Clicking on any box will open up a more complete list of papers.

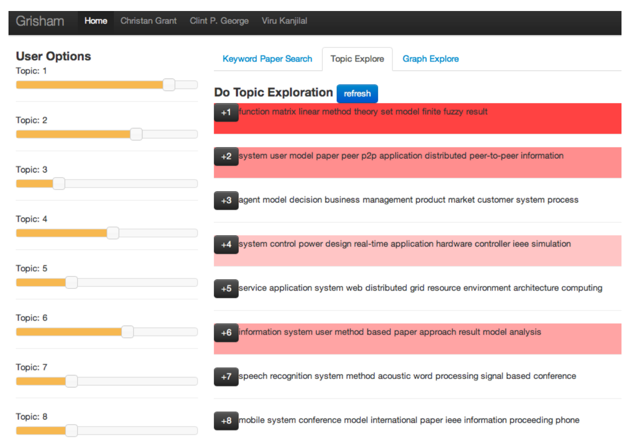

Figure 1: User configured topic exploration

Topic Explore The second tab on the website is for topic exploration and it allows a user to click on a specific topic to know more about the papers associated with the topic. Initially all the extracted topics from the corpus are shown along with their topic words. This list is color coded to distinguish the relevance of topics as indicated by the user model. The list is clickable and one a topic is clicked, relevant papers to that topic ranked based on the user model and displayed. A screenshot is shown in figure 2.

Figure 2: User configured topic exploration

Graph Explore One of the most interesting visualizations in the project is the graph explore which allows a user to conceptually drill down a graph of a paper and its citations in a recursive sequence. This is a common method of literature exploration where a user takes up a base paper and then reads up all the papers which have been cited in that base paper. This is recursively performed on the secondary papers also. Though effective, this technique tell the user which ones he should pursue and which ones he should now. The graph explore functionality allows a user to perform this in a more visually appealing and relevant manner. Once a user decides on a base paper and enters it in the system, the system will show a graph representation of its citations which are ranked based on his profile (user model). This will help him pursue the most relevant papers first. Clicking on any secondary paper will open up the graph further and list it's citations ranked taking into account the user model.

Discussion

In this section we discuss the pros and cons of the two methods.

Conclusion

We describe *Grisham*, a system for topic-based paper search and exploration given a user model. This is a demonstration of a promising new search paradigm. Any research who would like to do exploratory search or a literature reviews will find the system beneficial. During the demonstration we will allow participants to freely interact with the system. We will discuss the algorithm and formulas that drive *Grisham* with attendees.

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. Journal of Machine Learning Research 3:993–1022.

Dou, Zhicheng and Song, Ruihua and Wen, Ji-Rong. 2007. A large-scale Evaluation and Analysis of Personalized Search Strategies. In Proceedings of the 16th international conference on World Wide Web, WWW '07, 581–590. New York, NY, USA: ACM.

Dumais, S.; Furnas, G.; Landauer, T.; Deerwester, S.; Deerwester, S.; et al. 1995. Latent semantic indexing. In Proceedings of the Text Retrieval Conference.

Geman, S., and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6:721–741.

George, C.; Wang, D.; Wilson, J.; Epstein, L.; Garland, P.; and Suh, A. 2012. A machine learning based topic exploration and categorization on surveys. In Machine Learning and Applications (ICMLA), 2012 11th International Conference on, volume 2, 7–12.

Griffiths, T. L., and Steyvers, M. 2004. Finding scientific topics. Proceedings of the National Academy of Sciences 101:5228–5235.

Hoffman, M.; Blei, D. M.; and Bach, F. 2010. Online learning for latent dirichlet allocation. In NIPS.

Ma, Zhongming and Pant, Gautam and Sheng, Olivia R. Liu. 2007. Interest-based Personalized Search. ACM Trans. Inf. Syst. 25(1).

Porter, M. 1980. An algorithm for suffix stripping. Program: electronic library and information systems 14(3):130–137.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 45–50. Valletta, Malta: ELRA.

Salton, G.; Wong, A.; and Yang, C. S. 1975. A vector space model for automatic indexing. Commun. ACM 18(11):613–620.

Tang, J.; Zhang, J.; Yao, L.; and Li, J. 2008. Extraction and mining of an academic social network. In Proceedings of the 17th international conference on World Wide Web, WWW '08, 1193–1194. New York, NY, USA: ACM.

Yang, T.; Torget, A.; and Mihalcea, R. 2011. Topic modeling on historical newspapers. ACL HLT 2011 96.