Discovery of Numerous Specific Topics via Term Co-Occurrence Analysis

Omid Madani
SRI International, AI Center
Menlo Park, CA 94025
madani@ai.sri.com

Jiye Yu
Stanford University, EPGY
Stanford, CA 94305
jyu2009@stanford.edu

ABSTRACT

We describe efficient techniques for construction of large term co-occurrence graphs, and investigate an application to the discovery of numerous fine-grained (specific) topics. A topic is a small dense subgraph discovered by a random walk initiated at a term (node) in the graph. We observe that the discovered topics are highly interpretable, and reveal the different meanings of terms in the corpus. We show the information-theoretic utility of the topics when they are used as features in supervised learning. Such features lead to consistent improvements in classification accuracy over the standard bag-of-words representation, even at high training proportions. We explain how a layered pyramidal view of the term distribution helps in understanding the algorithms and in visualizing and interpreting the topics.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Clustering*

General Terms

Algorithms

Keywords

unsupervised learning, text mining, co-occurrence graphs, topic discovery, feature induction, feature augmentation

1. INTRODUCTION

Consider trying to obtain quick insights into how a term is mentioned in a corpus of text (in news articles, emails, abstracts, postings, and so on). The term of interest may correspond to an entity, such as a person, an organization, or a product, or to an event, etc. (*e.g.*, "Bill Clinton", "religion", "oil spill"). Reading each document that contains the term is one approach, but this can be very time consuming, and it may not readily reveal the salient "topics", *i.e.*, events, activities, roles, or associations that the term (the concept corresponding to the term) participated in. Algorithms for quickly extracting such *specific* topics would find applications in intelligence analysis, discovery of user interests, document and corpus summarization, document tagging and routing, and so on. In this paper, we explore the use of co-occurrence graphs on terms for the discovery of such topics. Co-occurrence graphs find diverse uses in various information retrieval and language processing tasks [4, 2, 14, 15, 5]. In this paper, we first briefly present techniques for large-scale construction of such graphs, and then develop an application to fine-grained topic discovery.

In many scenarios, such as for the topic discovery application described here, the graphs may need to be continually updated or otherwise rebuilt periodically, therefore memory and/or time efficiency can be important. We describe methods for efficient construction of a type of co-occurrence graph wherein edges reflect conditional probabilities, or a close variant, as we explain. We compute term co-occurrence information by sliding a window over each term appearing in a document (*e.g.*, [9, 14, 5]). During graph construction, new edges reflecting co-occurrence may be added to the graph, and edges with small weights may be dropped. We distinguish between three types of edge dropping, and explain the effects of each. Unlike other types of graphs, the nodes (terms) in the co-occurrence graph should not be treated the same way: the graph should not be viewed as "flat". We explain how consideration of term frequencies, in particular the *pyramid view* of the term distribution, finds several uses in algorithm design and in visualizing and making a better sense of the discovered topics.

We mine an association graph for relatively small dense subgraphs (10s of terms), or semicliques, *i.e.*, our "topics." We find that the semicliques correspond to a variety of subjects and events such as natural disasters, elections, sports events, group and organizational activities, and political controversies. Naturally, a given term, such as a person's name, can participate in several topics, often reflecting different senses of the term, or roles for the corresponding referent. We explore the application of the topics as features for supervised learning, finding that the topics improve performance even at high training proportions. We also compare to latent Dirichlet allocation [1], and highlight the advantages of co-occurrence analysis for fine-grained topic discovery. Thus, we provide evidence that topic discovery via co-occurrence analysis is an effective way to mine text collections, complementing the more commonly studied methods such as document clustering and existing term-document analysis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

	N	#uni	|V|	|d|	T
newsgroups	19k	96k	122k	208	6mins
TDT5	280k	394k	466	350	6 hours
Reuters	800k	600k	730k	237	12 hours

Table 1: The data sets we use. N, #uni, $|V|, |d|$, and T, respectively denote the number of documents in corpus, the number of unigrams (words), the vocabulary size (includes bigrams), avg. document length, and the time it took to build the graph.

```
GenerateGraph($L$,$W_0$, $DF_0$, $K$)
/* process documents in a random order */
1. For each document $d$ in corpus:
1.1 For each term $v$ in document $d$:
1.1.1 $DF(v) \leftarrow DF(v) + 1$ /* update doc frequency */
    /* compute the proximity value of term u wrt to v */
1.1.2 For each term $u$, $w^d_{v,u} \leftarrow \frac{\sum_{1 \le i \le tf^d(v)} w^d_{v,u,i}}{tf^d(v)}$
1.1.3 $w'_{v,u} \leftarrow w'_{v,u} + w^d_{v,u}$ /* update cumul. edge value */
1.1.4 OPTIONAL: If $DF(v) > DF_0$, /* v seen enough? */
      /* If so, drop its small connections */
1.1.4.1 For all neighbors $u$, if $\frac{w'_{v,u}}{DF(v)} < W_0$ then $w'_{v,u} \leftarrow 0$
/* After processing all documents, finalize */
/* graph weights (and, optionally, drop more edges) */
2. For each pair of terms $v$ and $u$ where $w'_{v,u} > 0$:
2.1 $w_{v,u} \leftarrow \frac{w'_{v,u}}{DF(v)}$ /* the edge weight */
2.2 If $w_{v,u} < max(\frac{K}{DF(v)}, W_0)$, $w_{v,u} \leftarrow 0$. /* too weak.. */
2.3 If $\frac{w_{v,u}}{w_{v_0,u}} < 10$, $w_{v,u} \leftarrow 0$./* "PMI filtering" */
```

Figure 1: Graph generation. $tf^d(v)$ denotes number of occurrences of v in document d, and $w^d_{v,u,i}$ denotes the proximity value of u with respect to v for the ith occurrence of v (in Boolean weighting, simply 0, if outside window, otherwise 1). The special term v_0 is used in computing PMI stats (step 2.3). v_0 is placed at every word occurrence, and weights from it ($w_{v_0,u}$) are computed. Defaults: $L = 20, W_0 = 0.01, DF_0 = 50, K = 3$.

techniques. The expanded version of this paper contains further descriptions and experiments [11].

2. DATA AND GRAPH CONSTRUCTION

Table 1 shows our data sets: newsgroups [7], the Reuters collection [13], and TDT5 (Topic Detection and Tracking, for 2004, [8]). We performed some tokenization, such as lower casing words, some stemming, and changing numbers to NUM. The vocabulary includes both unigrams and a subset of possible bigrams and trigrams (phrases such as "new jersey"). We generated the ngrams via the same graph construction process, which we explain next. Experiments were run on a Dual-Core AMD Opteron (25GB RAM, 2.8GHz), and the graph construction and semiclique discovery code was written in both Java and C++.

Figure 1 presents our main graph construction algorithm. The algorithm begins with an empty graph (no edges), and processes the corpus documents in a random order. The algorithm keeps and updates edge weights for each term as well as a special term v_0 (v_0 is used to determine statistical significance). The edge weights could simply be conditional probabilities. In particular, $w_{v,u}$ can be viewed as $P(u|v)$, meaning the probability that term u is seen sufficiently close to v, within L terms on either side of v, given that a random position is picked from a randomly picked document, and v is seen in that position. More generally, they can be expected proximity values [11]. By sliding a window of size L over each position (term occurrence) in the document, and keeping counts, the weights (conditional probabilities) are updated. The algorithm can drop edges during graph construction for memory efficiency (there are other for preserving efficiency). Finally, after the corpus processing is finished, the graph is pruned further: edge weights that are not sufficiently statistically significant and those that do not pass a test of sufficiency of "informativeness" or "surprise-level" are dropped (like point-wise mutual-information ([12])). Instead of the two-tier updates, first summarizing (averaging) the document, then updating the global weights, we could have done the updates in one step. The two tier update lessens the effects of long documents. This choice is also application dependent.

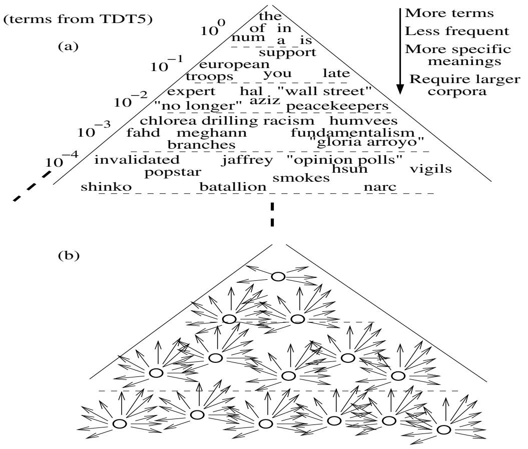

Figure 2: The layered pyramidal view of terms. (a) A selection of terms in TDT5 placed in the pyramid according to their document frequency (df). (b) Terms with similar df tend to reciprocate directed edges, and due to edge dropping during or after graph construction, terms tend to connect to other terms within their own layer or to higher layers.

The Pyramid View. Terms (single words and meaningful phrases) are approximately distributed according to Zipf's law, the number of terms growing rapidly with decreasing frequency. The terms can therefore be viewed as residing in a pyramid, where a few most frequent terms residing at the top (Figure 2(a)). We note that if two terms have roughly the same frequency, or equal (nonzero) marginal probabilities, then their conditional probabilities must be equal: $P(u) = P(v) \Leftrightarrow P(u|v) = P(v|u)$. More generally, with $k > 0$,

$$P(u) = kP(v) \Leftrightarrow P(u|v) = kP(v|u) \qquad (1)$$

(as $kP(v|u) = k\frac{P(v,u)}{P(u)} = k\frac{P(v,u)}{kP(v)} = P(u|v)$). The implications of equivalence (1) are several for the way the graph computation works: edges are reciprocated and have a similar weight when the terms are in the same tier of the pyramid, *i.e.*, when the terms have close frequencies. Moreover, with the edge dropping using the minimum threshold W_0,

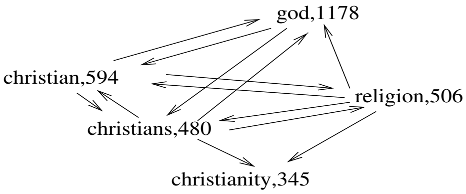

Figure 3: The immediate neighbors of "christianity" (together with doc. frequencies), with connection weight of at least 0.05 from "christianity" to them (newsgroup data). All the edges (of weight greater than $W_0 = 0.01$) among the neighbors are shown. The edges from christianity are not shown. In this corpus, "god" appears in more contexts (or has more meanings) than the others.

```
GenerateSemiCliques($G(V, E)$, $k$, $r$, $\tau$)
1. $Q \leftarrow \emptyset$. /* Q is the global set of semicliques */
2. Repeat k times for each term $v$.
 2.1. $S \leftarrow \{v\}$. /* S is a candidate semiclique */
 2.2. Repeat until no more additions possible:
 2.2.1 For some $u \notin S$: /* examine neighbors of $S$ */
 2.2.1.2 If addition of $u$ keeps $S$ $r$-connected, $S \leftarrow S \cup \{u\}$.
 2.3. If $|S| \geq 3$, add $S$ to $Q$ if non-redundancy conditions
      are met (see Section 3).
```

Figure 4: A topic is a maximal semiclique. Each term in a semiclique S (a set of terms) is connected to at least $r(|S| - 1)$ other terms in the semiclique ($r \geq 0.5$ in our experiments). Edges are undirected here: It is assumed there is a connection as long as a weight in one direction in the co-occurrence graph exceeds W_0. A newly generated semiclique is added to Q if no other semiclique S' has high overlap with it ($\frac{|S \cap S'|}{min(|S|, |S'|)} \leq 0.5$), or otherwise S has higher size (in which case the others with high overlap are removed from Q).

outgoing edges of a term tend to connect it to terms in the same or above layers more than to terms in the lower layers (Figure 2(b)). An effect that dropping edges by PMI (or similar constraints) has is that it tends to cut edges that go all the way to top tiers (most frequent words). Such terms are often too general to be informative. With our edge removals, we found that the outdegrees were in the 10s and low 100s for all terms except the very infrequent ones.

An Example Neighborhood. Figure 3 shows the neighborhood for "christianity" from processing the newsgroups data set, for those neighbors whose edge weight are greater than 0.05 (and pass the PMI constraint). Showing the neighbors in a hierarchy helps understand the generality of terms in the corpus. As may be expected, edges going upward have higher weights than the corresponding reverse edge.

3. SEMICLIQUES: DISCOVERY AND USE

Although it is generally difficult to determine the boundaries for groupings of graph nodes, *i.e.*, deciding what to include and what not, selection of groups of terms ("communities") with high inter-connectivity, semicliques, can be useful in various tasks, such as discovering events or topics in test corpora, document tagging, discovery of user interests, and so on. Semicliques, for us, are simply maximal sufficiently connected subgraphs. Sufficient connectivity means each term is connected to at least r fraction of the other terms in the semiclique. The process for semiclique finding is given in Figure 4. The input is a co-occurrence graph. Here, unless otherwise specified, this is the graph output using all the steps in Figure 1 (with $L = 20$, $W_0 = 0.005$, PMI filtering, etc), with edges under $W_{min} = 0.01$ (the minimum weight of interest) removed. The graph is made undirected (either direction suffices for connection), and we also ignore edge weights in clique discovery. Let Q denote the set of semicliques, initially empty. Repeatedly, a term is picked as seed, one or more iterations of maximal semiclique generation is conducted, and those candidate semiclique (with minimum size 3) that either do not highly overlap with members of Q, or otherwise have more nodes, are added to Q (the smaller members with high overlap are removed), in order to avoid redundancy. Various operations such as checks for adding a term to a semiclique or checking for intersection size among semicliques are performed efficiently via efficient data structures such as hash maps. Semiclique generation takes in the order of several minutes (newsgroups) to a few hours for the larger data sets. Thousands of semicliques are created (*e.g.*, ten thousand for newsgroups with $r = 0.6$, over a hundred thousand on Reuters).

We briefly explain some of the choices for semiclique discovery. The constraint $r \geq 0.5$ is a simple way of keeping the topic focused and avoiding the loose semicliques (*i.e.*, with small cuts, or possibly spanning multiple meanings of a term). We experimented with different parameters for window size L, and $L \leq 5$ yielded very few semicliques, while $L \geq 40$ yielded often relatively topically loose semicliques. We experimented with several techniques for semiclique discovery (greedy, etc). We found that multiple random walks from each node yields considerably more useful semicliques than deterministic (*e.g.*, greedy) generation.

The generated semicliques, reflecting highly connected terms that tend to occur close to one another, tend to be very interpretable. Two example semicliques containing "John Kerry" (democratic senator and 2004 presidential candidate) from TDT5 are:

```
1. [(joseph lieberman,153)(john kerry,394)
(democratic,14198)(gephardt,181)(sen,3138)(sens,174)
(massachusetts 1524)(presidential,9908)]
2. [(aspirant,36)(democratic,14198)(john kerry,394)
(presidential,9908)(gani fawehinmi,22)(sen,3138)]
```

Tagging Documents with Topics. Intuitively, the higher the fraction of the semiclique members that appear in the document, the more relevant the topic is to the document. For determining the tag value of topic S on a document d (both treated as sets), we simply consider the terms in the intersection of the document and the topic ($d \cap S$), and experimented with Boolean (plain) weighting, log (document) frequency weighting, and inverse document frequency weighting:

$$\frac{\sum_{v \in d \cap S} f(v)}{\sum_{v \in S} f(v)}, \tag{2}$$

where $f(v) = 1$ for Boolean weighting, and $log(N/df(v))$ and $1/df(v)$ for log and inverse frequency weighting respectively, where N is corpus size. Boolean weighting was noticeably inferior in both our subjective assessments and in our supervised learning experiments (see next). Log frequency performed the best.

Training Portion →	0.8	0.5	0.1
	Newsgroups		
semicliques	14.4, 0.44%	15.3, 0.50%	15.3, 0.77%
LDA k=100	11.8, 0.15%	14.6, 0.45%	18.8, 2.1%
LDA k=300	13.3, 0.3%	14.9, 0.41%	18, 1.29%
LDA k=600	11.6, 0.12%	14.4, 0.25%	17, 0.45%

Table 2: Improvements, via augmenting tfidf feature vectors, from using semiclique features or LDA topics under different training proportions (0.8, 0.5, 0.1). The average number of wins in 10 trials (from 20 classes), and average absolute improvement in Max F1 is given. LDA with k=300 and k=600 were stopped respectively after 30 and 10 iterations.

Boost in Supervised Learning. If semicliques capture many of the topics discussed in a corpus, we expect that some such semicliques would align well with the high-level classes assigned to the documents by humans ("sports", "politics", etc). More generally, we expect that the semicliques capture useful regularities. We found that augmenting the regular word-based features with the topic features (those with log df or inv. df tag value exceeding a threshold, such as 0.1 or higher) significantly improved supervised learning performance (the F1 score), even at high training proportions (such as 80-20 splits, using SVM classifiers). We compared with LDA topics [1] on newsgroups as well as a subset of Reuters (under different number of hidden topics, k=20, 40, 100, 300, 600). Table 2 displays a subset of our results. Semiclique features, being in the thousands, had an advantage at higher training proportions, while LDA topics yielded higher performance boosts at lower training proportions. LDA topics were substantially slower to compute for k in the 100s.[1] With fairly broad (LDA) topics, it is hard to discern what the role of a given term is within the topic. On the other hand, recovering more general (or abstract) topics, for example via relaxing the connectivity ratio r and/or increasing the window size, should result in semicliques that are broader and in effect more similar to LDA topics. This is a future direction.

4. RELATED WORK

Term co-occurrence (or co-location) relations find a number of applications in diverse tasks, such as semantic distance computation, synonymy, and query expansion (*e.g.*, [4, 2, 14, 15, 5]). Focusing on recovering salient connections and removing weak edges has also been used in large-scale multiclass learning [10]. Here, the goal is not pure prediction, but learning associations. LSI is an elegant matrix method based on term-document co-occurrence patterns [3], and together with the probabilistic extensions, these methods have become very versatile and are well-studied (*e.g.*, [6, 1]). As we saw with LDA, these methods appear to be more appropriate for discovering relatively few (low 100s) and relatively broad topics, and interpretation and efficiency can be an issue. There is substantial work on computing semicliques (or quasicliques), dense subgraphs, clustering, and communities in networks. Here, it is very important to allow for overlapping groupings to support multiple senses and events, unlike much (graph) clustering work.

5. SUMMARY

We presented efficient algorithms for computing graphs of co-occurrence relations, and explored extracting highly dense subgraphs as candidate topics. We highlighted the utility of the layered pyramid view of terms. The proposed approach yields numerous fine-grained topics that are interpretable groupings of terms. We observed that the topics could augment the standard term features to improve supervised learning performance. There are many directions for future work, for example in exploring variations to topic finding methods.

Acknowledgments. Many thanks to Shahin Saadati for his assistance, and to Erik Yeh for suggestions on supervised learning. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004, in the context of research on an adaptive cognitive system (CALO), under DARPA's PAL (Perceptive Assistant that Learns) program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), or the Air Force Research Laboratory (AFRL).

6. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[2] P. F. Brown, V. J. Della-Pietra, P. V. deSouza, C. J.Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[3] S. Deerwester, S. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 1990.

[4] L. Doyle. Indexing and abstracting by association. System Development Corporation, Unisys Corporation, 1962.

[5] D. Freitag, M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, and Z. Wang. New experiments in distributional representations of synonymy. In *CoNLL*, 2005.

[6] T. Hofmann. Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAI*, 1999.

[7] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[8] LDC. Topic detection and tracking. In *http://www.ldc.upenn.edu/Projects/TDT/*, 2005.

[9] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behaviour Research Methods, Instruments, and Computers*, 2, 1996.

[10] O. Madani, M. Connor, and W. Greiner. Learning when concepts abound. *J. of Machine Learning Res.*, 10, 2009.

[11] O. Madani and J. Yu. Discovery of numerous specific topics via term co-occurrence analysis. Technical report, in preparation, 2010.

[12] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[13] T. Rose, M. Stevenson, and M. Whitehead. The Reuters Corpus Volume 1 - from yesterday's news to tomorrow's language resources. In *Third International Conference on Language Resources and Evaluation*, 2002.

[14] H. Schutze and J. O. Pedersen. A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing and Management: an International Journal*, 33, 1997.

[15] A. Veing and P. van der Weerd. Conceptual grouping in word co-occurrence networks. In *IJCAI*, 1999.

[1] Several days. We used the C implementation available at cs.princeton.edu/~blei/topicmodeling.html