Semantic Website Clustering

I-Hsuan Yang, Yu-tsun Huang, Yen-Ling Huang

1. Abstract

We propose a new approach to cluster the web pages. Utilizing an iterative reinforced algorithm, the model extracts semantic feature vectors from user click-through data. We then use LSA (Latent Semantic Analysis) to reduce the feature dimension and K-means algorithm to cluster documents. Compared to the traditional way of feature extraction (lexical binomial model), our new model has better purity (75%) and F-measure (52%). We can further use features combined from both methods and reach purity 82% and F-measure 52%. Moreover, the same method can be used to cluster queries, and with the result purity 74% and F-measure 43%.

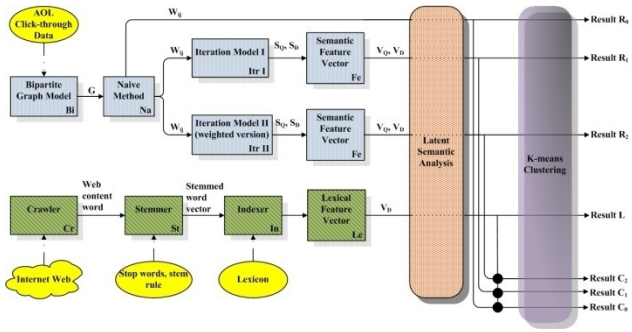
2. Introduction

As the tremendous fast growing speed of the number of web pages on the Internet, automatic approach for clustering web pages are more desirable than ever before. Many research groups are developing various techniques trying to solve the web page clustering problem and there are also many applications available as well. Besides the lexical features which are used for traditional clustering method, we further consider utilizing click-through features in order to capture more relationship on semantics. We revise and improve the iterative reinforced algorithm to define new similarity measure in several dimensions on which we can build semantic cluster on both query and web pages.

In the traditional cluster, it uses only the text to build features, but there is a lot of information such as picture, multimedia, meta-data, hyperlinks… that won't be captured. Since the click through data is manipulated by a huge set of users, it can be seen as the judgment from actual users. Therefore, the real semantic similarity can be easily observed via calculating the users' real clicks. We will use the AOL click-through logs and web pages automatically crawled from the Internet as the training data. We then compare the performance between our new features and traditional lexical based features. We also use User perception test as our evaluation method. Besides web page clustering, we can use this method as clustering query terms. There are also useful applications for query term cluster, such as also try suggestions, query term auto completion.

3. Proposed Approach

Figure 1: Clustering Model Flow Diagram



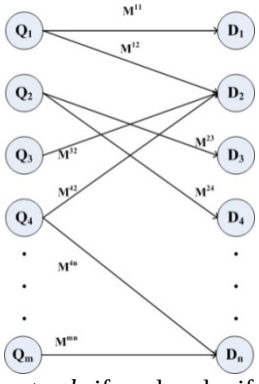
3.1 Semantic feature vector extraction (blue blocks in Figure 1)

Instead of lexical frequency, we want to use the real semantic meaning of each web page. Using the AOL click-through data fits our needs, since those clicks are judged from real users' cognition and conception. We assume the relevance between queries and documents is very accurate based on a great deal of data cumulated. Here the documents are web pages.

Whenever a user clicks a link from the search result page (AOL data is from Google search engine), it produces a vector containing five fields: user's ID (usually cookie number), query, query timestamp, item rank, URL. Here we use the query and URL field $\langle Q, D \rangle$ to build our bipartite graph model.

3.1.1 Bipartite Graph Model

In Xue's recent work [1], they used a bipartite graph model to fit the click through data scenario and improved the search result relevance. One node of left hand side represents one query key word and one node of right hand side represents one document. Assume there are m queries and n documents, the graph contains node $q_i \; \forall i \in \{1, \dots, m\}$ at left and $d_j \; \forall j \in \{1, \dots, n\}$ at right. There exists a link from q_i to d_j if and only if there is at least one click data



containing the tuple $\langle Q_i, D_j \rangle$. Moreover, each link has weight M^{ij}, which means the frequency of the tuple $\langle Q_i, D_j \rangle$. It can be seen as one kind of relevance between Q_i and D_j.

In this project, since the difficulty of labeling the gold standard clustering, we use a subgraph with two different sizes $m = n \cong 100$ and $m = n \cong 1000$. Therefore, the input of this model is the raw AOL click-through data and output will be a data structure of a bipartite graph.

3.1.2 Naïve Method Feature Vector

For each document, we will produce a feature vector with dimension m (the num of total queries). Define the normalized frequency:

$$W_{ij} = \frac{M^{ij}}{\sum_{k \in \{1,\dots,m\}} M^{kj}}$$

Therefore, for document $d_j, j \in \{1, \dots, n\}$, it contains the feature vector$\langle W_{1j}, W_{2j}, \dots, W_{mj} \rangle$.

3.1.3 Iteration Method I

The features from naïve method can be used for clustering itself, but there are some problems to overcome: noise, incompleteness, sparseness and volatility. The point is we don't want too many zeros' in one vector, whereas the original weight W is very sparse. Here, we utilize an iteration method to fix the problem of raw click data. This method is based on the assumption of co-visited method: Web pages are similar if they are visited by similar queries, and queries are similar if they visit similar web pages. Here we define $S_Q[q_s, q_t]$ to be the similarity between queries q_s and q_t, and $S_D[d_s, d_t]$ to be the similarity between documents d_s and d_t. S_D and S_Q are initialized to 1 if the components are identical.

$$S_Q^{(0)} = \begin{cases} 0 \ (q_s \neq q_t) \\ 1 \ (q_s = q_t) \end{cases}$$
$$S_D^{(0)} = \begin{cases} 0 \ (d_s \neq d_t) \\ 1 \ (d_s = d_t) \end{cases}$$

For each iteration, we use the update rule based on the co-visited assumption above and update until converge:

$Repeat\ until\ converge\{$

$$S_Q[q_s, q_t] = \frac{C}{|O(q_s)||O(q_t)|} \sum_{i=1}^{|O(q_s)|} \sum_{j=1}^{|O(q_t)|} S_Q[O^i(q_s), O^j(q_t)]$$

$$S_D[d_s, d_t] = \frac{C}{|I(d_s)||I(d_t)|} \sum_{i=1}^{|I(d_s)|} \sum_{j=1}^{|I(d_t)|} S_D[I^i(d_s), I^j(d_t)]$$

$\}$

Where C is the decay factor, and is set to be 0.7 here. $O(q_s)$ represents all the out-links for query q_s, and $I(d_s)$ represents all the in-links for document d_s. After several runs of iteration (in our experiments, 10~15 runs for size 100 and 20~25 runs for size 1000), S_Q and S_D will converge to a fixed number and to be output to the semantic feature vector extraction model.

3.1.4 Iteration Method II

Since Iteration method I doesn't use the frequency M^{ij}, Iteration method II uses the weighted version. The revised version of the update rule:

$Repeat\ until\ converge\{$

$$S_Q[q_s, q_t] = \frac{C}{\sum_{i \in O(q_s), j \in O(q_t)} M^{ij}} \sum_{i \in O(q_s), j \in O(q_t)} (M^{is} + M^{jt}) S_Q[O^i(q_s), O^j(q_t)]$$

$$S_D[d_s, d_t] = \frac{C}{\sum_{i \in I(d_s), j \in I(d_t)} M^{ij}} \sum_{i \in I(d_s), j \in I(d_t)} (M^{is} + M^{jt}) S_D[I^i(d_s), I^j(d_t)]$$

$\}$

In our experiments, it converges a little faster than method I.

3.1.5 Semantic Feature Vector

Now we have $S_Q[q_s, q_t] \quad \forall s, t \in \{1, \dots, m\}$, $S_D[d_s, d_t] \forall s, t \in \{1, \dots, n\}$, and $W_{ij} \ \forall i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$. Here we will use the co-visited assumption again: for each pair $i, j \ where \ i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$, first taking all of the queries q_k which is similar to q_j ($S_Q[q_s, q_t] \geq 0.01$), then sum up all the values $S_Q[q_s, q_t] \times W_{kj}$, and assign this summation to W'_{ij}. Therefore, the vector $\langle W'_{1j}, W'_{2j}, \dots, W'_{mj} \rangle$ will be the semantic feature vector of document d_j.

$$W'_{ij} = \sum_{k: S_Q[q_k, q_i] \geq 0.01} S_Q[q_k, q_i] \times W_{kj}$$

3.2 Lexical Feature Vector Extraction (green blocks in Figure 1)

We use lexical words as features to define the similarity between documents in terms of lexical meaning. Two documents would be considered highly related if they share plenty of common lexical words. We build a web crawler to extract all the content text in each document (In this paper it's actually website) ,applying the Porter stemming algorithm and stop word filter to filter out unnecessary word to create the lexicon for all training data. We build the index for each document from the lexicon and thus the feature for each document is consisted of all the lexicon words in the lexicon.

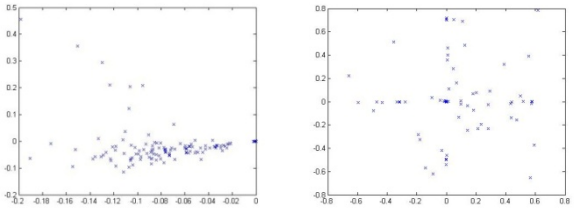
3.3 Latent Semantic Analysis (pink blocks in Figure 1)

Since we have a relatively sparse data even after applying the iteration method, we use LSA to reduce the dimension of the feature vector for the document and condense the feature vector to more "concept" space. Using the standard SVD decomposition to get the singular value and the eigenvector for $\Sigma^T\Sigma$ and $\Sigma\Sigma^T$, we set a target dimension k equal to the number of clusters labeled in gold-standard clustering (k=21 in our experiment) and reduce the higher dimension feature vector to k-dimension feature vector.

3.4 Combine Semantic Features with Lexical Features

We think that the two different kind of feature are essentially complementary. By combining these two feature sets we can achieve better performance. The lexical features can capture the lexical relation between documents and queries while the semantic feature sets can capture the semantic relationships. As shown in the Figure 2 below, we apply LSA to reduce the dimension to 2-D to show distribution of our training data in 2-D space demonstrating the idea. Semantic features can significantly help to explore the relationship between documents which could be hard to see in lexical features.

Figure 2: Document data distribution in 2-D projection



3.5 K-means Algorithm

We use K-means algorithm to cluster websites and user queries terms respectively. We set the k equal to the number of clusters labeled in gold-standard clustering and start with random seeds. The purity, precision, recall and F-measure are computed by averaging several trials.

3.6 Application

This approach can be applied to several applications.

3.6.1 Automatic web categorization

We always need a good way to automatically categorize the web pages on the Internet. The growing speed of the Internet is too fast that the cost of manually categorizing the web pages by hand is extremely expensive or even unaffordable. An automatically generated clustering provides the users an easier way to navigate and browse on the Internet.

3.6.2 Document-to-Document Search

By constructing the clusters of relevant documents, given a specific document, we can search for the documents which are highly related in terms of both lexical and semantic.

3.6.3 Query-to Query Search

By constructing the clusters of relevant query, given a specific query, we can provide the user other highly relevant query terms as the common "also try" functionality widely used on search engines and e-commerce documents.

3.6.4 Semantic Relevance Web Search

In our model, document d_j will have the feature vector $\langle W'_{1j}, W'_{2j}, \ldots, W'_{mj} \rangle$. This vector can be utilized to build additional metadata $W_{1j} \cdot q_1 + W_{2j} \cdot q_2 + \cdots + W_{mi} \cdot q_m$. For d_j. This metadata will improve search result quality by adding the semantic meaning into the relevance between query and document. By adjusting weight for semantic relevant metadata, we can have different kinds of search result (semantically or lexically).

4 Experiment

4.1 Data description

We use AOL click-through log data collection as our training data. This collection consists of ~20M web queries collected from ~650k users over three months from 01 March, 2006 - 31 May, 2006. The data is sorted by anonymous user ID and sequentially arranged. This click-through data collection provides real query log data that is based on real users. It could be used for personalization, query reformulation or other types of search research. The detail statistics of the collection is showed in Table 1.

Table 1 : AOL Click-Through dataset stats

Basic Collection Statistics	
lines of data	36,389,567
instances of new queries (w/ or w/o click-through)	21,011,340
requests for "next page" of results	7,887,022
user click-through events	19,442,629
queries w/o user click-through	16,946,938
unique (normalized) query	10,154,742
unique user ID's	657,426

Every data entry in this click-through collection data has the columns {AnonID, Query, QueryTime, ItemRank, ClickURL}.

AnonID - an anonymous user ID number.

Query - the query issued by the user, case shifted with most punctuation removed.

QueryTime - the time at which the query was submitted for search.

ItemRank - if the user clicked on a search result, the rank of the item on which they clicked is listed.

ClickURL - if the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

We extract two test dataset as our development set. The smaller set contains approximately 100 user queries and documents. The larger set contains approximately 1000 user queries and documents. After ranking all the queries and documents by the frequency in our dataset, we ignore the first 50 queries and documents in order to reduce the number of navigational search in our dataset and extract the queries, documents and corresponding links between them in two different sizes as our test set.

4.2 Experiment procedure

We conduct the experiment on both semantic feature vector and lexical feature vector and also the composite feature vector. Various parameters of linear combination of the two feature vector are used for the composite feature vector and we perform the experiments on two test sets of different size. The gold-standard clustering for our development sets are manually labeled by human for evaluation purpose. The experiment result and clustering example will be showed in the following section.

4.3 Evaluation Measures

There are three main approaches for the clustering evaluation: gold-standard, task-oriented and user evaluation. In our experiment, we use gold-standard as the evaluation measures for our clustering. For gold-standard approach, we manually construct an ideal clustering by human labeling. The ideal clusters are then compared against the machine generated clusters. A machine generated clustering is considered perfect if it matches ideal clustering in gold-standard. There are two ways for a cluster to be less than perfect: It may have poor quality as it doesn't match any cluster well in the gold-standard, and it may have poor coverage to those websites in gold-standard clustering. We use Purity [3] and F-measure [3] as our evaluation measures for a clustering. They are based on the precision and recall of the clusters. Precision is defined as the fraction of documents in the cluster that also appear in the gold-standard. Recall is defined as the fraction of documents in the gold-standard cluster that also appear in the machine generated cluster. Therefore, the Purity of a clustering is the average precision of all clusters relative to their best matching clusters in the gold-standard. The F-measure is defined as the average F-measure of the clusters relative to the clusters in gold-standard.

4.4 Result

We conduct the experiment on two tasks:

- Internet Website Clustering
- User Query Clustering

For website clustering, we conduct the experiment on two development sets. We have 7 result sets corresponding to 7 different feature vectors. In Result R0-R2 we only use the semantic features to generate the clustering. In Result L we only use lexical features to generate the clustering and in Result C0-C1 we use the composite features which contain both semantic and lexical information.

In the Table 2 below, we can see that naïve method and result L which only use lexical features give us baseline results. Iteration method I and II both perform better than naïve method as expected. We get even better result at 82% of purity and 52% of F-measure when combining those two feature sets together which is reasonable since we expect the two feature sets to be complementary. In Table 3 we show the experiment result on development set 2.

Due to the limitation of affordable work of manually labeling gold-standard, we use a slightly different method to evaluate the clustering. We only label part of the gold-standard and selectively match the machine generated clusters to the best matched cluster in the partial gold-standard and only evaluate the purity here to understand mainly the quality of the clustering. Same trend can be observed as in development set 1 that the iteration methods perform better than the naïve method. We also show the result with different parameter $alpha$ for the interpolation parameter when combining the two feature sets for Naïve method, iteration method I and iteration method II in Figure 3-5.

Table 2: Experiment result for document development set 1

Website Development Set 1 (size = 100)				
	Purity	F-measure	Precision	Recall
Result R0	0.7249	0.4448	0.7249	0.4209
Result R1	0.6942	0.4872	0.6942	0.4583
Result R2	0.7465	0.5154	0.7346	0.5019
Result L	0.7766	0.4062	0.7412	0.3484
Result C0	0.7989	0.4827	0.6491	0.4612
Result C1	0.8149	0.4959	0.6863	0.481
Result C2	0.8156	0.5154	0.7465	0.5019

Table 3: Experiment result for document development set 2

Website Development Set 2 (size = 1000)	
	Purity
Result R0	0.4514
Result R1	0.5811
Result R2	0.5687
Result L	0.5356
Result C0	0.638
Result C1	0.6046
Result C2	0.6536

Figure 3: Experiment Result for Naïve Method

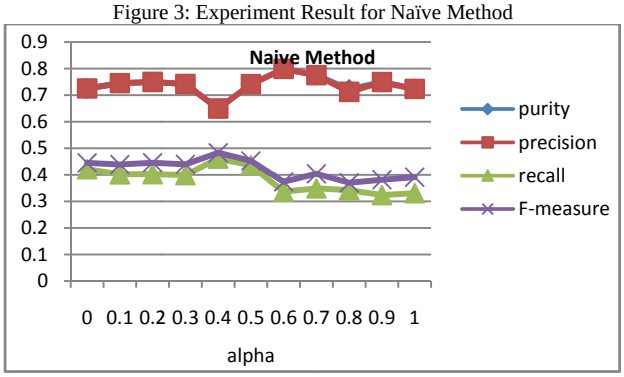


Figure 4: Experiment Result for Iteration Method I

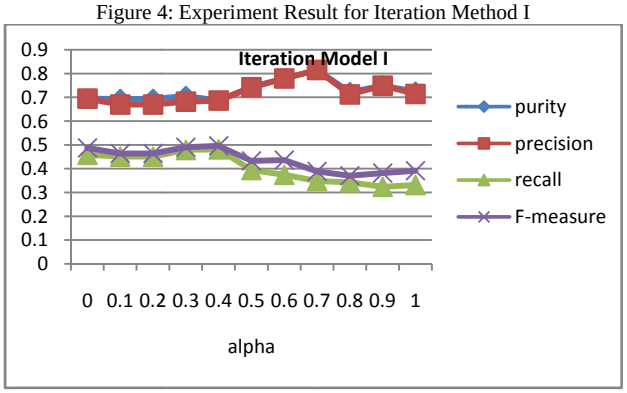
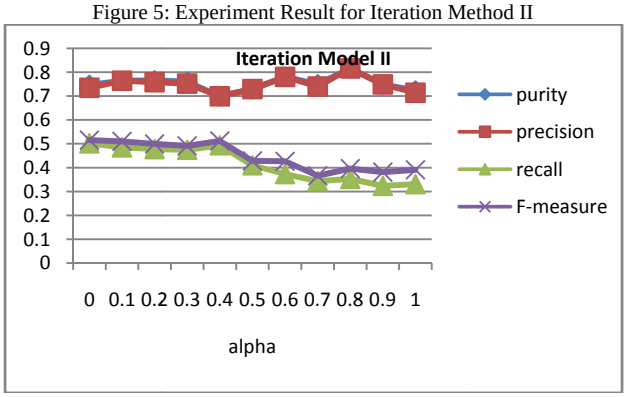


Figure 5: Experiment Result for Iteration Method II



We also conduct the experiment on the user query data. Using the same method described in Section 3.1, we can also compute the similarity between user queries and generate clustering on user queries.

Table 4: Experiment result for query development set

User Query Development Set 1 (size = 100)				
	Purity	F-measure	Precision	Recall
Result R0	0.7589	0.3733	0.7589	0.3135
Result R1	0.7377	0.4346	0.6751	0.419
Result R2	0.7224	0.3988	0.7154	0.3468

In Figure 6 below we show the actual clustering result from Result C2 for Website Development set 1. It is easy to see that similar websites in terms of both lexical and semantic meaning are clustered together.

Figure 6: Clustering example for website development set 1

5 Conclusion

In this web clustering project, we utilize the semantic information from click-through data to extract feature vectors. This novel feature extraction model complements the traditional lexical feature extraction method and provides an effective way to generate semantic rich clustering. The iteration method also overcomes the difficulty of sparseness from the nature of click-through data and converges to a fixed point representing the real similarity between queries and documents. Using the combination of semantic and lexical features we can achieve the best performance in terms of cluster purity and F-measure.

6 Reference

[1] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, W.G. Fan. Optimizing Web Search Using Web Click-through Data. In Proceedings of the thirteenth ACM international conference on Information and knowledge management, Washington D.C., USA. November 08-13, 2004.

[2] G. Pass, A. Chowdhury, C. Torgeson, "A Picture of Search" The First International Conference on Scalable Information Systems, Hong Kong, June, 2006.

[3] A. Strehl. Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining. PhD thesis, Faculty of the Graduate School of The University of Texas at Austin, 2002.