Learning to Cluster Web Search Results

Hua-Jun Zeng[1] Qi-Cai He[2] Zheng Chen[1] Wei-Ying Ma[1] Jinwen Ma[2]

[1]Microsoft Research, Asia
49 Zhichun Road
Beijing 100080, P.R.China
{hjzeng, zhengc, wyma}@microsoft.com

[2]LMAM, Department of Information Science,
School of Mathematical Sciences, Peking University,
Beijing 100871, P. R. China
heqicai@pku.edu.cn,
jwma@math.pku.edu.cn

ABSTRACT

Organizing Web search results into clusters facilitates users' quick browsing through search results. Traditional clustering techniques are inadequate since they don't generate clusters with highly readable names. In this paper, we reformalize the clustering problem as a salient phrase ranking problem. Given a query and the ranked list of documents (typically a list of titles and snippets) returned by a certain Web search engine, our method first extracts and ranks salient phrases as candidate cluster names, based on a regression model learned from human labeled training data. The documents are assigned to relevant salient phrases to form candidate clusters, and the final clusters are generated by merging these candidate clusters. Experimental results verify our method's feasibility and effectiveness.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search process, Clustering, Selection process*; G.3 [**Probability and Statistics**]: Correlation and Regression Analysis

General Terms

Algorithms, Experimentation

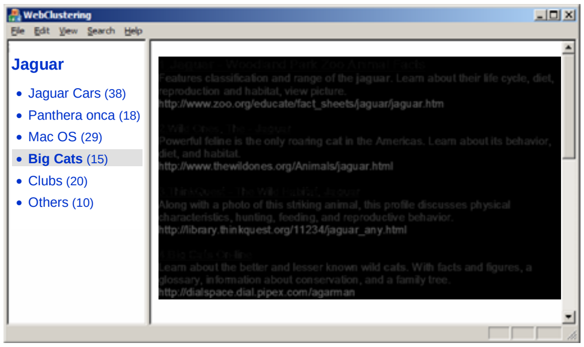
Keywords

Search result organization, document clustering, regression analysis

1. INTRODUCTION

Existing search engines such as Google [4], Yahoo [15] and MSN [12] often return a long list of search results, ranked by their relevancies to the given query. Web users have to go through the list and examine the titles and (short) snippets sequentially to identify their required results. This is a time consuming task when multiple sub-topics of the given query are mixed together. For example, when a user submits query "jaguar" into Google and wants to get search results related to "big cats", s/he should go to the 10th, 11th, 32nd and 71st results.

A possible solution to this problem is to (online) cluster search results into different groups, and to enable users to identify their required group at a glance. Hearst and Pedersen [6] showed that relevant documents tend to be more similar to each other, thus the clustering of similar search results helps users find relevant results. In the above example for query "jaguar", if there is a group named "big cats", the four relevant results will be ranked high in the corresponding list (as shown in Figure 1). Several previous works [16][17][6][11][10] are conducted to develop effective and efficient clustering technology for search result organization. In addition, Vivisimo [14] is a real demonstration of this technique.



Jaguar

- Jaguar Cars (38)
- Panthera onca (18)
- Mac OS (29)
- **Big Cats** (15)
- Clubs (20)
- Others (10)

1.Jaguar - Woodland Park Zoo Animal Facts
Features classification and range of the jaguar. Learn about their life cycle, diet, reproduction and habitat, view picture.
http://www.zoo.org/educate/fact_sheets/jaguar/jaguar.htm

2.Wild Ones, The - Jaguar
Powerful feline is the only roaring cat in the Americas. Learn about its behavior, diet, and habitat.
http://www.thewildones.org/Animals/jaguar.html

3.ThinkQuest - The Wild Habitat, Jaguar
Along with a photo of this striking animal, this profile discusses physical characteristics, hunting, feeding, and reproductive behavior.
http://library.thinkquest.org/11234/jaguar_any.html

4.Big Cats On-line
Learn about the better and lesser known wild cats. With facts and figures, a glossary, information about conservation, and a family tree.
http://dialspace.dial.pipex.com/agarman

Figure 1. An Example of Search Result Clustering

Clustering methods don't require pre-defined categories as in classification methods. Thus, they are more adaptive for various queries. Nevertheless, clustering methods are more challenging than classification methods because they are conducted in a fully unsupervised way. Moreover, most traditional clustering algorithms cannot be directly used for search result clustering, because of some practical issues. Zamir and Etzioni [16][17] gave a good analysis on these issues. For example, the algorithm should take the document snippets instead of the whole documents as input, since the downloading of original documents is time-consuming; the clustering algorithm should be fast enough for online calculation; and the generated clusters should have readable descriptions for quick browsing by users, etc. We also follow these requirements to design our algorithm.

In this paper, we reformalize the search result clustering problem as a salient phrases ranking problem. Thus we convert an unsupervised clustering problem to a supervised learning problem. Although a supervised learning method requires additional training data, it makes the performance of search result grouping significantly improve, and enables us to evaluate it accurately. Given a query and the ranked list of search results, our method first parses the whole list of titles and snippets, extracts all possible phrases (n-grams) from the contents, and calculates

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

several properties for each phrase such as phrase frequencies, document frequencies, phrase length, etc. A regression model learned from previous training data is then applied to combine these properties into a single salience score. The phrases are ranked according to the salience score, and the top-ranked phrases are taken as salient phrases. The salient phrases are in fact names of candidate clusters, which are further merged according to their corresponding documents.

Our method is more suitable for Web search results clustering because we emphasize the efficiency of identifying relevant clusters for Web users. It generates shorter (and thus hopefully more readable) cluster names, which enable users to quickly identify the topics of a specified cluster. Furthermore, the clusters are ranked according to their salience scores, thus the more likely clusters required by users are ranked higher.

The paper is organized as follows. Some related works are introduced in Section 2. The problem is defined in Section 3, together with the whole algorithm described. In Section 4, we enumerate several properties for salient phrase ranking. The learning techniques that combine these properties are described in Section 5. The evaluations and clustering result examples are presented in Section 6. Finally we conclude the paper and give some future works in Section 7.

2. RELATED WORKS

The problem of clustering search results has been investigated in a number of previous works. Some of them (e.g. [3][6][11][10]) apply traditional clustering algorithms which first cluster documents into topically-coherent groups according to content similarity, and generate descriptive summaries for clusters. However, these summaries are often unreadable, which make it difficult for Web users to identify relevant clusters. Zamir and Etzioni [16][17] presented a Suffix Tree Clustering (STC) which first identifies sets of documents that share common phrases, and then create clusters according to these phrases. Our candidate phrase extraction process is similar to STC but we further calculate several important properties to identify salient phrases, and utilize learning methods to rank these salient phrases.

Some topic finding [1][8] or text trend analysis [9] works are also related to our method. The difference is that we are given titles and short snippets rather than whole documents. Meanwhile, we train regression model for the ranking of cluster names, which is closely related to the efficiency of users' browsing.

3. PROBLEM FORMALIZATION AND ALGORITHM

We convert the unsupervised clustering problem into a supervised ranking problem. More precisely, we are given the original ranked list of search result $R=\{r(d_i|q)\}$, where q is current query, d_i is a document, and r is some (unknown) function which calculates the probability that d_i is relevant to q. Traditional clustering techniques attempt to find a set of topic-coherent clusters C according to query q. Each cluster is associated with a new document list, according to the probability that d_i is relevant to both q and current cluster:

$$C = \{R_j\}, \text{ where } R_j = \{r(d_i|q, R_j)\} \quad (1)$$

In contrast, our method seeks to find a *ranked* list of clusters C', with each cluster associated with a cluster name as well as a new ranked list of documents:

$$C' = \{r'(c_k, R_k|q)\}, \text{ where } R_k = \{r(d_i|q, c_k)\} \quad (2)$$

As shown in Eq. 1 and Eq. 2, we modify the definition of clusters by adding cluster names c_k, and emphasize the ranking of them by function r', in order to improve the readability of clusters. Since we eliminate the requirement of topic-coherence of clusters, the complexity of the algorithm is lowered down. The non-topic-coherence isn't supposed to be a drawback of our method because it doesn't affect the efficiency of users' browsing behavior.

Our algorithm is composed of the four steps:

1. Search result fetching,
2. Document parsing and phrase property calculation,
3. Salient phrase ranking, and
4. Post-processing.

We first get the webpage of search results returned by a certain Web search engine. These webpages are analyzed by an HTML parser and result items are extracted. Generally, there are only titles and query-dependent snippets available in each result item. We assume these contents are informative enough because most search engines are well designed to facilitate users' relevance judgment only by the title and snippet, thus it is able to present the most relevant contents for a given query. Each extracted phrase is in fact the name of a candidate cluster, which corresponds to a set of documents that contain the phrase. Meanwhile, several properties for each distinct phrase are calculated during the parsing. These properties are described in detail in Section 4.

In the parsing, titles and snippets can be weighted differently, since there is generally a higher probability that salient phrases occur in titles. We apply stemming to each word using Porter's algorithm. The stop words are included in n-gram generation, so that they could be shown when they are adjacent to meaningful keywords in cluster names. In the post-processing, we filter out pure stop words. For the same reason, the query words themselves are also included in the parsing but are filtered out in the post-processing.

Given the properties, we utilize a regression model (as described in Section 5), which is learned from previous training data, to combine these properties into a single salience score. The salience phrases are then ranked by the score in descending order. After salient phrases are ranked, the corresponding document lists constitute the candidate clusters, with the salient phrases being cluster names.

In the post-processing, the phrases that contain only stop words or the query words are filtered out. We then merge the clusters and phrases, to reduce duplicated clusters. Specifically, if the overlapped part of two clusters exceeds a certain threshold (in our experiment, we use 75% as the threshold), they are merged into one cluster. Meanwhile, the cluster names are adjusted according to the new generated cluster. Finally, the topmost clusters are shown to user.

When a user selects a cluster, the corresponding document list is shown to the user, with both query words and salient phrases highlighted. This document list could be in the original order, or be re-ranked according to the associated salient phrase.

4. SALIENT PHRASES EXTRACTION

In this section, we list the five properties which are calculated during the document parsing. These properties are supposed to be relative to the salience score of phrases. In the following, we denote the current phrase (an *n*-gram) as *w*, and the set of documents that contains *w* as *D*(*w*).

Phrase Frequency / Inverted Document Frequency

This property is calculated just as the traditional meaning of Term Frequency / Inverted Document Frequency (*TFIDF*).

$$TFIDF = f(w) \cdot \log \frac{N}{|D(w)|} \tag{3}$$

where *f* represents frequency calculation.

Intuitively, more frequent phrases are more likely to be better candidates of salient phrases; while phrases with higher document frequency might be less informative to represent a distinct topic.

Phrase Length

The phrase length (denoted by *LEN*) property is simply the count of words in a phrase. For example, *LEN*("big")=1 and *LEN*("big cats")=2. Generally, a longer name is preferred for users' browsing.

$$LEN = n \tag{4}$$

Intra-Cluster Similarity

Intuitively, if a phrase is a good representation of a single topic, the documents which contain the phrase will be similar to each other. We use Intra-Cluster Similarity (*ICS*) to measure the content compactness of documents contain the phrase. First, we convert documents into vectors in the vector space model: $\mathbf{d}_i=(x_{i1}, x_{i2}, ...)$. Each component of the vectors represents a distinct uni-gram and is weighted by TFIDF of this uni-gram. For each candidate cluster, we then calculate its centroid as:

$$\mathbf{o} = \frac{1}{|D(w)|} \sum_{d_i \in D(w)} \mathbf{d}_i$$

ICS is calculated as the average cosine similarity between the documents and the centroid.

$$ICS = \frac{1}{|D(w)|} \sum_{d_i \in D(w)} \cos(\mathbf{d}_i, \mathbf{o}) \tag{5}$$

Cluster Entropy

For given phrase *w*, the corresponding document set *D*(*w*) might overlaps with other $D(w_i)$ where $w_i \neq w$. At one extreme, if *D*(*w*) is evenly distributed in $D(w_i)$, *w* might be a too general phrase to be a good salient phrase. At the other extreme, if *D*(*w*) seldom overlaps with $D(w_i)$, *w* may have some distinct meaning. Take query "jaguar" as an example, "big cats" seldom co-occur with other salient keywords such as "car", "mac os", etc. Therefore the corresponding documents may constitute a distinct topic. However, "clubs" is a more general keyword which may co-occur with both "car" and "mac os", thus it should have less salience score.

We use Cluster Entropy (CE) to represent the distinctness of a phrase.

$$CE = -\sum_t \frac{|D(w) \cap D(t)|}{|D(w)|} \log \frac{|D(w) \cap D(t)|}{|D(w)|} \tag{6}$$

where it is defined that 0·log0=0.

Phrase Independence

According to [2], a phrase is independent when the entropy of its context is high (i.e. the left and right contexts are random enough). We use *IND* to measure the independence of phrases. The following is the equation for IND_l which is independence value for left context, where 0·log0=0 is also defined.

$$IND_l = -\sum_{t=l(W)} \frac{f(t)}{TF} \log \frac{f(t)}{TF}$$

The IND_r value for right context could be calculated similarly. The final *IND* value is the average of those two.

$$IND = \frac{IND_l + IND_r}{2} \tag{7}$$

5. LEARNING TO RANK SALIENT PHRASES

Given the above five properties, we could use a single formula to combine them and calculate a single salience score for each phrase. However, this might be too heuristic to adapt to different domains. Instead, we utilize training data to learn a regression model.

Regression is a classic statistical problem which tries to determine the relationship between two random variables $\mathbf{x} = (x_1, x_{2,\ ...,}\ x_p)$ and *y*. In our case, independent variable **x** can be just the vector of the five properties described in Section 4: **x** = (*TFIDF*, *LEN*, *ICS*, *CE*, *IND*), and dependent *y* can be any real-valued score. We use *y* to sort salient keywords in a descending order, thus the most salient keywords are shown on the top.

Several regression models could be used, such as linear regression, logistic regression [5] and support vector regression [5][13]. We summarize them in the below and will further compare their effectiveness in the experiments.

Linear Regression

Linear regression attempts to explain the relationship of **x** and *y* with a straight line fit to the data. The linear regression model postulates that:

$$y = b_0 + \sum_{j=1}^{p} b_j x_j + e \tag{8}$$

where the "residual" *e* is a random variable with mean zero. The coefficients b_j ($0 \le j \le p$) are determined by the condition that the sum of the square residuals is as small as possible. Therefore the linear combination with b_j should be better than those with any other coefficients. The variables X_j can come directly from inputs, or some transformations, such as log or polynomial, of inputs.

Logistic Regression

When the dependent variable *Y* is a dichotomy, logistic regression is more suitable because what we want to predict is not a precise numerical value of a dependent variable, but rather the probability that it is 1 rather than 0 ($q = P(y=1)$).

Logistic regression attempts to find coefficients b_j ($0 \le j \le p$) to fit **x** to a logistic transformation of the probability, which is also called logit.

$$\text{logit}(q) = \log\frac{q}{1-q} = b_0 + \sum_{j=1}^{p} b_j x_j + e \qquad (9)$$

Whereas q can only range from 0 to 1, logit(q) ranges from negative infinity to positive infinity.

Instead of using a least-squared deviations criterion for the best fit, logistic regression uses a maximum likelihood method, which maximizes the probability of getting the observed results given the regression coefficients.

Support Vector Regression

In support vector regression, the input **x** is first mapped onto a high dimensional feature space using some nonlinear mapping, and then a linear model is constructed in this feature space. Support vector regression uses a new type of loss function called ε-insensitive loss function:

$$L_\varepsilon(y, f(\mathbf{x},\omega)) = \begin{cases} 0 & \text{if } |y - f(\mathbf{x},\omega)| \le \varepsilon \\ |y - f(\mathbf{x},\omega)| - \varepsilon & \text{otherwise} \end{cases} \qquad (10)$$

Support vector regression tries to minimize $\|\omega\|^{2.}$ This can be described by introducing (non-negative) slack variables ξ_i, ξ_i^*, i=1,...,n, to measure the deviation of training samples outside ε-insensitive zone. Thus support vector regression is formalized as minimization of the following functional:

$$\min \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$\text{s.t.} \begin{cases} y_i - f(\mathbf{x}_i,\omega) \le \varepsilon + \xi_i^* \\ f(\mathbf{x}_i,\omega) - y_i \le \varepsilon + \xi_i \\ \xi_i\ , \xi_i^* \ge 0, i = 1,...,n \end{cases} \qquad (11)$$

This optimization problem can be transformed into the dual problem and so that non-linear kernel functions could be used to do non-linear regression.

6. EXPERIMENTS

We conduct several experiments to validate the effectiveness of the proposed properties and learning methods.

6.1 Experiment Setup

A real search result clustering system is designed, as shown in Figure 1. The system accepts query inputs from users and pass them to one of the following search engines: Google, MSN, and AltaVista (but in our experiments, only MSN is used). The default result numbers are set to 200. This system is used for both training data collection and algorithm evaluation.

In the parsing, we extract all n-grams from the documents where $n \le 3$, and the phrases with frequency no greater than 3 times are considered as noise and are filtered out.

We use SVM-Light [7] and set the option "-z r" to do support vector regression. In all the support regression experiments, the parameters C and ε are set to default.

6.1.1 Evaluation Measure

Traditional clustering algorithm is difficult to be evaluated, but in our method, evaluation is relatively easy because the problem is defined to be a ranking problem. Thus we could use classical evaluation method in Information Retrieval.

We use precision (P) at top N results to measure the performance:

$$P@N = \frac{|C \cap R|}{|R|} \qquad (12)$$

where R is the set of top N salient keywords returned by our system, and C is the set of manually tagged correct salient keywords. In most our experiments, we use P@5, P@10 and P@20 for evaluation.

6.1.2 Training Data Collection

We asked 3 human evaluators to label ground truth data for 30 queries. The 30 queries are selected from one day's query log from MSN search engine. We specially select three types of queries: ambiguous queries, entity names and general terms, since these queries are more likely to contain multiple sub-topics and will benefit more from clustering search results. All the 30 queries are listed in Table 1.

Table 1. Thirty queries selected from query log

Type	Queries
Ambiguous queries	jaguar, apple, saturn, jobs, jordan, tiger, trec, ups, quotes, matrix
Entity names	susan dumais, clinton, iraq, dell, disney, world war 2, ford
General terms	health, yellow pages, maps, flower, music, chat, games, radio, jokes, graphic design, resume, time zones, travel

For each query, we extract all the n-grams (n≤3) from the search results as candidate phrases, order them alphabetically, and show them to the evaluators. There are one or two hundreds candidate phrases for each query. The three evaluators are asked to first browse through all search results returned by our system, and then select from the candidates 10 "good phrases" (assign score 100 to them) and 10 "medium phrases" (assign score 50 to them). The scores of other phrases are zero. The agreements of the 3 evaluators are high for good phrases. For example, in all the good phrases for query "jaguar", 5 phrases are selected by all 3 evaluators, 4 are selected by 2 evaluators and other 13 are selected by only 1 evaluator. But for medium phrases the agreements are much lower.

Finally, we add the three scores together and assign 1 to the y values of phrases with score greater than 100, and assign 0 to the y values of others. The average ratio of the positive examples (whose y value is 1) is about 0.17. Take "jaguar" as example again, there are totally 130 examples, in which 20 are positive examples. We only assign 0 or 1 to the y values to facilitate the comparison of 3 regression models, but it should be noted that the testing output of regression model ranges from 0 to 1 in logistic regression, and ranges from negative infinity to positive infinity in other regressions.

The manually selected phrases often fail to match against our generated phrases just because of some minor difference. Here we store each manually tagged phrase as a sequence of word stems, with stop words removed. Generated phrases are processed in the same way before we do exact matching.

6.2 Experimental Results

We first compare different properties and different learning methods for salient phrases ranking.

6.2.1 Property Comparison

We first use the each single property described in Section 4 to rank phrases, and evaluate the precisions for all the 30 queries. The average precisions at top 5, top 10 and top 20 are shown in Figure 2. Note that many phrases have the same *LEN* value, so *TFIDF* is used as secondary ranking criterion in the evaluation of *LEN*.

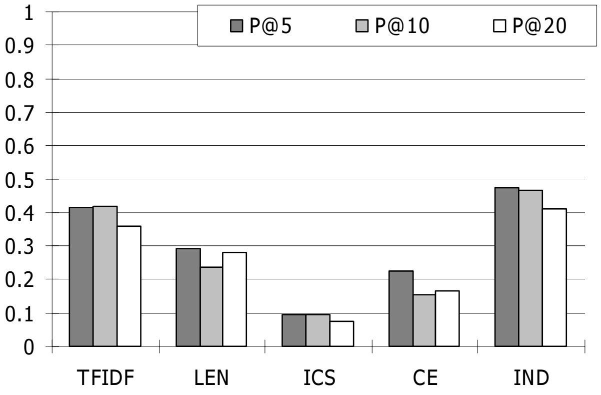


Figure 2. Performance for each single property

From Figure 2, we can see that each property doesn't work very well alone, but Phrase Independence (*IND*) (whose P@5=47.4%) and *TFIDF* (whose P@5=41.4%) are better indicators for phrase salience score. It is interesting to note that Intra-Cluster Similarity (*ICS*) is a not a good indicator. The reason might be that documents are composed of short titles and snippets, so that the vector space model-based similarity has large error.

6.2.2 Learning Methods Comparison

We randomly partition the ground-truth data into 3 parts and use three-fold cross validation to evaluate the average performance of linear regression, logistic regression, and support vector regressions. For support vector regression, different kernel functions are used: linear kernel (denoted by SV-L), RBF kernel (denoted by SV-R) and sigmoid tanh kernel (denoted by SV-S). The comparison of the 5 methods is shown in Figure 3.

By using regression, we achieve significant improvement on the precision compared to any single property. For example the p@5 of linear regression is 73.3%, outperforms about 30% over the best precision when using single property *IND*. We can also find that the performance of linear regression (P@5=73.3%), logistic regression (P@5=72%) and support vector regression with linear kernel (P@5=71.3%) are almost same. This shows the linearity of our problem.

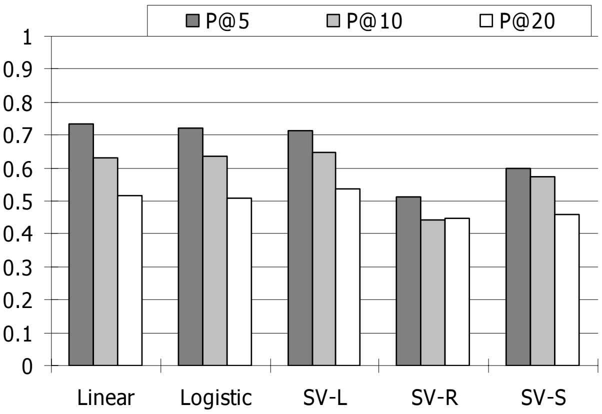


Figure 3. Performance comparison for different regression methods

We write down the coefficients of one of the linear regression models, as follows:

$$
\begin{aligned}
y = -0.427 \;& + 0.146 \times TFIDF \\
& + 0.241 \times LEN \\
& - 0.022 \times ICS \\
& + 0.065 \times CE \\
& + 0.266 \times IND
\end{aligned}
$$

In the above equation, each single property is normalized by their corresponding maximal value, so that we could observe which property plays more important role in the linear combination. From the above equation, we can see that the *IND*, *LEN* and *TFIDF* are more important than other properties. The coefficient of Intra-Cluster Similarity (*ICS*) is even negative, which also indicate that the content-based similarity has small impact on the salient phrase ranking.

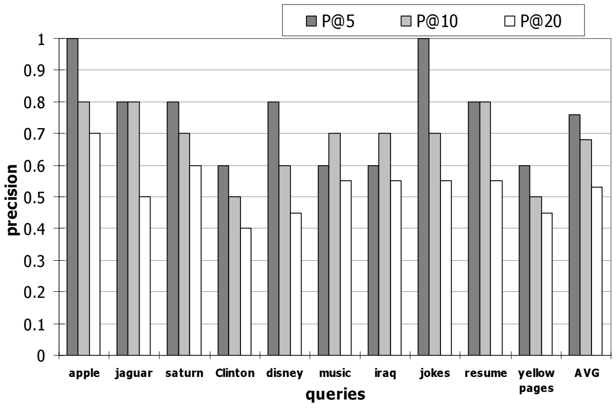


Figure 4. Using support vector regression (linear kernel) for various queries

Figure 4 shows individual ranking precision for 10 example queries, where we use the rest of queries as training data. The X-axis is the 10 queries with the average of them at the right-most column. From this figure, we can see that the performance depends heavily on the search results returned by Web search engine. For some queries such as "apple" and "jokes", the Web search engine results are mainly in a single domain (most results for "apple" are about computer). Therefore the vocabularies are relatively limited, and the salient phrases can be extracted precisely. But for queries like "Clinton" and "yellow pages", the search engine results contain various vocabularies. The performance for them is relatively low.

6.2.3 *Input Document Number*

We also use the precision of one query to explain the reason why we use top 200 search results as basic document set, as the experiment result shown in Figure 5. It is clear that the three precision measures arrive at peak when the result count equal to 200. Although the training set is based on 200 search results, the figure still effectively shows that our algorithm only require a small number of document input to achieve fairly good performance.

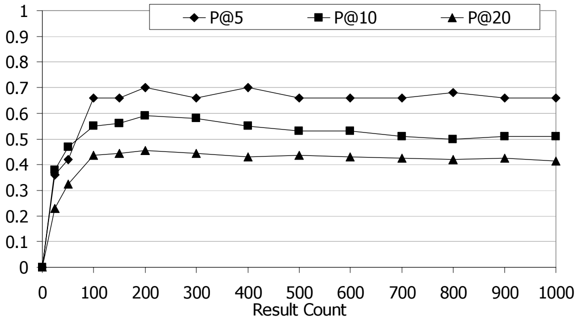


Figure 5. Performance curve along with document number

6.2.4 *Time Complexity, Coverage and Overlap*

We select a query as the example and analyze the time complexity of our algorithm as shown in Figure 6, in which the X-axis stands for the number of results returned from original search engine, and the Y-axis is the time spent in the whole algorithm (in seconds). The support vector regression is chosen as regression model. We didn't optimize the program code, and the time values in this figure are total processing time, including Web page parsing. But we could still observe from this figure that the time complexity is approximately linear.

Figure 7 shows the coverage of clusters generated by our algorithm for 10 queries. The X-axis is the 10 queries with the average of them at the right-most column. We can see from the figure that, in average, the clusters of top 10 salient phrases contain about half of the search results. This might be a drawback of our proposed method, compared to traditional clustering algorithms. We will further refine it in the future by designing more sophisticate cluster merge algorithm.

Figure 8 shows the overlap of the top N clusters. The X-axis is the same 10 queries as Figure 7. In average, the overlap of top 5 clusters is about 35%, which means there are about 65 distinct documents in 100 documents. The overlap of top 20 clusters is about 60%, which means there are only 40 distinct documents in 100 documents.

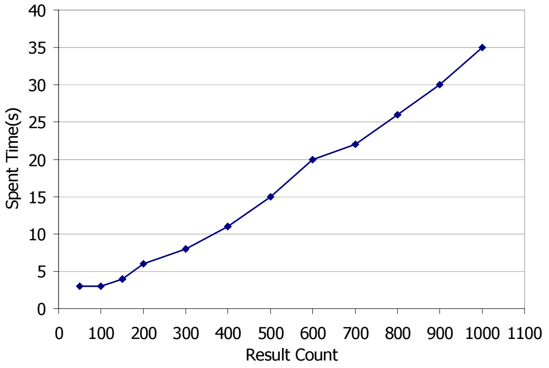


Figure 6. Time complexity analysis

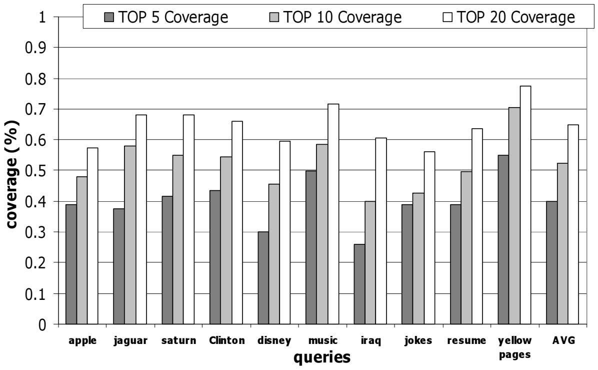


Figure 7. Coverage of generated clusters

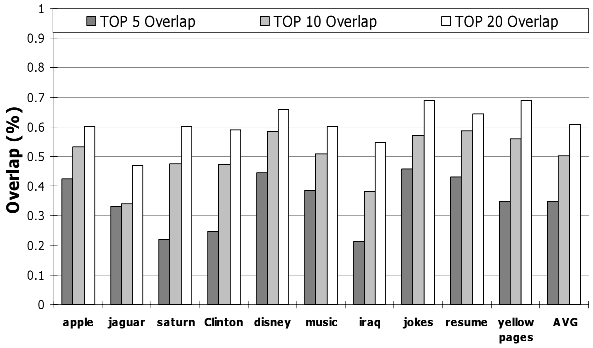


Figure 8. Overlap of generated clusters

6.2.5 *Example Queries*

We select three queries from three types of queries, i.e., "jaguar" from ambiguous queries, "iraq" from entity names and "resume" from general terms. For the three queries, we list the top ten salient phrases and the corresponding top five document titles, as shown in Figure 9. For each salient phrase, we also show its occurrence frequency in parenthesis.

7. CONCLUSION AND FUTURE WORKS

We reformalize the search result clustering problem as a supervised salient phrase ranking problem. Several properties, as well as several regression models, are proposed to calculate salience score for salient phrase. Experimental results demonstrate that we can generate correct clusters with short names (thus hopefully is more readable), thus could improve users' browsing efficiency through search result.

We will further investigate several problems on search result clustering. First, we will try to extract syntactic features for keywords and phrases to assist the salient phrase ranking. Second, current clustering is still a flat clustering method. We believe a hierarchical structure of search results is necessary for more efficient browsing. Third, some external taxonomies such as Web directories contains much knowledge which is familiar to Web users, thus a combination of classification and clustering might be helpful in this application.

8. REFERENCES

[1] Liu B., Chin C. W., and Ng, H. T. Mining Topic-Specific Concepts and Definitions on the Web. In Proceedings of the Twelfth International World Wide Web Conference (WWW'03), Budapest, Hungary, 2003.

[2] Chien L. F. PAT-Tree-Based Adaptive Keyphrase Extraction for Intelligent Chinese Information Retrieval. In Proceedings of the 20th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), pages 50-58, Phliadelphia, 1997.

[3] Cutting D. R., Karger D. R., and Pederson J. O. Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections. In Proceedings of the 16th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93), pages 125-135, Pittsburgh, PA, 1993.

[4] Google search engine, (2004) http://www.google.com.

[5] Hastie T., Tibshirani R., and Friedman J. The Elements of Statistical Learning. New York: Springer-Verlag, 2001.

[6] Hearst M. A., Pedersen J. O. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96), Zurich, June 1996.

[7] Joachims T., Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning. Schölkopf B. and Burges C. and Smola A. (ed.), MIT-Press, 1999.

[8] Lawrie D. and Croft W. B. Finding Topic Words for Hierarchical Summarization. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), pages 349-357, 2001.

[9] Lent B., Agrawal R., and Srikant R. Discovering Trends in Text Databases. In Proceedings of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD'97), Newport Beach, California, August 1997.

[10] Leouski A. V. and Croft W. B. An Evaluation of Techniques for Clustering Search Results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst, 1996.

[11] Leuski A. and Allan J. Improving Interactive Retrieval by Combining Ranked List and Clustering. Proceedings of RIAO, College de France, pp. 665-681, 2000.

[12] MSN search engine, (2004) http://search.msn.com.

[13] Smola, A. J. and Schlkopf, B. A Tutorial on Support Vector Regression. NeuroCOLT2 Technical Report Series, NC2-TR-1998-030. October, 1998.

[14] Vivisimo clustering engine, (2004) http://vivisimo.com.

[15] Yahoo search engine, (2004) http://www.yahoo.com.

[16] Zamir O., Etzioni O. Grouper: A Dynamic Clustering Interface to Web Search Results. In Proceedings of the Eighth International World Wide Web Conference (WWW8), Toronto, Canada, May 1999.

[17] Zamir O., Etzioni O. Web Document Clustering: A Feasibility Demonstration, Proceedings of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR'98), 46-54, 1998.

Clustering results for "**jaguar**"

1.	**car** (43)	1. Factory Car Audio Repair For All Makes and Models ... 2. Factory Car Audio Repair For All Makes and Models ... 3. Jaguar - Classic cars for sale, classified ads, muscle cars, ... 4. Jaguar Cars 5. Jaguar Cars	6.	**models** (21)	1. Jaguar Models - Main Page (resin model kits) 2. Factory Car Audio Repair For All Makes and Models... 3. Factory Car Audio Repair For All Makes and Models ... 4. Jaguar automobile history and database - main index 5. Autobytel Jaguar Search Results
2.	**panthera onca** (18)	1. Jaguar (Panthera onca) 2. Jaguar - Panthera onca 3. Jaguar (Panthera onca) 4. CSG Species Accounts: Jaguar (Panthera onca) 5. jaguar	7.	**jaguar history** (8)	1. Jaguar automobile history and database - main index 2. Jaguar History 3. Jaguar History 4. AtariAge - Atari Jaguar History 5. Defending the Land of the Jaguar : A History of ...
3.	**atari jaguar** (13)	1. Atari Jaguar FAQ 2. AtariAge - Atari Jaguar Rarity Guide Search 3. Game Winners - Atari Jaguar cheats, codes, hints, ... 4. The Atari Times - Jaguar 5. AtariAge - Atari Jaguar History	8.	**cats** (16)	1. Race for the Big Cats from Care2.com and WCS 2. Racing Cats :: The unofficial Jaguar F1 Fan Site 3. Race for the Big Cats - Tiger, Jaguar and Snow ... 4. Jaguar 5. jaguar
4.	**mac os** (13)	1. Apple - Mac OS X 2. Apple - Mac OS X - Overview 3. Mac OS X 10.2 Jaguar - Page 1 - (09/2002) -10.2/... 4. Jaguar, next major Mac OS X update coming this ... 5. MacNN | Feature: Mac OS X 10.2 Jaguar Report	9.	**animal** (8)	1. Animal Fact Sheets 2. Yahooligans! Animals: Jaguar 3. Animal Spirit Guides Shamanism 4. Jaguar -- Kids' Planet -- Defenders of Wildlife 5. Jaguar Posters & Art Prints - Your Purchase ...
5.	**jaguar club** (14)	1. Jaguar Drivers Club (JDC) - Based at Luton, England. 2. Jaguar Clubs of North America 3. Jaguar Club of Florida 4. Jaguar car club in Seattle, Home of JDRCNWA 5. JEC Homepage	10.	**jaguar enthusiasts** (7)	1. Mac OS X 10.2 Jaguar - Page 1 - (09/2002) ... 2. E-Type Lovers 3. Carlynx - Jaguar, Jaguars, Jaguar Links. 4. Jaguar 5. Jaguar Clubs of North America

Clustering results for "**resume**"

1.	**cover letter** (31)	1. JobStar--Resumes & Cover Letters 2. Resume and Cover Letter Services by Resume to Referral 3. ResumeZapper.com - E-Mail your Resume and Cover ... 4. The Resume Guide, homepage for Susan Ireland's resume, ... 5. A and A Resume Services - resume and cover letter writing ...	6.	**professional resume** (14)	1. Preparing Your Resume - advice on quality resume reparation ... 2. Computer Jobs and Technical Employment for Computer ... 3. Resume Writing Service 4. Resume Writing 5. Chuck Kahn's Resume
2.	**job** (50)	1. Employment 911 - Job Search, Resume Posting, Job Posting, ... 2. Post Your Resume--Let the Job Find You - FlipDog.com 3. Computer Jobs and Technical Employment for Computer ... 4. Resume Writing Tips - 60 Free Resume and Job Search ... 5. The Resume Guide, homepage for Susan Ireland's resume, ...	7.	**free resume** (14)	1. Resume Writing Tips - 60 Free Resume and Job Search ... 2. Free Resume Examples and Resume Writers 3. Free Resume Examples 4. Resume Writing 5. earthtimes.org - Sites for resume
3.	**resume writing** (37)	1. RESUME WRITING | How to write a masterpiece of a resume 2. A and A Resume Services - resume and cover letter writing ... 3. Resume Writing Tips - 60 Free Resume and Job Search ... 4. JobWeb - Your Guide to Resume Writing 5. #1 Resume Writing Services & Resume Tips Resource Center	8.	**career** (18)	1. Resumes, Jobs, Employment and Career Resources 2. Monster: Career Advice 3. Career Kids Links 4. Career Center - Resume and Letter Writing 5. Employment 911 - Job Search, Resume Posting, Job Posting, ...
4.	**services** (40)	1. Resume and Cover Letter Services by Resume to Referral 2. #1 Resume Writing Services & Resume Tips Resource Center 3. Resume Writing Service 4. A and A Resume Services - resume and cover letter writing ... 5. Preparing Your Resume - advice on quality resume...	9.	**resume samples** (10)	1. A and A Resume Services - resume and cover letter writing ... 2. Resume Samples 3. Resume and Cover Letter Guide - Sample Resume 4. JobWeb - Resumes & Interviews 5. JobWeb - Your Guide to Resume Writing
5.	**employment** (15)	1. Computer Jobs and Technical Employment for Computer ... 2. Employment 911 - Job Search, Resume Posting, Job Posting, ... 3. Resumes, Jobs, Employment and Career Resources 4. InternJobs.com Reach Internship Employers with Your Resume 5. ZenSearch.com - e-Shopping Guide - Money & Employment ...	10.	**experience** (16)	1. JobWeb - Resumes & Interviews 2. The Writing Center at Rensselaer Polytechnic Institute 3. Chuck Kahn's Resume 4. Birds-Eye.Net Owner's (Bruce Bahlmann) Resume 5. Resume

Clustering results for "**iraq**"

1.	**war** (32)	1. AlterNet: War on Iraq 2. War in Iraq - Christianity Today Magazine 3. End The War 4. Iraq Aftermath: The Human Face of War: AFSC 5. NOLA.com: War on Iraq	6.	**country** (19)	1. Library of Congress / Federal Research Division / Country ... 2. U.S. Department of State: Iraq Country Information 3. Iraq Country Analysis Brief 4. ArabBay.com: Arab Countries/Iraq 5. Countries: Iraq: Arabic Search Engine: Directory of arabic ...
2.	**middle east** (31)	1. Middle East Studies: Iraq 2. Amnesty International Report 2002 - Middle East and North ... 3. Human Rights Watch: Middle East and Northern Africa : ... 4. Columbus World Travel Guide - Middle East - Iraq - Overview 5. Iraq/Middle East	7.	**special report** (13)	1. Guardian Unlimited | Special reports | Special report: Iraq 2. Operation Iraqi Freedom - A White House Special Report 3. RFE/RL Iraq Report 4. Amnesty International Report 2002 - Middle East and North ... 5. Ethnologue report for Iraq
3.	**map** (18)	1. UT Library Online - Perry-Casta?eda Map Collection - Iraq ... 2. ABC Maps of Iraq; Flag, Map, Economy, Geography, ... 3. Flags of Iraq - geography; Flags, Map, Economy, Geography, ... 4. Lonely Planet - Iraq Map 5. Map of Iraq	8.	**guide** (13)	1. Lonely Planet World Guide | Destination Iraq | Introduction 2. Columbus World Travel Guide - Middle East - Iraq - Overview 3. Herald.com - Your Miami Everything Guide 4. Kansas.com - Your Kansas Everything Guide 5. Kansascity.com - Your Kansas City Everything Guide
4.	**saddam hussein** (13)	1. Iraq Resource Information Site - News History Culture People ... 2. U.S. Department of State - Saddam Hussein's Iraq 3. Iraq Crisis - Global Policy Forum - UN Security Council 4. New Scientist | Conflict in Iraq 5. Almuajaha - The Iraqi Witness: home	9.	**united nations** (11)	1. Mission of Iraq to the United Nations 2. united nations 3. United for Peace and Justice 4. U.S. Department of State: Iraq Country Information 5. Iraq Crisis - Global Policy Forum - UN Security Council
5.	**human rights** (11)	1. Human Rights Watch: Middle East and Northern Africa : Iraq ... 2. Iraq: Amnesty International's Human Rights Concerns 3. Human Rights Watch: Background on the Crisis in Iraq 4. Iraq:Amnesty International's Human Rights Concerns for 5. Iraq Aftermath: The Human Face of War: AFSC	10.	**travel, business** (16)	1. Iraq: Complete travel information to Iraq, travel facts, ... 2. EIN news - Iraq - Political, Business and Breaking ... 3. Iraq - Travel Warning 4. Columbus World Travel Guide - Middle East - Iraq - ... 5. Iraq Visa Application - Tourist Visas, Business Visas, ...

Figure 9. Clustering result for query "jaguar", "resume" and "iraq"