SCUBA DIVER: SUBSPACE CLUSTERING OF WEB SEARCH RESULTS

Fatih Gelgi, Srinivas Vadrevu and Hasan Davulcu
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, U.S.A.

Keywords: Web search, subspace clustering, guided search.

Abstract: Current search engines present their search results as a ranked list of Web pages. However, as the number of pages on the Web increases exponentially, so does the number of search results for any given query. We present a novel subspace clustering based algorithm to organize keyword search results by simultaneously clustering and identifying distinguishing terms for each cluster. Our system, named Scuba Diver, enables users to better interpret the coverage of millions of search results and to refine their search queries through a keyword guided interface. We present experimental results illustrating the effectiveness of our algorithm by measuring purity, entropy and F-measure of generated clusters based on Open Directory Project (ODP).

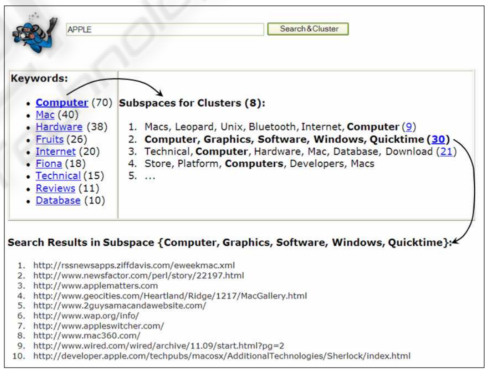
1 INTRODUCTION

Current search engines present their search results as an ordered list of Web pages ranked in terms of their relevance to the user's keyword query. As the number of pages on the Web is increasing exponentially, the number of search results for any given query is also increasing at the same rate. For example there are 544 million results returned by Google search engine[1] for the keyword query 'apple'. Therefore even though there are millions of potentially relevant documents for any given user's query and interest, only top few ranking documents in the first few pages of a search engine results can be found and explored.

An alternate way to browse the search results returned by a search engine is to organize them into related clusters to guide the user in her search. When the search results are clustered and organized in terms of their distinguishing features, it may become easier for users (i) to interpret the coverage of millions of search results and (ii) to refine their search queries.

Traditionally, clustering has been applied to Web pages in the context of document clustering such as Clusty[2] and Mooter[3]. They have fair performance and cluster descriptions are not identified in these systems. (Crescenzi et al., 2005) utilizes the structure of HTML pages to cluster them. They utilize layout and presentation properties of blocks of similarly presented links in identifying clusters of Web pages. Their clustering algorithm is based on the observation that similar pages share the same layout and presentation structures.

In this paper, we present a way to organize the search results by clustering them into groups of Web pages that belong to the same category. Subspace clustering (Parsons et al., 2004) is an extension of traditional clustering that seeks to find clusters in different subspaces of feature space within a data set. We choose subspace clustering since it provides the distinguishing features that make up each cluster, in



Keywords:

- Computer (70)
- Mac (40)
- Hardware (38)
- Fruits (26)
- Internet (20)
- Fiona (18)
- Technical (15)
- Reviews (11)
- Database (10)

Subspaces for Clusters (8):

1. Macs, Leopard, Unix, Bluetooth, Internet, **Computer** (9)
2. **Computer, Graphics, Software, Windows, Quicktime (30)**
3. Technical, **Computer**, Hardware, Mac, Database, Download (21)
4. Store, Platform, **Computers**, Developers, Macs
5. ...

Search Results in Subspace {Computer, Graphics, Software, Windows, Quicktime}:

1. http://rssnewsapps.ziffdavis.com/eweekmac.xml
2. http://www.newsfactor.com/perl/story/22197.html
3. http://www.applematters.com
4. http://www.geocities.com/Heartland/Ridge/1217/MacGallery.html
5. http://www.2guysamacandawebsite.com/
6. http://www.wap.org/info/
7. http://www.appleswitcher.com/
8. http://www.mac360.com/
9. http://www.wired.com/wired/archive/11.09/start.html?pg=2
10. http://developer.apple.com/techpubs/macosx/AdditionalTechnologies/Sherlock/index.html

Figure 1: Clustered results for keyword query 'apple'.

[1]http://www.google.com
[2]http://www.clusty.com
[3]http://www.mooter.com

Gelgi F., Vadrevu S. and Davulcu H. (2007).
SCUBA DIVER: SUBSPACE CLUSTERING OF WEB SEARCH RESULTS.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 334-339
DOI: 10.5220/0001288503340339
Copyright © SciTePress

addition to clustering the documents. These features can be used as additional guidance information in organizing the search results.

Figure 1 illustrates the keyword query ‘apple’ and the clusters of Web pages along with their subspace labels. We refine the clusters obtained by subspace clustering by merging similar clusters and partitioning them so that each Web page belongs to only one cluster. The figure also shows the related keywords comprising all labels, ordered in terms of their occurrence frequency. These keywords provide an additional way to filter and navigate through the search results. Figure 1 shows top four clusters along and their subspaces for the keyword query ‘apple’ after filtering the subspaces with the keyword ‘computer’ from the related keywords area. The individual results of any matching cluster can be displayed by clicking on the corresponding subspace.

Related work includes clustering techniques have been used previously in the area of information retrieval to organize the search results. (Zeng et al., 2004) models the search result clustering problem as ranking salient phrases extracted from the search results. The phrases (cluster names) are ranked by using a regression model with features extracted from the title and snippets of search results. (Beil et al., 2002) works on frequent-term based text clustering. One of their data set is collected from Yahoo! subject hierarchy classified into 20 categories with 1560 Web pages. Their best F-measure is 43% for the Web data which is substantially low compared with their experiments on text documents. Other researchers (Leouski and Croft, 1996; Leuski and Allan, 2000) utilize traditional machine learning based clustering algorithms to cluster the search results and choose descriptive cluster names from various frequency based features of the documents.

The rest of the paper is organized as follows. Problem description and formalization is given in Section 2. Details of our algorithm is presented in Section 3 and the experiments is presented in Section 4. We conclude and give the future work in Section 5.

2 PROBLEM DESCRIPTION

In large datasets such as Web, clustering algorithms have to deal with two challenges as indicated in (Agarwal et al., 2006): (i) *sparsity* – the data is sparse if most of the entries of vectors are zero, (ii) *high-dimensional data* – the number of features in dataset is large. In the Web domain, we have both problems since the features are terms in the domain, and a Web page is related with a very small fragment of this term set. Classical clustering techniques use feature transformation and feature selection to overcome the described problems above. On the other hand, newer techniques such as subspace clustering (Parsons et al., 2004) localizes its search and is able to identify clusters that exist in multiple, possibly overlapping subspaces instead of examining the dataset as a whole. Subspace clustering also uncovers the relevant features of the clusters and keeps the original ones that make the interpretation easier.

Context can be defined as the subset of terms in the domain. Hence one can consider context as the subspace of the Web pages in the same cluster. In our formalization, $\mathcal{F}$ refers to the feature set selected from all the terms in the collection of Web pages.

Definition 1 (Subspace Cluster) *A subspace cluster is a cluster $C = \langle f, w \rangle$ composed of a pair of sets where $f \subseteq \mathcal{F}$ is the feature set (or the subspace) of the cluster and $w \subseteq \mathcal{W}$ is the set of Web pages belonging to the cluster. We denote $f(C)$ and $w(C)$ as the features and Web pages of a cluster respectively.*

Formal definition of our problem can be stated as:

Problem statement 1 *Given a set of Web pages $\mathcal{W} = \{W_1, W_2, \dots W_n\}$ resulting from an ambiguous keyword search $\mathcal{K}$, find the clusters of Web pages together with their descriptive terms. A Web page W_i is defined as the set of the its terms.*

This problem can be reduced a subspace clustering problem as follows:

Problem statement 2 *Given a binary $n \times m$ matrix M, where rows represent n Web pages ($\mathcal{W} = \{D_1, D_2, \dots D_n\}$) and columns represent m terms ($\mathcal{F} = \{t_1, t_2, \dots t_m\}$), find subspace clusters of Web pages, $W \subseteq \mathcal{W}$ defined in the subspace of terms, $T \subseteq \mathcal{F}$.*

Here $D_i = \langle d_{i1}, d_{i2}, \dots d_{im} \rangle$ represents the binary vector of the Web page W_i where $d_{ij} = 1$ when $t_j \in W_i$. In the matrix M, d_{ij} correspond to the entry in the i^{th} row and j^{th} column, that is $M_{ij} = d_{ij}$.

3 SCUBA DIVER ALGORITHM

Our subspace clustering algorithm given in Algorithm 1 is composed of three steps: *feature selection*, *subspace clustering*, and *merging and partitioning*.

3.1 Feature Selection

This step is given the first line in $ScubaDiver$ function ($ScubaDiver$ refers to the main function of the algorithm) in Algorithm 1. $SelectFeatures$ function

Algorithm 1 Clustering of Web Search Results

```
$ScubaDiver(\mathcal{W}, \mathcal{K})$
Input: $\mathcal{K}$, the search keyword; $\mathcal{W}$, a set Web pages resulting from keyword
    search $\mathcal{K}$
Output: $\mathcal{C}$, set of subspace clusters of the Web pages $\mathcal{W}$
1: $\mathcal{F} \leftarrow SelectFeatures(\bigcup_{W \in \mathcal{W}} W)$
2: for $\forall W_i \in \mathcal{W}$
3:   for $\forall t_j \in \mathcal{F}$
4:     if $t_j \in W_i$ then $M_{ij} \leftarrow 1$
5:     else $M_{ij} \leftarrow 0$
6: $\mathcal{C} \leftarrow SCuBA(M, \alpha, \beta)$
7: $\mathcal{C} \leftarrow Merge(\mathcal{C}, \delta, \xi)$
8: $\mathcal{C} \leftarrow Partition(\mathcal{W}, \mathcal{C})$
9: return $\mathcal{C}$
End of ScubaDiver

$Merge(\mathcal{C}, \delta, \xi)$
Input: $\mathcal{C}$, subspace clusters; $\delta$, threshold for the common features; $\xi$,
    threshold for the common Web pages;
Output: $\mathcal{C}$, set of merged subspace clusters of the Web pages $\mathcal{W}$
1: for $\forall C_i \in \mathcal{C}$
2:   for $\forall C_j \in (\mathcal{C} - \{C_i\})$
3:     if $(Jaccard(f(C_i), f(C_j)) > \delta) \wedge$
4:       $(Jaccard(w(C_i), w(C_j)) > \xi)$ then
5:       $C' \leftarrow \langle f(C_i) \cup f(C_j), w(C_i) \cup w(C_j) \rangle$
6:       $\mathcal{C} \leftarrow \mathcal{C} - \{C_i, C_j\} \cup \{C'\}$
7: return $\mathcal{C}$
End of Merge

$Partition(\mathcal{W}, \mathcal{C})$
Input: $\mathcal{W}$, set of Web pages; $\mathcal{C}$, subspace clusters;
Output: $\mathcal{C}$, set of partitioned subspace clusters of the Web pages $\mathcal{W}$
1: for $\forall W_i \in \mathcal{W}$
2:   $closest \leftarrow \arg\max_{C_j \in \mathcal{C}}\{similarity(v(C_j), v(W_i))\}$
3:   $w(closest) \leftarrow w(closest) \cup \{W_i\}$
4:   $\mathcal{C}' \leftarrow \{X | X \in (\mathcal{C} - \{closest\}) \wedge W_i \in w(X)\}$
5:   for $\forall C_j \in \mathcal{C}'$
6:     $w(C_j) \leftarrow w(C_j) - \{W_i\}$
7: return $\mathcal{C}$
End of Partition
```

selects features for clustering among all the terms appeared in the data.

In $SelectFeatures$ function, data has first been preprocessed then features have been selected by three common frequency based methods in Information Retrieval (IR). *Term frequency* of a term t is, $TF(t) = \frac{n_t}{\sum_{k \in \mathcal{F}} n_k}$ where n_t is the number of occurrences of t in the domain. *Document frequency* of t is, $DF(t) = \frac{|D(t)|}{n}$ where $D(t)$ is the set of Web pages t appears. A common variation of *Term frequency / inverse document frequency* of t is, $TF/IDF(t) = TF(t). \lg \frac{1}{DF(t)}$. Readers should note that deep analysis of feature selection is out of scope of this paper.

Based on the three measures above the features are selected as top-m ranked terms among all the terms in the data excluding the search keyword $\mathcal{K}$. For example, top-10 TF/IDF terms obtained from the Web pages with the search keyword *apple* in our experiments are 'mac', 'recipe', 'ipod', 'search', 'software', 'computer', 'macintosh', 'list', 'store' and 'pie'.

3.2 Subspace Clustering

Due to its efficiency and effectiveness on binary data, we use the subspace clustering algorithm SCuBA described in (Agarwal et al., 2006) which is part of an article recommendation system for researchers. In Algorithm 1, lines from 2 to 5 prepare the binary matrix M for SCuBA and line 6 generates the subspace clusters. SCuBA works in two steps: (i) it compacts the data by reducing the $n \times m$ matrix to n rows each of which is a list of the size of its corresponding Web page, and (ii) it searches for the subspaces by comparing each row to the successive rows to find the intersecting rows and their subsets of columns.

SCuBA produces many small subspace clusters with many overlapping Web pages. Example of two subspace clusters generated from *apple* search have $\{macintosh, computer, imac, mac\}$ and $\{macintosh, check, mac\}$ as their feature sets containing 4 Web pages each, 2 of which are common.

3.3 Merging and Partitioning

Final clusters are determined after merging step as the post-processing on subspace clusters in lines 7 and 8 in Algorithm 1. In the Web data subspace clusters obtained from SCuBA are in the form of sub-matrices which are not significant in number and size. Clusters needs to be relaxed to have irregular shapes and to cover more data as opposed to strict sub-matrices.

A common scenario is the overlapping subspace clusters which are the subsets of the same actual cluster and the context. Hence it is reasonable to merge the subspace clusters C_i and C_j if they share certain amount of terms and Web pages. Jaccard similarity coefficient is one of the common measure for comparing the similarity and diversity of the sets. The Jaccard coefficient is defined as the size of the intersection divided by the size of the union of the sets, $Jaccard(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$.

In Algorithm 1, $Merge$ function merges the subspace clusters by adopting the Jaccard coefficient to measure the similarity of the set of features and Web pages of two clusters. If the similarity is more than the thresholds δ and ξ respectively, we merge the clusters. Formally, in lines 3-6 if $Jaccard(f(C_i), f(C_j)) > \delta$

Table 1: Selected search keywords and their corresponding categories in data sets.

Keyword	*# of categories*	*# of pages*	*Categories*
apple	6	648	Computers(463), Fruits(136), Locations(17), Music(21), Movies(6), Games(5)
gold	6	670	Shopping(471), Mining(151), Movies(28), Motors(11), Games(8), Sports(1)
jaguar	4	138	Cars(78), Video games(48), Animals(9), Music(3)
paper	10	1422	Shopping(626), Materials(380), Academic(113), Arts(107), Money(78), Games(39), Computers(27), Consultants(23), Environment(19), Movies(10)
saturn	4	71	Cars(22), Planets(21), Anime(19), Video games(9)

and $Jaccard(w(C_i), w(C_j)) > \xi$, the merged cluster C' becomes $C' = \langle f(C_i) \cup f(C_j), w(C_i) \cup w(C_j) \rangle$.

Following the same example given in Section 3.2, the two subspace clusters C_i and C_j is merged due to the high overlap in their features and Web pages. Their Jaccard coefficients for the features and Web pages are $Jaccard(f(C_i), f(C_j)) = 0.4$ and $Jaccard(w(C_i), w(C_j)) = 0.33$.

Note that subspace clustering is a soft clustering method that allows a Web page to be in multiple categories. This flexibility is more realistic for real world data. Besides, hard clustering is also possible. A simple idea is to assign each Web page to its closest subspace cluster and eliminate the duplicates from other clusters. In the $Partition$ function in Algorithm 1, line 2 calculates the closest cluster for each Web page W_i. Here $v(\cdot)$ returns the feature vector. Subspace cluster vector can be considered as the feature vector of that cluster, and cosine measure can be used for the similarity, $similarity(x, y) = \frac{x \bullet y}{|x||y|}$ where x and y are the vectors of the cluster and the Web page.

Cosine is determined as one of the best measures for Web page clustering, and also for binary and sparse data (Strehl et al., 2000). Lines 4-6 remove the duplicates from other clusters. All the remaining Web pages which doesn't belong to any cluster are put in one cluster called 'others'.

4 EXPERIMENTS

In our experiments, we identified some challenging keywords inspired from (Zeng et al., 2004) on *Open Directory Project*[4] (ODP).

[4]http://www.dmoz.org

We used the search keywords 'apple', 'gold', 'jaguar', 'paper' and 'saturn' for data collection. For each search keyword we prepared the data from the resulting Web pages as given in Table 1. In the preprocessing step, the common data types of values such as percentage, dates, numbers etc., stop words, and punctuation symbols are filtered using simple regular expressions to standardize the data.

In each keyword data, the features are selected as top-1000 terms which are ranked based on TF/IDF measure as mentioned in Section 3.1. We consider 1000 terms are sufficiently enough to capture the contextual information given in the Web pages. Discussions on feature selection are given in Section 4.2.

Subspace clusters are required to have at least 3 features and 3 Web pages for the thresholds α and β. For merging subspaces, the thresholds δ and ξ are determined as 0.2 and 0.1 respectively.

4.1 Evaluation Metrics

To evaluate the quality of our results, the clusters are compared with the actual categories given in ODP. We use the common evaluation metrics in clustering (Rosell et al., 2004) such as *precision*, *recall*, *F-measure*, *purity*, and *entropy*. Precision, $p_{ij} = \frac{n_{ij}}{n_i}$ and recall, $r_{ij} = \frac{n_{ij}}{n_j}$ compare each cluster i with each category j where n_{ij} is the number of Web pages appear in both the cluster i and the category j, n_i and n_j are the number of Web pages in the cluster i and in the category j respectively. F-measure, $F_{ij} = \frac{2p_{ij}r_{ij}}{p_{ij}+r_{ij}}$ is a common metric calculated similarly to the one in IR. The F-measure of a category j is $F_j = \max_i\{F_{ij}\}$ and similarly the overall F-measure is:

$$F = \sum_j \frac{n_j}{n} F_j. \quad (1)$$

Quality of each cluster can be calculated by purity and entropy. Purity measures how pure is the cluster i by $\rho_i = \max_j\{p_{ij}\}$. The purity of the entire clustering can be calculated by weighting each cluster proportional to its size as:

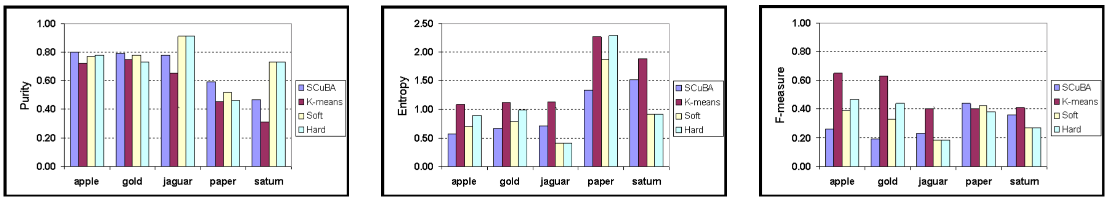
$$\rho = \sum_i \frac{n_i}{n} \rho_i \quad (2)$$

where n is the total number of Web pages.

The entropy of a cluster i is $E_i = -\sum_j p_{ij} \log p_{ij}$. Calculating the weighted average over all clusters gives the entire entropy of the clustering:

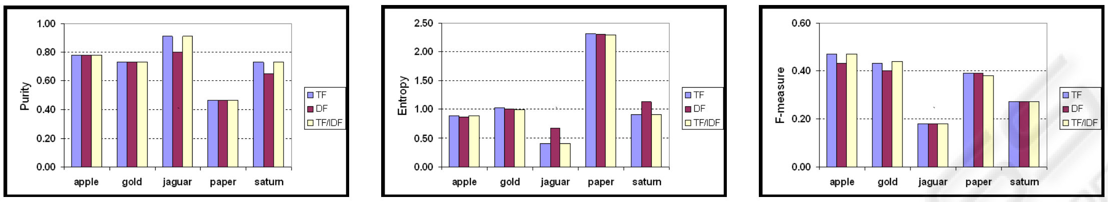
$$E = \sum_i \frac{n_i}{n} E_i. \quad (3)$$

Note that for soft clustering, n in Equations 2 and 3 has to be the sum of the sizes of the clusters since



(a) (b) (c)

Figure 2: Comparison of clustering methods.



(a) (b) (c)

Figure 3: Effect of the future selection methods.

a Web page might appear in more than one clusters. F-measure is related with the size of categories hence n remains the same in Equation 1.

4.2 ODP Results

First we use a wrapper which sends the given search keyword to `http://www.dmoz.org`, and collects the resulting categories and the Web pages belong to those categories. Collected pages are categorized by their main ODP categories. Next, all the text is extracted from the collected Web pages.

K-means clustering method is used as a baseline measure to demonstrate to quality of our method. K-means is one of the common clustering methods preferred for its speed and quality. In our experiments, even K is provided which is a great advantage for K-means over our method. For each keyword data, K-means has been executed 20 times and the results are the average of all runs. Purity, entropy and F-measure are deviated in the interval of ± 0.05.

Figure 2 and Table 2 presents the performance of the subspace clustering vs. the baseline method, K-means. *SCuBA* refers to the initial subspace clusters generated by SCuBA algorithm, *Soft* refers the soft clusters after merging the subspace clusters, and *Hard* refers to the final hard clusters after partitioning.

As shown in Figure 2(a) and (b), our method surpasses K-means in purity and entropy measures significantly although the number of clusters are provided to K-means. SCuBA generates small but pure subspace clusters. However, F-measure is low due to the excessive number of clusters, and there are too many redundant clusters with many duplicate pages as shown in Table 3. Although sensitive and aggressive merging in our method reduces the number of clusters substantially and increases the F-measure, there are still too many clusters and duplicates in soft clustering, and F-measure is not comparable with K-means. In addition, F-measure is especially low in *jaguar* and *saturn* data since they don't have rich subspaces that prevents to identify common subspace clusters to merge.

Features of some of the subspace clusters are presented in Table 4. One can notice the common terms such as 'contact', 'privacy', and 'see' in *paper* data that lead the worst quality among the other keywords. Merging due to the common terms degenerates the purity of the clusters as opposed to F-measure. Another reason for the purity to be low in *paper* data is its context ambiguity; the context of categories have more overlaps than the other keywords have. For instance, many Web pages in *Shopping*, *Materials*, *Arts*, and *Money* have content about the quality of papers.

Determining the number of clusters has been a challenging problem for clustering methods. As shown in Table 3, our algorithm generated more clusters than the actual for *apple* and *gold* keywords. Sometimes main clusters can not be generated due to the completely separate context in subcategories. For example, *hardware* and *software* subcategories are identified however they are not clustered together. Clusters are less in *paper* data due to the merging caused by common terms in the data.

Top-1000 terms are sufficient to include all the descriptive and discriminative terms in the collections. However, they also contain common and ambiguous terms which hinders clustering process such as 'e-mail','forum', and 'contact'.

TF, DF and TF/IDF measures have not affected

Table 2: Performance comparison of clustering methods. *P*, *E* and *F* refers to purity, entropy and F-measure respectively.

	SCuBA			Soft			K-means			Hard		
Keyword	P	E	F	P	E	F	P	E	F	P	E	F
apple	0.80	0.57	0.26	0.77	0.70	0.39	0.72	1.08	0.65	0.78	0.89	0.47
gold	0.79	0.67	0.19	0.78	0.78	0.33	0.75	1.11	0.63	0.73	0.99	0.44
jaguar	0.78	0.71	0.23	0.91	0.40	0.18	0.65	1.12	0.40	0.91	0.40	0.18
paper	0.59	1.34	0.44	0.52	1.87	0.42	0.45	2.27	0.40	0.46	2.29	0.38
saturn	0.47	1.52	0.36	0.73	0.91	0.27	0.31	1.89	0.41	0.73	0.91	0.27

Table 3: Comparison of actual clusters with clustering methods. *Clusters* and *Pages* refers to number of clusters and number of total Web pages in the clusters respectively.

	# of	*# of*	SCuBA		Soft		K-means	Hard
Keyword	*Cat.*	*pages*	Clusters	Pages	Clusters	Pages	Clusters	Clusters
apple	6	648	801	4402	181	2406	6	23
gold	6	670	578	3276	165	2077	6	27
jaguar	4	138	15	88	4	95	4	4
paper	10	1422	2054	12593	120	3516	10	6
saturn	4	71	12	86	3	66	4	3

Table 4: Sample features of one subspace cluster for each keyword.

Keyword	*Sample features*	*Related Category*
apple	macintosh, users, computer, product, check, imac, mac	Computers
gold	necklace, ring, pendant, earring bracelet, shipping, silver, jewelry	Shopping
jaguar	auto, cars, parts, support	Cars
paper	printing, contact, privacy, product, unique, see	Shopping
saturn	planet, sun, image, satellite	Planets

the performance of subspace clustering significantly as shown in Figure 3. Especially, TF and TF/IDF showed nearly the same performance. As opposed to the expectations, although the common terms are ranked in lower, it is surprising to find out TF/IDF is not able to eliminate them.

Consequently, our subspace clustering based algorithm preserves the quality of the clusters initially obtained from SCuBA while significantly reducing the number of clusters. Furthermore, it is better than a common state of art clustering algorithm although the number of clusters are provided.

5 CONCLUSION

We present a novel subspace clustering based algorithm to organize search results by simultaneously clustering and identifying their distinguishing terms. We present experimental results illustrating the effectiveness of our algorithm by measuring purity, entropy and F-measure of generated clusters based on Open Directory Project (ODP). As the future work, we will work on feature selection to eliminate common terms to increase the quality of the features. Currently, merging is a simple rule based method, whereas one can explore the use of sophisticated relationship analysis over clusters.

REFERENCES

Agarwal, N., Haque, E., Liu, H., and Parsons, L. (2006). A subspace clustering framework for research group collaboration. *International Journal of Information Technology and Web Engineering*, 1(1):35–38.

Beil, F., Ester, M., and Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of SIGKDD'02*, pages 436–442, New York, NY, USA. ACM Press.

Crescenzi, V., Merialdo, P., and Missier, P. (2005). Clustering web pages based on their structure. *Data and Knowledge Engineering*, 54:279–299.

Leouski, A. and Croft, W. B. (1996). An evaluation of techniques for clustering search results. Technical Report IR-76, University of Massachusetts, Amherst.

Leuski, A. and Allan, J. (2000). Improving interactive retrieval by combining ranked lists and clustering. In *Proceedings of RIAO'2000*, pages 665–681.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: A review. *SIGKDD Explorations*, 6(1):90.

Rosell, M., Kann, V., and Litton, J.-E. (2004). Comparing comparisons: Document clustering evaluation using two manual classifications. In *Proceedings of ICON'04*.

Strehl, A., Ghosh, J., and Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Proceedings of AAAI'00*, pages 58–64. AAAI.

Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., and Ma, J. (2004). Learning to cluster web search results. In *Proceedings of ACM SIGIR'04*, pages 210–217, New York, NY, USA. ACM Press.