GDClust: A Graph-Based Document Clustering Technique

M. Shahriar Hossain, Rafal A. Angryk
Department of Computer Science,
Montana State University, Bozeman,
MT 59715, USA
E-mail: {mshossain, angryk}@cs.montana.edu

Abstract

This paper introduces a new technique of document clustering based on frequent senses. The proposed system, GDClust (Graph-Based Document Clustering) works with frequent senses rather than frequent keywords used in traditional text mining techniques. GDClust presents text documents as hierarchical document-graphs and utilizes an Apriori paradigm to find the frequent subgraphs, which reflect frequent senses. Discovered frequent subgraphs are then utilized to generate sense-based document clusters. We propose a novel multilevel Gaussian minimum support approach for candidate subgraph generation. GDClust utilizes English language ontology to construct document-graphs and exploits graph-based data mining technique for sense discovery and clustering. It is an automated system and requires minimal human interaction for the clustering purpose.

1. Introduction

The goal of this paper is to present a new, human-like hierarchical clustering technique driven by recent discoveries in the area of graph-based data mining. Our approach is motivated by typical human behavior, when given a task of organizing multiple documents. As an example, consider the behavior of scientific book editor, who needs to organize multiple research papers into a single book volume, with a hierarchical table of contents. Typically, research papers, even when coming from the same area, are written (1) in multiple writing styles, (2) on different levels of detail, and (3) in reference to different aspects of an analyzed area. Instead of searching for identical words and counting their occurrences, like many well-known computer-based text clustering techniques do [2]–[4], the human brain usually remembers only a few crucial keywords representing senses, which provide the editor with a compressed representation of the documents. These senses are then used to fit a given research paper into a book organization scheme, reflected by the table of contents. In our work, we replace editor's knowledge with ontology and use it to discover common senses that can then be used to organize documents.

In GDClust, we construct document-graphs from text documents and apply an Apriori paradigm [18] for discovering frequent subgraphs from them. We utilize a hierarchic representation of English terms, WordNet [1], to construct document-graphs. Since each document can be represented as graph of related terms, they can be searched for frequent subgraphs using graph mining algorithms. We aim to cluster documents depending on the similarity of the subgraphs in the document-graphs. GDClust enables clustering of documents providing humanlike sense-based searching capabilities, rather than focusing only on the co-occurrence of frequent terms. It follows the way human beings process the text data. As the outcome of GDClust, we achieve subgraphs of meaningful senses.

The rest of the paper is organized as follows. Section 2 describes the literature review of this work. The overall GDClust system is portrayed in section 3. Some illustrative experimental results are discussed in section 4. We conclude the paper in section 5.

2. Literature review

The benefit of GDClust is that it is able to group documents in the same cluster even if they do not contain common keywords, but still possess the same sense. Existing clustering techniques cannot perform this sort of discovery [2]–[4] or do this work only to a limited degree (e.g., Latent Semantic Index (LSI) [5]).

With GDClust, we aim to develop a document clustering technique that is able to cluster documents depending on senses rather than depending on the

exact match of keywords. Developing algorithms that discover all frequently occurring subgraphs in a large graph database is particularly challenging and computationally intensive, as graph and subgraph isomorphism play a key role throughout the computations [6]. Nevertheless, graph models have been used in complex datasets and recognized as useful by various researchers in chemical domain [7], computer vision technology [8], image and object retrieval [9], social network analysis [10] and machine learning [11]. In our work, we utilize the power of using graphs to model a complex sense of text data.

There are well-known subgraph discovery systems like FSG (Frequent Subgraph Discovery) [6], gSpan (graph-based Substructure pattern mining) [12], DSPM (Diagonally Subgraph Pattern Mining) [13], and SUBDUE [14]. These works let us to believe that the concept of construction of document-graphs and discovering frequent subgraphs to gain sense-based clustering of our work is feasible. All these systems deal with multiple aspects of efficient frequent subgraph mining. Most of them have been tested on real and artificial datasets of chemical compounds. None of them has been applied however, to mine the text data. In this paper, we discuss GDClust that performs frequent subgraph discovery from text repository with the aim of document clustering.

Agrawal et al. [18] proposed the Apriori approach for association rule mining. There had been extensive research works for generating association rules from frequent itemsets [19]–[20]. Besides, there are some transaction reduction approaches proposed by Agrawal et al. [18], and Han et al. [26]. We apply a variation of mining multilevel association rules [26] for the frequent sense discovery process and propose a novel Gaussian minimum support strategy for subgraph discovery in multiple levels of the taxonomy.

We introduce a sense based document clustering technique for the first time in the text-mining area. The work closest to our approach, we managed to find is a graph query refinement method proposed by Tomita et al. [15]. Their system depends on user interaction for the hierarchic organization of a text query. In contrast, we depend on a predefined ontology [1], for automated retrieval of frequent subgraphs from text documents. GDClust offers a fully automated system that utilizes Apriori-based subgraph discovery technique to harness the capability of sense-based document clustering.

3. System overview

This section portrays the techniques used for sense discovery and document clustering in GDClust.

3.1. Document-graph construction algorithm

GDClust utilizes BOW Toolkit [16] and WordNet 2.1 taxonomy to convert a document to its corresponding document-graph (Table 1). We utilized the WordNet's noun taxonomy, which provides a *hypernymy-hyponymy* relation between concepts and allows constructing a Concept Tree with up to 18 levels of abstractions. A *concept* is a set of synonymous words named *synset*. All nouns in WordNet are merged to a single topmost synset (i.e.,*{entity}*).

Table 1. Algorithm for construction of document-graphs.

(1) For each document D_i, construct a document-graph G_i, where $1<i<n$, and n is the total number of documents {
(2) For each keyword, k_j where $1<j<m$ and m is the number of keywords in document D_i {
(3) Traverse WordNet taxonomy up to the topmost level. During the traversal, consider each synset as a vertex. E is considered as a directed edge between two vertices V_1 and V_2, iff V_2 is the hypernym of V_1.
(4) E is labeled by $V_1::V_2$. If there is any repeated vertex or edge that was detected earlier for another keyword k_t $(t \neq j)$ of the same document, D_i, do not add the repeated vertices and edges to G_i, otherwise, add vertices and edges to G_i.
(5) } *// End of "For each keyword"*
(6) } *// End of "For each document"*

Our document-graph construction algorithm selects informative keywords from a document and retrieves corresponding synsets from WordNet. Then, it traverses up to the topmost level of abstraction to discover all related abstract terms and their relations. The graph of the links between keywords' synsets of each document and their abstracts compose the individual document-graph.

3.2. Utilizing Apriori paradigm for frequent sense discovery

GDClust uses frequent subgraphs as representation of common senses among the document-graphs. Two document-graphs, containing some common frequent subgraphs, do not have to have common keywords. Our system not only looks at the original keywords, but also looks at the origin of the keywords and their neighboring (i.e., abstract) synsets. Two different words, leading to the same hypernym, are going to generate two highly similar subgraphs, reflecting a common sense.

We use an Apriori paradigm, designed originally for finding frequent itemsets in market basket datasets

Table 2. Apriori algorithm for discovering frequent subgraphs.

D, a database of document-graphs.
min_sup, the minimum support count threshold
Output: *L*, frequent subgraphs in *D*.
Method:
(1) $L1$= find_frequent_1-edge_subgraphs(D);
(2) for ($k=2$; $L_{k-1}\neq\Phi$; k++){
(3) C_k=apriori_gen(L_{k-1});
(4) for each document-graph $g \in D$ {
(5) C_g=subset(C_k, g);
(6) for each candidate $c \in C_k$
(7) c.count++;
(8) }
(9) L_k={ $c \in C_k \mid c.count \geq min\_sup$}
(10) }
(11) Return $L=\cup_k L_k$

[18], to mine the frequent subgraphs from the document-graphs. In our work, subgraphs correspond to items in traditional frequent itemset discovery. The algorithm is portrayed in Table 2. The *find_frequent_1-edge_subgraphs* procedure utilizes the *dynamic minimum support strategy* (section 3.2.1) to select *1*-edge subgraphs from the document-graphs. The *apriori_gen* procedure in the algorithm performs joining and pruning of graphs. In the join operation, the list of subgraphs L_{k-1}, is joined with another L_{k-1} to generate potential candidates for the next Apriori's iteration. A *k*-edge candidate subgraph is generated by combining two *(k-1)*-edge subgraphs of L_{k-1}. The procedure removes candidate subgraphs that contain a subgraph which is not frequent. Details are described in section 3.2.2.

For improving the efficiency of Apriori algorithm, we used hash-based technique [21]. Besides, document-graphs are pruned with the observation that a document-graph which does not contain any frequent *k*-edge subgraph cannot contain any frequent *(k+1)*-edge subgraph. Therefore such a document-graph can be removed for further consideration.

3.2.1. Ontologically-constrained generation of subgraph candidates with dynamic minimum support. We use this approach to limit number of candidate subgraphs with extremely abstract and very specific meanings. Since the WordNet's ontology merges to a single term, the topmost level of abstraction is a common vertex for all the generated document-graphs, yielding subgraphs that involve the vertex with topmost level of abstraction to be less informative for clustering. Moreover, terms near to the lowest level of abstraction are less important due to their rare appearance in the document-graphs. As a result, terms appearing within the intermediate levels of the taxonomy are more representative clusters' labels than subgraphs containing terms at higher and lower levels.

We use a novel *dynamic minimum support imposed Apriori paradigm* to discover the frequent senses from document-graphs. This approach was motivated by the work on mining multilevel association rules [26]. The significant difference of our approach is that the technique operates neither with uniform minimum support, nor with linearly reduced support. Rather, it imposes the minimum support to the subgraphs in Gaussian normalization fashion, assigning different minimum support thresholds based on the term's abstraction level. To do this assignment in shorter time, instead of using WordNet, we use a *master document-graph*. A *master document-graph* is a sum of all our document-graphs, intersected with the WordNet taxonomy. An edge of the *master document-graph* is ranked according to the levels in WordNet taxonomy (currently, 18 abstraction levels). At the same time, the edges of the *master document-graph* do not have to cover all these 18 levels. Therefore, maximum abstraction levels in the *master document-graph* is bounded by $l_{max} \leq 18$.

For the preliminary investigations, we have chosen Gaussian normalization strategy. The Gaussian function possesses the shape matching our criteria of requiring smaller minimum support for the terms located at the intermediate levels, and assigning higher minimum support thresholds to the terms located at the lower and higher levels of the *master document-graph*. The approach imposes importance to the mid-levels of the taxonomy formed by master document-graph, with the assumption based on an observation that the generated document-graphs would contain a lot of common, but uninteresting, subgraphs at the topmost level, and distinct, but not frequent, subgraphs at the bottom levels. The first would generate large clusters with low inter-cluster similarity, and the second would generate huge number of very small clusters.

The Gaussian function can be defined as:

$$f(x) = Ae^{-(x-b)^2 \div 2c^2} \quad (1)$$

where A is the height of the Gaussian peak, b is the position of the center of the peak and c is defined as:

$$c = \frac{w}{2\sqrt{2\ln(2)}} \quad (2)$$

where w is the width of the curve at $A/2$. In our case, $b = l_{max} / 2$. We apply this behavior to model the minimum support of mining multilevel senses from WordNet taxonomy. This is illustrated in Figure 1. The hierarchy drawn in the figure indicates our *master document-graph*. The Gaussian graph indicates that minimum support is the largest at the highest and

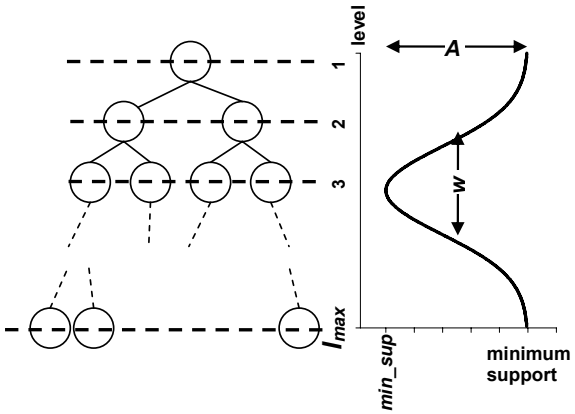

Figure 1. Gaussian minimum support strategy for multilevel mining.

lowest levels (i.e., level *1* and level l_{max}). The model generates our pre-defined minimum support, *min_sup* only at the mid level of the taxonomy and applies gradual increment of minimum support at higher and lower levels. One can shift the *min_sup* value to other levels by changing *b* of equation (1). Moreover, more subgraphs can be pruned from the candidate list by reducing *w* to make the curve narrower.

3.2.2. Candidate generation mechanism. The document-graph construction algorithm ensures that a document-graph would not contain more than one edge between two vertices. Additionally, the overall sense discovery concept ensures that a subgraph does not appear more than once in a document graph, unlike chemical compounds [7]. In our case, all the edges and vertices of a document-graph are labeled. We generate a *(k+1)*-edge candidate subgraph by combining two *k*-edge subgraphs where these two *k*-edge subgraphs have a common *core subgraph* [6] of *(k-1)*-edges. In GDClust, each *k-edge subgraph object* is composed of a connected edge-list and a list of possible edges that generated this *k*-edge subgraph from a *(k-1)*-edge subgraph.

3.3. Clustering text documents

GDClust uses Hierarchical Agglomerative Clustering (HAC) [22] to group documents together. We construct a dissimilarity matrix for every pair of document-graphs. Dissimilarity between a pair of document-graphs G_1 and G_2 is measured using the formula: *d=1.0-similarity*, where similarity is [23]:

$$sim(G_1, G_2) = \frac{count(SG(G_1) \cap SG(G_2))}{count(SG(G_1) \cup SG(G_2))} \quad (3)$$

where $SG(G_1)$ and $SG(G_2)$ are the sets of frequent subgraphs that exist in document-graph G_1 and G_2 respectively. The dividend of equation (3) indicates the number of common frequent subgraphs between document-graphs G_1 and G_2, and the divisor indicates the total number of unique frequent subgraphs in G_1 and G_2.

To get the accurate number of clusters we evaluated results of HAC using silhouette coefficient [24]-[25].

4. Performance of GDClust

GDClust detected 1458 unique edges in the generated 1000 document-graphs. The algorithm discovered largest frequent subgraph with 11 edges. Figure 2 shows an estimation of execution time of the Apriori algorithm for sense discovery. The gray line shows cumulative discovery time and the black line indicates individual *k*-edge subgraph discovery time. For this simulation, *min_sup* (of Figure 1) is set to 3% (this allows the dynamic minimum supports to be in the range [3, 100]) and the *c* value of equation (1) is derived with *w=(50% of A)* in equation (2). To show the impact of multilevel Gaussian minimum support, number of selected *1*-edge subgraphs from the candidate list of 1458 edges with different *min_sup* is portrayed in Table 3. Number of selected *1*-edge subgraphs becomes static below certain *min_sup*. In this simulation, number of selected *1*-edge subgraphs is always *174* for *min_sup<=4*. For other *min_sup* values, a lower *min_sup* in the Gaussian minimum support function would produce higher number of *1*-edge subgraphs at the very first level of the Apriori strategy of GDClust. This indicates that reduction of *min_sup* value may not result in further inclusion of *1*-edge subgraphs. In that case, if necessary, *w* should be increased to include additional *1*-edges subgraphs.

More and more edges can be pruned even with fixed *min_sup*, but varying *w* of the multilevel Gaussian minimum support curve. A narrower Gaussian curve with smaller *w* would result in smaller number of subgraphs, whereas a broader Gaussian curve with larger *w* will generate more *1*-edge subgraphs. This behavior is reflected in Table 4. It

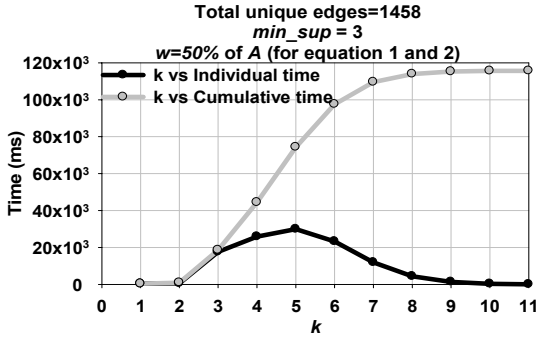

Figure 2. *k*-edge subgraph discovery time with 1000 document-graphs.

Table 3. Number of *1*-edge subgraphs selected from 1458 edges with different *min_sup* of Gaussian minimum support approach (*w* is set to 50% of *A* of equation 1).

min_sup(%)	*N(1)*
10	97
9	121
8	130
7	151
6	159
5	166
4	174
3	174
2	174
1	174

N(1) indicates number of selected *1*-edge subgraphs.

Table 4. Number of *1*-edge subgraphs selected from 1458 edges with different *w* of Gaussian minimum support approach, and fixed *min_sup=5%*.

w = % of A. (from equation 1)	*N(1)*
10	2
20	27
30	120
40	234
50	275
60	289
70	309
80	341
90	381
100	425

N(1) indicates number of selected *1*-edge subgraphs.

shows that with a fixed *min_sup*, number of selected *1*-edge subgraphs increase with increasing values of *w*.

One can also shift the center of the peak by controlling the *b* value of equation (1), making the curve skewed in any direction. In our preliminary experiments, we kept the curve symmetric, with the assumption that most of the senses of the document-graphs are represented by the midlevel of the taxonomy and similar document-graphs have the tendency to start overlapping at midlevel.

4.1. Document-graph clustering

The discovered subgraphs of the previous section are used for the clustering purpose. The 1000 documents were chosen from 10 different groups of *20-newsGroup Dataset* [27]. Figure 3 shows average silhouette coefficients at different number of clusters obtained by hierarchical agglomerative clustering. As the tendency of the curve is downward after certain number of clusters, we displayed silhouettes up to 25 clusters. The graph shows that the maximum average

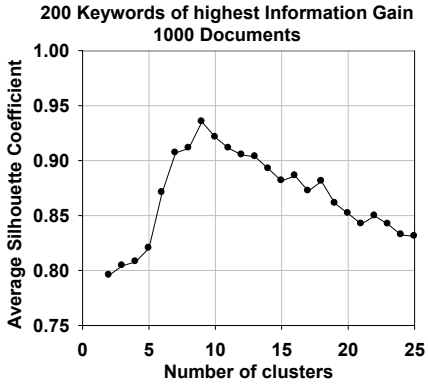

Figure 3. Calculated average silhouette coefficient at different number of clusters.

silhouette coefficient is found when the number of clusters is 9. This is close to the number of groups in the input documents. It is to be noted that, all the average silhouettes displayed in Figure 3 are greater than 0.75 which is particularly good. This means that average silhouette coefficient remains high in the neighborhood of natural number of clusters (i.e. 10) and gradually falls downward in such a plot. Our results show, that the quality of clustering reaches its maximum for 9 clusters. This demonstrates a close match of cluster numbers with the number of predefined groups of the dataset.

5. Conclusion

GDClust presents a new technique for clustering text documents based on co-occurrence of frequent senses in the documents. The developed novel approach offers an interesting, sense-based alternative to the commonly used bag-of-tokens technique for clustering text documents. Unlike traditional systems, GDClust harnesses its clustering capability from the frequent senses discovered in the documents. It utilizes graph-based mining technology to discover frequent senses. GDClust is an automated system and requires minimal user interaction for its operations.

In the close future, we want to look carefully at the concept of the inexact matching of subgraphs [14], as we believe it can be used effectively during our clustering process. We expect that the inexact matching would allow us to select only larger subgraphs generated by the Apriori approach, which could farther decrease computational costs involved in the phase of frequent subgraph candidate analysis. We are also interested in extending our multilevel dynamic minimum support approach. For our preliminary investigations, we used Gaussian minimum support strategy for the taxonomy. Finding the most

appropriate minimum support model remains as one of important aspects of our future work.

References

[1] Miller G.A. and Charles W.G., "Contextual Correlates of Semantic Similarity", *Language and Cognitive Processes*, vol. 6(1), 1991, pp. 1–28.

[2] F. Sebastiani, "Machine learning in automated text categorization", *ACM Comp. Surveys*, vol. 34(1), 2002, pp. 1–47.

[3] C. D. Manning and H. Schutze, "Foundations of Natural Language Processing", *MIT Press*, 1999.

[4] C. Cleverdon, "Optimizing convenient online access to bibliographic databases", *Inf. Survey and Use*, vol. 4(1), 1984, pp. 37–47.

[5] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis", *Journal of the Society for Inf. Science*, vol. 41(6), 1990, pp. 391–407.

[6] M. Kuramochi and G. Karypis, "An efficient algorithm for discovering frequent subgraphs", *IEEE Trans. on KDE*, vol. 16(9), 2004, pp.1038–1051.

[7] R. N. Chittimoori, L. B. Holder, and D. J. Cook, "Applying the SUBDUE substructure discovery system to the chemical toxicity domain", *Proc. of the 12th Intl. FLAIRS Conf.*, 1999, pp. 90–94.

[8] D. A. L. Piriyakumar, and P. Levi, "An efficient A* based algorithm for optimal graph matching applied to computer vision", *GRWSIA-98*, Munich, 1998.

[9] D. Dupplaw and P. H. Lewis, "Content-based image retrieval with scale-spaced object trees", *Proc. of SPIE: Storage and Retrieval for Media Databases*, vol. 3972, 2000, pp. 253–261.

[10] M. E. J. Newman, "The structure and function of complex networks", *SIAM Review*, vol. 45(2), 2003, pp. 167–256.

[11] K. Yoshida and H. Motoda, "CLIP: Concept learning from inference patterns", *Artificial Intelligence*, vol. 75(1), 1995, pp. 63–92.

[12] X. Yan and J. Han, "gSpan: graph-based substructure pattern mining", *Proc. of IEEE ICDM*, 2002, pp. 721–724.

[13] C. Moti and G. Ehud, "Diagonally Subgraphs Pattern Mining", *Proc. of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2004, pp. 51–58.

[14] N. Ketkar, L. Holder, D. Cook, R. Shah and J. Coble, "Subdue: Compression-based Frequent Pattern Discovery in Graph Data", *Proc. of the ACM KDD Workshop on Open-Source Data Mining,* August 2005, pp. 71–76.

[15] J. Tomita, and G. Kikui, "Interactive Web search by graphical query refinement", *Proc. of the 10th Intl. WWW Conf,* 2001, pp. 190–191.

[16] A. McCallum, "Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering", *http://www.cs.cmu.edu/~mccallum/bow/*

[17] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, Morgan Kaufmann Pub., 2006, pp. 234–242.

[18] R. Agrawal, and R. Srikant, "Fast Algorithms for Mining Association Rules", *Proc. of Intl. Conf. on VLDB*, Santiago, Chile, 1994, pp. 487–499.

[19] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger, "The Quest Data Mining System", *Proc. of KDD'96*, USA, 1996, pp. 244–249.

[20] H. Mannila, H. Toivonen, and I. Verkamo, "Efficient Algorithms for Discovering Association Rules", *Proc. of AAAI Workshop on Knowledge Discovery in Databases*, Seattle, Washington, USA, 1994, pp. 181–192.

[21] J. S. Park, M. S. Chen, P. S. Yu, "An effective hash-based algorithm for mining association rules", *Proc. of the ACM SIGMOD'95*, San Jose, CA, USA, 1995, pp. 175–186.

[22] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", *Proc. of ACM SIGMOD'96*, Quebec, Canada, 1996, pp. 103–114.

[23] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang, "Finding interesting associations without support pruning", *IEEE Trans. on KDE*, vol. 13(1), 2001, pp. 64–78.

[24] F. Lin, C. M. Hsueh, "Knowledge map creation and maintenance for virtual communities of practice", *Intl. Journal of Inf. Processing and Management, ACM,* vol. 42(2), 2006, pp. 551–568.

[25] P. N. Tan, M. Steinbachm, V. Kumar, *Introduction to data mining*, Addison-Wesley, 2005, pp. 539–547.

[26] J. Han, and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases", *Proc. of the 21th Intl. Conf. on VLDB*, Switzerland, 1995. pp. 420–431.

[27] "20 News Groups Dataset", *http://people.csail.mit.edu/jrennie/20Newsgroups/*