Detecting multiple facets of an event using graph-based unsupervised methods

Pradeep Muthukrishnan
Dept of EECS
University of Michigan
mpradeep@umich.edu

Joshua Gerrish
School of Information
University of Michigan
jgerrish@umich.edu

Dragomir R. Radev
Dept of EECS &
School of Information,
University of Michigan
radev@umich.edu

Abstract

We propose a new unsupervised method for topic detection that automatically identifies the different facets of an event. We use pointwise Kullback-Leibler divergence along with the Jaccard coefficient to build a topic graph which represents the community structure of the different facets. The problem is formulated as a weighted set cover problem with dynamically varying weights. The algorithm is domain-independent and generates a representative set of informative and discriminative phrases that cover the entire event. We evaluate this algorithm on a large collection of blog postings about different news events and report promising results.

1 Introduction

Finding a list of topics that a collection of documents cover is an important problem in information retrieval. Topics can be used to describe or summarize the collection, or they can be used to cluster the collection. Topics provide a short and informative description of the documents that can be used for quickly browsing and finding related documents.

Inside a given corpus, there may be multiple topics. Individual documents can also contain multiple topics.

Traditionally, information retrieval systems return a ranked list of query results based on the similarity between the user's query and the documents. Unfortunately, the results returned will often be redundant. Users may need to reformulate their search to find the specific topic they are interested in. This active searching process leads to inefficiencies, especially in cases where queries or information needs are ambiguous.For example, a user wants to get an overview of the Virginia tech shootings, then the first query he/she might try is "Virginia tech shooting". Most of the results returned would be posts just mentioning the shootings and the death toll. But the user might want a more detailed overview of the shootings. Thus this leads to continuously reformulating the search query to discover all the facets of the event.

2 Related Work

Topic detection and tracking was studied extensively on newswire and broadcast collections by the NIST TDT research program (Allan et al.,). The large number of people blogging on the web provides a new source of information for topic detection and tracking.

The TDT task defines topics as "an event or activity, along with all directly related events and activities." In this paper we will stay with this definition of topic.

Zhai et al. proposed several methods for dealing with a related task, which they called *subtopic retrieval* (Zhai et al., 2003). This is an information retrieval task where the goal is to retrieve and return documents that cover the different subtopics of a given query. As they point out, the utility of each document is dependent on the other documents in the ranking, which violates the independent relevance assumption traditionally used in IR.

Blei et al. (Blei et al., 2003) proposed Latent Dirichlet Allocation (LDA), a generative model that allows sets of documents to be explained by unobserved groups of documents, each based on a single topic. The LDA model assumes the bag-

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 609–616
Manchester, August 2008

of-words model and posits that each document is composed of different topics. Specifically, each word's existence is attributed to one of the document's topic. This algorithm outputs a set of n-grams for each topic whereas our algorithm models each subtopic using a single n-gram. Due to limitations of time we were not able to compare this approach with ours. We plan to have this comparison in our future work.

To reduce the complexity of this task, a candidate set of subtopics needs to be generated that cover the document collection. We choose to use a keyphrase detection algorithm to generate topic labels. Several keyphrase extraction algorithms have been discussed in the literature, including ones based on machine learning methods (Turney, 2000), (Hulth, 2003) and tf-idf ((Frank et al., 1999)). Our method uses language models and pointwise mutual information expressed as the Kullback-Leibler divergence.

Kullback-Leibler divergence has been found to be an effective method of finding keyphrases in text collections. But identification of keyphrases is not enough to find topics in document. The keyphrases identified may describe the entire collection, or aspects of the collection. We wish to summarize subtopics within these collections.

The problem of subtopic detection is also related to novelty detection in (Allan et al.,). In this problem, given a set of previously seen documents, the task is to determine whether a new document contains new or novel content. The TREC 2002 novelty track, the task was to discard sentences that did not contain new material. This is similar to our goal of reducing redundancy in the list of returned subtopics.

In most cases, novelty detection is implemented as an online algorithm. The system has a set of existing documents they have seen up until a certain point. The task is to determine whether a new document is novel based on the previous documents. Once a decision has been made, the status of that document is fixed. The subtopic detection task differs from this because it is an offline task. The algorithm typically has access to the entire document set. Our method differs from this novelty detection task in that it has access to the entire document collection.

2.1 Existing redundancy measures

Zhang et al. examine five different redundancy measures for adaptive information filtering (Zhang et al.,). Information filtering systems return relevant documents in a document stream to a user. Examples of information filtering systems include traditional information retrieval systems that return relevant documents depending on the user's query.

The redundancy measures Zhang et al. examine are based on online analysis of documents. They identify two methods of measuring redundancy:

- Given n documents, they are considered one by one, and suppose we have processed i documents and we have k clusters. Now we need to process the $i + 1^{th}$ document. We compute the distance of the $i + 1^{th}$ document with the k clusters and add the document to the closest cluster if the distance is above a certain threshold, else we create a new cluster with only the $i + 1^{th}$ document.

- Measure distance between the new document and each previously seen document.

They evaluate several measures like set difference, geometric distance, Distributional similarity and mixture models. Evaluating the five systems, they found that cosine similarity was the most effective measure, followed by the new mixture model measure.

3 Data

We choose several news events that occurred in 2007 and 2008 based on the popularity in the blogosphere. We were looking for events that were widely discussed and commented on. The events in our collection are the top-level events that we have gathered. Table 1 lists the events that were chosen for analysis:

To help illustrate our subtopic detection method, we will use the Virginia Tech tragedy as an example throughout the rest of this paper. People throughout the blogosphere posted responses expressing support and condolences for the people involved, along with their own opinions on what caused it.

Figures 1 and 2 show two different responses to the event. The quote in figure 1 shows an example post from LiveJournal, a popular blogging community. In this post, the user is discussing his view on gun control, a hotly debated topic in the aftermath of the shooting. Figure 2 expresses another person's emotional response to this event. Both posts show different aspects of the same story. Our subtopic detection system seeks to automatically

Event	Description	Posts	Dates
iPhone	iPhone release hype	48810	June 20 , 2007 - July 7, 2007
petfoodrecall	Melamine tainted petfood recall	4285	March 10, 2007 - May 10, 2007
spitzer	Eliot Spitzer prostitution scandal	10379	March 6, 2008 - March 23, 2008
vtech	Virginia Tech shooting	12256	April 16, 2007 - April 30, 2007

Table 1: Major events summarized

identify these and other distinct discussions that occur around an event.

> After the Virginia Tech murders, there's the usual outcry for something to be done, and in particular, for more gun control. As usual, I am not persuaded. The Virginia Tech campus had gun control, which meant that Cho Seung-Hui was in violation of the law even before he started shooting, and also that no law-abiding citizens were able to draw.

Figure 1: Example blog post from LiveJournal discussing gun control (Rosen, 2007)

> ... Predictably, there have been rumblings in the media that video games contributed to Cho Seung-Hui's massacre at Virginia Tech. Jack Thompson has come out screaming, referring to gamers as "knuckleheads" and calling video games "mental masturbation" all the while referring to himself as an "educator" and "pioneer" out to "right" society. ...

Figure 2: Example blog post discussing video games (hoopdog, 2007)

Figure 3 shows a generalized Venn diagram (Kestler et al., 2005) of the cluster overlap between different keyphrases from the Virginia Tech event.

3.1 Preprocessing

Data was collected from the Blogocenter bloglines database. The Blogocenter group at UCLA has been retrieving RSS feeds from the Bloglines, Blogspot, Microsoft Live Spaces, and syndic8 aggregators for the past several years. They currently have over 192 million blog posts collected.

For each news item, relevant posts were retrieved, based on keyword searching and date of blog post. Posts from the date of occurrence of the item to two weeks after the event occurred

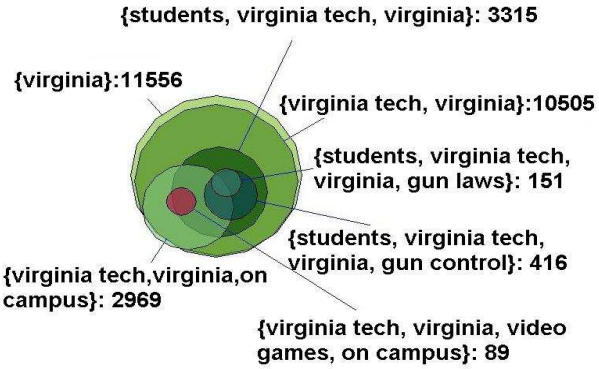

Figure 3: Generalized Venn diagram of topic overlap in the Virginia Tech collection

were gathered, regardless of the actual length of the event.

Since many RSS feeds indexed by Bloglines are from commercial news organizations or commercial sites, we had to clean up the retrieved data. Table 1 lists the event we analyzed along with basic statistics.

4 Method

Our algorithm should find discriminative labels for the different topics that exist in a collection of documents. Taken together, these labels should satisfy the following conditions:

- Describe a large portion of the collection
- The overlap between the topics should be minimal

This task is similar to Minimum Set Cover, which is NP-complete (Garey and Johnson, 1990). Therefore, trying to find the optimal solution by enumerating all possible phrases in the corpus would be impossible, instead we propose a two-step method for subtopic detection.

The first step is to generate a list of candidate phrases. These phrases should be informative and representative of all of the different subtopics. The second step should select from these phrases consistent with the two conditions stated above.

4.1 Generating Candidate Phrases

We want to generate a list of phrases that have a high probability of covering the document space. There are many methods that could be used to find informative keyphrases. One such method is using the standard information retrieval TF-IDF model (Salton and McGill, 1986).

Witten et al.(Witten et al., 1999) proposed KEA, an algorithm which generates a list of candidate keyphrases using lexical features. They keyphrases are then selected from these candidates using a supervised machine learning algorithm. This approach is not plausible for our purposes because of the following two reasons.

1. The algorithm is domain-dependent and needs a training set of documents with annotated keyphrases. But our data sets come from various domains and it is not a very viable option to create a training set for each domain.

2. The algorithm generates keyphrases for a single document, but for our purposes we need keyphrases for a corpus.

Another method is using Kullback-Leibler divergence to find informative keyphrases. We found that KL divergence generated good candidate topics.

Tomokiyo and Hurst (2003) developed a method of extracting keyphrases using statistical language models. They considered keyphrases as consisting of two features, *phraseness* and *informativeness*. Phraseness is described by them as the "degree to which a given word sequence is considered to be a phrase." For example, collocations could be considered sequences with a high phraseness. Informativeness is the extent to which a phrase captures the key idea or main topic in a set of documents.

To find keyphrases, they compared two language models, the target document set and a background corpus. Pointwise KL divergence was chosen as the method of finding the difference between two language models.

The KL divergence $D(p||q)$ between two probability mass functions $p(x)$ and $q(x)$ with alphabet χ is given in equation 1.

$$D(p||q) = \sum_{x \in \chi} p(x) log \frac{p(x)}{q(x)} \quad (1)$$

KL divergence is an asymmetric function. $D(p||q)$ may not equal $D(q||p)$.

Pointwise KL divergence is the individual contribution of x to the loss of the entire distribution. The pointwise KL divergence of a single phrase w is $\delta_w(p||q)$:

$$\delta_w(p||q) = p(w) log \frac{p(w)}{q(w)} \quad (2)$$

The phraseness of a phrase can be found by comparing the foreground n-gram language model against the background unigram model. For example, if we were judging the phraseness of "gun control", we would find the pointwise KL divergence of "gun control" between the foreground bigram language model and the foreground unigram language model.

$$\varphi_p = \delta_w(LM_f g^N || LM_f g^1) \quad (3)$$

The informativeness of a phrase can be found by finding the pointwise KL divergence of the foreground model against the background model.

$$\varphi_i = \delta_w(LM_f g^N || LM_b g^N) \quad (4)$$

A unified score can be formed by adding the phraseness and informative score: $\varphi = \varphi_p + \varphi_i$

4.2 Selecting Topic Labels

Once keyphrases have been extracted from the document set, they are sorted based on their combined score. We select the top n-ranked keyphrases as candidate phrases. This step will hereafter be referred to as "KL divergence module".

Based on our chosen task conditions regarding coverage of the documents and minimized overlap between topics, we need an undirected mapping between phrases and documents. A natural representation for this is a bipartite graph where the two sets of nodes are phrases and documents. Let the graph be: $G = (W, D, E)$ where W is the set of candidate phrases generated by the first step and D is the entire set of documents. E is the set of edges between W and D where there is an edge between a phrase and a document if the document contains the phrase.

We formulate the task as a variation of Weighted Set Cover problem in theoretical computer science. In normal Set Cover we are given a collection of sets S over a universe U, and the goal is to select a minimal subset of S such that the whole universe, U is covered. Unfortunately this problem is NP-complete (Garey and Johnson, 1990), so we must

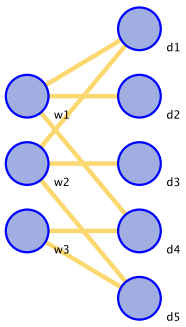

Figure 4: Bipartite graph representation of topic document coverage, where the d_i's are the documents and the w_i's are the n-grams

settle for an approximate solution. But fortunately there exist very good α-approximation algorithms for this problem (Cui, 2007).

The difference in Weighted Set Cover is that each set has an associated real-valued weight or cost and the goal is to find the minimal or maximal cost subset which covers the universe U.

In our problem, each phrase can be thought of as a set of the documents which contain it. The universe is the set of all documents.

4.3 Greedy Algorithm

To solve the above problem, we propose a greedy algorithm. This algorithm computes a cost for each node iteratively and selects the node with the lowest cost at every iteration. The cost of a keyphrase should be such that we do not choose a phrase with very high coverage, like "Virginia" and at the same time not choose words with very low document frequency since a very small collection of documents can not be judged a topic.

Based on these two conditions we have come up with a linear combination of two cost components, similar to Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998).

1. Relative Document Size:

$$f_1(w_i) = \frac{|adj(w_i)|}{N} \tag{5}$$

where $|adj(w_i)|$ is the document frequency of the word.

This factor takes into account that we do not want to choose words which cover the whole document collection. For example, phrases such as "Virginia" or "Virginia tech" are bad subtopics, because they cover most of the document set.

2. Redundancy Penalty:

We want to choose elements that do not have a lot of overlap with other elements. One measure of set overlap is the Jaccard similarity coefficient:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

$$f_2(w_i) = 1 - \frac{\sum_{w_j \in W - w_i} J(w_i, w_j)}{|W| - 1} \tag{7}$$

This component is essentially $1-$ average Jaccard similarity.

We calculate the pairwise Jaccard coefficient between the target keyphrase and every other keyphrase. The pairwise coefficient vector provides information on how much overlap there is between a keyphrase and every other keyphrase. Phrases with a high average Jaccard coefficient are general facets that cover the entire collection. Phrases with a low Jaccard coefficient are facets that cover specific topics with little overlap.

3. Subtopic Redundancy Memory Effect

Once a keyphrase has been chosen we also want to penalize other keyphrases that cover the same content or documents. Equation 8 represents a redundancy "memory" for each keyphrase or subtopic. This memory is updated for every step in the greedy algorithm.

$$R(w_i) = R(w_i) + J(w_i, w_j) \tag{8}$$

where w_j is the newly selected phrase.

A general cost function can be formed from a linear combination of the three cost components. We provide two parameters, α and β to represent the trade-off between coverage, cohesiveness and intersection. For our experiments, we found that an α value of 0.7 and a β value of 0.2 performed well.

$$\begin{aligned} cost(w_i) = \alpha \times f_1(w_i) \\ +\beta \times f_2(w_i) \\ +(1-(\alpha+\beta)) \times R(w_i) \end{aligned} \tag{9}$$

The pseudocode for the greedy algorithm is given in Figure 5. It should be noted that the algorithm requires the costs to be recomputed after every iteration. This is because the cost of a keyphrase may change due to a change in any of the three components. This is because after selecting a keyphrase, it might make another keyphrase redundant, that is, covering the same content. This makes the whole problem a dynamic weighted set cover problem. Hence, the performance guarantees associated with the greedy algorithm for the Weighted Set Cover problem do not hold true for the dynamic version.

Algorithm *Greedy algorithm for weighted set-cover*
Input: Graph $G = (W, D, E)$
1. N: number of documents to cover
2.
Output: Set of discriminative phrases for the different topics
3. $W = \{w_1, w_2, \ldots, w_n\}$
4. $W_{chosen} = \emptyset$
5. $num\_docs\_covered = 0$
6. **while** $num\_docs\_covered < N$
7. **do for** $w_i \in W$
8. **do** $cost(w_i) = \alpha \times f_1(w_i)$
9. $+\beta \times f_2(w_i)$
10. $+(1 - (\alpha + \beta)) \times R(w_i)$
11. $w_{selected} = \underset{w}{\operatorname{argmax}}\, cost(w_i)$
12. **for** $w_i \in W$
13. **do** $R(w_i) = R(w_i) + J(w_{selected}, w_i)$
14. $num\_docs\_covered = num\_docs\_covered + |adj(w_{selected})|$
15. $W_{chosen} = W_{chosen} \cup \{w_{selected}\}$
16. $W = W - \{w_{selected}\}$
17. $D = D - adj(selected)$
18. return W_{chosen}

Figure 5: A greedy set-cover algorithm for detecting sub-topics

5 Experiments

As a baseline measure, we extracted the top k phrases from the word distribution as the topic labels. As a gold standard, we manually annotated the four different collections of blog posts. Each annotator generated a list of subtopics.

6 Evaluation

In evaluating topic detection, there exist two categories of methods, *intrinsic* and *extrinsic* (Liddy, 2001). Extrinsic methods evaluate the labels against a particular task whereas intrinsic methods measure the quality of the labels directly. We provide intrinsic and extrinsic evaluations of our algorithm.

To evaluate our facet detection algorithm, we created a gold standard list of facets for each data set. A list of the top 300 keyphrases generated by the KL divergence module was given to two evaluators. The evaluators were the first and second author of this paper. The evaluators labeled each keyphrase as a positive example of a subtopic or a negative example of a subtopic. The positive examples taken together form the gold standard. For this evaluation process we defined a positive subtopic as a cohesive collection of documents discussing the same topic.

Cohen's Kappa coefficient (Cohen, 1960) was calculated for the gold standard. Table 6 lists the κ value for the four data sets.

iPhone	petfoodrecall	spitzer	vtech
0.62	0.86	0.77	0.88

Table 2: Kappa scores for the gold standard

The kappa scores for the *petfoodrecall* and *vtech* datasets showed good agreement among the raters, while the spitzer data set had only fair agreement. For the *iPhone* data set, both evaluators had a large amount of disagreement on what they considered subtopics.

A separate group of two evaluators was given the output from our graph-based algorithm, a list of the top KL divergence keyphrases of the same length, and the gold standard for all four data sets. Evaluators were asked to rate the keyphrases on a scale from one to five, with one indicating a poor subtopic, and five indicating a good subtopic. The number k of subtopics for the algorithm was cutoff where the f-score is maximized. The same number of phrases was chosen for KL divergence as well. Table 3 lists the cutoffs for the four data sets.

iPhone	*Petfood recall*	*Spitzer*	*Vtech*
25	30	24	18

Table 3: Number of generated subtopics for each collection.

In addition, the precision, F-score, coverage and average pairwise Jaccard coefficient were calculated for the four data sets. Precision, recall and the F-score are given in table 4. The precision, recall and F-score for the gold standards is one. The others are shown in table 5. Average pairwise Jaccard coefficient is calculated by finding the Jaccard coefficient for every pair of subtopics in the output and averaging this value. This value is a measure of the redundancy. The average relevance is a normalized version of the combined "phrase-

ness" and "informativeness" score calculated by the keyphrase detection module. This value is normalized by dividing by the KL divergence for the entire 300 phrase list. This provides a relevancy score for the output.

Data set	Precision	Recall	F-score
iphone			
KL-Divergence	0.08	0.10	0.09
Graph-based method	0.52	0.60	0.56
petfoodrecall			
KL-Divergence	0.37	0.39	0.38
Graph-based method	0.61	0.57	0.59
spitzer			
KL-Divergence	0.10	0.08	0.09
Graph-based method	0.79	0.59	0.68
vtech			
KL-Divergence	0.05	0.06	0.05
Graph-based method	0.72	0.76	0.74

Table 4: Precision, recall and F-score for the baseline and graph-based algorithm.

Data set	Coverage	Average pairwise JC	Normalized KL divergence	Human rating
iphone				
KL-Divergence	40168	0.08	18.19	1.92
Gold standard	12977	0.02	2.81	3.13
Graph-based	9850	0.01	1.98	2.82
petfoodrecall				
KL-Divergence	4280	0.18	19.53	1.82
Gold standard	2659	0.05	4.30	3.43
Graph-based	2055	0.01	1.75	2.81
spitzer				
KL-Divergence	9291	0.19	22.90	1.33
Gold standard	4036	0.03	2.29	3.31
Graph-based	2468	0.01	1.60	2.88
vtech				
KL-Divergence	12215	0.29	24.61	1.61
Gold standard	5058	0.03	2.79	3.76
Graph-based	4342	0.01	1.66	3.28

Table 5: Coverage, overlap and relevance and evaluation scores for the gold standard, baseline and graph-based method.

7 Results

Table 6 shows some of the different subtopics chosen by our algorithm for the different data sets. There is no manual involvement required in the algorithm except for the intial preprocessing to remove commercial news feeds and spam posts. Our graph-based method performs very well and almost achieves the gold standard's rating. The F-score for the *iPhone* data set was only 0.56, but we believe part of this may be because this data set did not have clearly defined subtopics, as shown by the low agreement (0.62) among human evaluators.

Spitzer	*Petfood recall*
Ashley Alexandra Dupre	Under Wal-Mart
Oberweis	Xuzhou Anying
Emperor's club	People who buy
Governor of New	Cuts and Gravy
Spitzer's resignation	Cat and Dog
Dr Laura	Cats and Dogs
Mayflower hotel	Food and Drug
Sex workers	Cyanuric acid
former New york	recent pet
High priced prostitution	industrial chemical
McGreevey	massive pet food
Geraldine Ferraro	Royal canin
High priced call	Iams and Eukanuba
legally blind	Dry food
money laundering	
Virginia Tech shooting	*iPhone*
Korean American	Photo sent from
Gun Ownership	Waiting in line
Holocaust survivor	About the iPhone
Mentally ill	Unlimited data
Shooting spree	From my iPhone
Don Imus	cell phones
Video Games	Multi-touch
Gun free zone	Guided tour
West Ambler Johnston	iPhone launch
Columbine High school	Walt Mossberg
Self defense	Apple Inc
Two hours later	Windows Mobile
Gun violence	June 29th
Seung Hui Cho	Web browser
Second Amendment	Activation
South Korean	

Table 6: Different topics chosen by the graph-based algorithm for the different data sets

Figure 6 shows the trade off between coverage and redundancy. This graph clearly shows that the overlap between the subtopics increases very slowly as compared to the number of documents covered. The slope of the curves increases slowly when the number of documents to be covered is small and later increases rapidly. This means that initially there are a lot of small focused subtopics and once we have selected all the focused ones the algorithm is forced to pick the bigger topics and hence the average pairwise intersection increases.

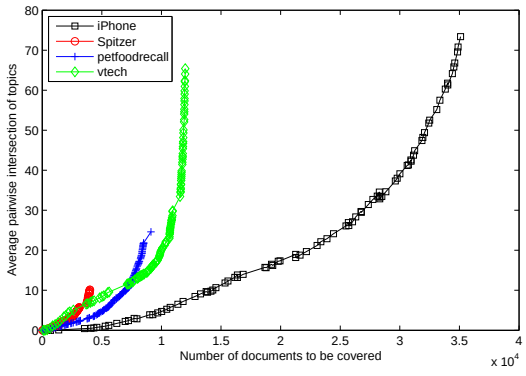

Figure 6: Subtopic redundancy vs. coverage

8 Conclusion

We have presented a new algorithm based on weighted set cover for finding subtopics in a corpus of selected blog postings. The algorithm performs very well in practice compared to the baseline standard, which outputs the top keyphrases according to the Kullback-Leibler divergence method. While the baseline standard outputs keyphrases which are redundant, in the sense, they cover the same documents, the graph-based method outputs keyphrases which have very little intersection. We provide a new method of ranking keyphrases that can help users find different facets of an event.

The identification of facets has many applications to natural language processing. Once facets have been identified in a collection, documents can be clustered based on these facets. These clusters can be used to generate document summaries or for visualization of the event space.

The keyphrases themselves provide a succinct summary of the different subtopics. In future work, we intend to investigate summarization of documents based on subtopic clustering using this method.

9 Acknowledgments

This work was supported by NSF grants IIS 0534323 “Collaborative Research: BlogoCenter - Infrastructure for Collecting, Mining and Accessing Blogs” awarded to The University of Michigan and ”iOPENER: A Flexible Framework to Support Rapid Learning in Unfamiliar Research Domains”, jointly awarded to U. of Michigan and U. of Maryland as IIS 0705832.” Also we would like to thank Vahed Qazvinian and Arzucan Özgür for helping with the evaluation and their valuable suggestions. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Allan, James, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. *SIGIR 2003*, pages 314–321.

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

Carbonell, Jaime G. and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *SIGIR 1998*, pages 335–336.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37.

Cui, Peng. 2007. A tighter analysis of set cover greedy algorithm for test set. In *ESCAPE*, pages 24–35.

Frank, Eibe, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. *Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673.

Garey, Michael R. and David S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman.

hoopdog. 2007. Follow-up: Blame game. http://hoopdogg.livejournal.com/39060.html.

Hulth, Anette. 2003. Improved automatic keyword extraction given more linguistic knowledge. *EMNLP 2003*, pages 216–223.

Kestler, Hans A., Andre Muller, Thomas M. Gress, and Malte Buchholz. 2005. Generalized venn diagrams: a new method of visualizing complex genetic set relations. *Bioinformatics*, 21:1592–1595, April.

Liddy, Elizabeth. 2001. Advances in automatic text summarization. *Information Retrieval*, 4:82–83.

Rosen, Nicholas D. 2007. Gun control and mental health. http://ndrosen.livejournal.com/128715.html.

Salton, G. and M. J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. New York, NY, USA.

Tomokiyo, Takashi and Matthew Hurst. 2003. A language model approach to keyphrase extraction. *ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, pages 33–40.

Turney, Peter D. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336.

Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *1st ACM/IEEE-CS joint conference on Digital libraries*, pages 254–255.

Zhai, Chengxiang, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. *SIGIR 2003*, pages 10–17.

Zhang, Yi, James P. Callan, and Thomas P. Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR 2002*, pages 81–88.