An Experimental Study on Automatically Labeling Hierarchical Clusters using Statistical Features

Pucktada Treeratpituk
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213, USA
puck@cs.cmu.edu

Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213, USA
callan@cs.cmu.edu

Categories and Subject Descriptors: H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing — *Linguistic processing.*

General Terms: Experimentation.

Keywords: document hierarchy, cluster labeling.

1. INTRODUCTION

Document hierarchies provide views of a collection at different levels of granularity, making it easy to visualize and explore large document collections. Topic descriptors at each level of the hierarchy play an important role in helping users to achieve those benefits. However, few previous works in automatic hierarchical document clustering focus on assigning good descriptors to the clusters they generate. A good cluster label should not only describe the main concept of the cluster, but also differentiate the cluster from its sibling and parent clusters. Consider a cluster "AI" in a hierarchy that contains sub-clusters such as "neural network," and "machine learning." Although "computer science" describes the main concept of the cluster "neural networks," in the context of this hierarchy it does not distinguish "neural networks" from its siblings. Most clustering algorithms describe a cluster with list of topical terms [2]. These topical terms are often selected based only on features from the cluster of interest, while ignoring its surrounding clusters. We propose a simple model that considers the structure of the hierarchy when assigning labels to clusters. Effectiveness of different statistical features is evaluated using the Open Directory Project as ground truth data. We also use a synthesized document hierarchy to investigate the effects of typical clustering errors on the effectiveness of the algorithm.

2. ALGORITHM

Most automatically-generated document hierarchies use lists of terms as topic descriptors. The terms selected as cluster labels are usually document terms that have high TF or TF.IDF values [1][2], and they are selected independently of other terms. However, the "goodness" of one term as a cluster label is related to its appearance in surrounding clusters. This paper proposes a model that uses both statistical features from the cluster and features from surrounding clusters to select cluster labels.

Each phrase in a clustered document is a potential label for the cluster. A simple linear model is used to combine a phrase's features into a *DScore* that measures its value as a cluster label. In addition to traditional features from the cluster itself, such as DF_S/#S, $TFIDF_S$, rank(DF_S/#S) and rank($TFIDF_S$), our DScore model also includes features from the parent cluster, such as DF_P/#P, $TFIDF_P$, rank(DF_P/#P) and rank($TFIDF_P$). In particular, the model includes the ranking-boosts between the cluster and its parent cluster in the features: log(rank(DF_P))-log(rank(DF_S)), and log(rank($TFIDF_P$))-log(rank($TFIDF_S$)). Ranking-boosts measure the relative importance increase of a phrase in the cluster from its parent. The log-scale in the ranking-boost formula reflects the intuition that the change in ranking is more important at the top of the ranking. For example, a term that the parent ranked 50th and the child ranked 5th is more important than one that the parent ranked 100th and the child ranked 55th. *DScore* is defined as:

$$
\begin{aligned}
DScore_P = {} & c_0 + c_1 * DF_S / \# S + c_2 * DF_P / \# P \\
& + c_3 * rank(DF_S) + c_4 * rank(DF_P) \\
& + c_5 * TFIDF_S + c_6 * TFIDF_P \\
& + c_7 * rank(TFIDF_S) + c_8 * rank(TFIDF_P) \\
& + c_9 * [\log(rank(DF_P)) - \log(rank(DF_S))] \\
& + c_{10} * [\log(rank(TFIDF_P)) - \log(rank(TFIDF_S))]
\end{aligned}
$$

The coefficients in the *DScore* model can be easily trained using any manually constructed hierarchies that provide a category label for each cluster, for example, the *Open Directory Project.* A training instance can be generated for each phrase in every cluster in the hierarchy. The *DScore* for a training instance phrase *P* in a cluster with a category label *CL* can be estimated with:

$$DScore_P{*} = \max_{SP \in Synonym(P)} \left\{ \frac{overlap(SP, CL)}{\max\{length(SP), length(CL)\}} \right\}$$

$$overlap(p1, p2) = \#\text{terms shared between p1 and p2}$$

The above formula estimates the *DScore* with the maximum overlap between the correct label and any synonyms of *P*, which can be obtained from sources like WordNet.

Since it is best to minimize the number of labels shown, the model can decide adaptively how many labels to display, based on the *DScore* of the top ranked label. If the top label has a high *DScore*, the model should only display the top ranked label, and vice versa. This threshold can also be automatically optimized using the same training data.

SIGIR'06, August 6–11, 2006, Seattle, WA, USA.

ODP category labels [parent/self]	#docs	Predicted labels (ranked by *DScore*)
artificial intelligence/agents	84	agent, software agent
artificial intelligence/conferences and events	72	conference, artificial intelligence, international conference
artificial intelligence/genetic programming	65	genetic, genetic algorithm
artificial intelligence/philosophy	46	philosophy, mind, science
security/conferences	14	secure conference, conference attend
security/honeypots and honeynets	62	honeypot, attack
security/news and media	73	attack, vulnerable, hack
alternative/ear candling	10	ear candle, ear
alternative/non-toxic living	53	toxic, environmental, safe
alternative/urine therapy	6	urine

Figure 1. Examples of labels ranked by the *DScore*, with an adaptive number of labels.

3. EXPERIMENTS & CONCLUSION

To evaluate the quality of labels produced, we used ODP category labels as ground truth data. We collected 25,143 web pages from a total of 165 ODP categories. For each category, we calculated *DScore* for every unigram, bigram and trigram phrase in the cluster that occurred more than a heuristic threshold (e.g., for unigrams, 20% of the documents in the cluster). All experiments were done using 5-fold cross-validation. In each fold four fifths of the categories were used to optimize the model's coefficients, while the remaining one fifth was used as the testing data. We used mean reciprocal rank (MRR), traditionally used in question-answering system evaluation, to measure the quality of the list of predicted labels. A predicted label is considered to be correct if one of its synonyms matches the cluster's category label in the ODP, which is a very strict definition of correctness. The synonyms were automatically obtained from WordNet.

Figure 2 shows the qualities of labels selected based on different features. Although the features traditionally used to rank cluster descriptors, such as DF$_S$/#S, TF$_S$, TFIDF$_S$, work reasonably well, features that take the parent cluster features into account, such as log[rank(DF$_P$)] - log[rank(DF$_S$)], produce much better labels. Combining every feature linearly decreases label quality. This is most likely due to the simplicity of the linear model. A further study is needed to study more effective combination functions.

So far we have assumed that the document hierarchy is perfect. However, our goal is to assign labels to hierarchies generated by clustering algorithms; they will not be perfect. Clustering errors were simulated in our testing data to provide a more realistic evaluation. We computed the cluster centroid for each category in the testing data; for each document we calculated the similarity between the document, the centroid of its cluster, and the centroids of its sibling clusters. If the closest centroid was not the document's correct cluster, the document was probabilistically swapped to the closest sibling cluster. The centroid calculation and similarity measure proposed in Scatter/Gather [2] were used to simulate clustering algorithm errors. Table 1. shows the MRR of the labels selected under different levels of clustering errors. The quality of the predicted labels suffered only slightly, even when 30% of the documents were assigned to the wrong cluster. We hypothesize that clustering errors tend to group homogenous documents together, thus the important topical terms are even more distinctively distributed, and the performance of a model that depends on statistical distributions does not decrease much.

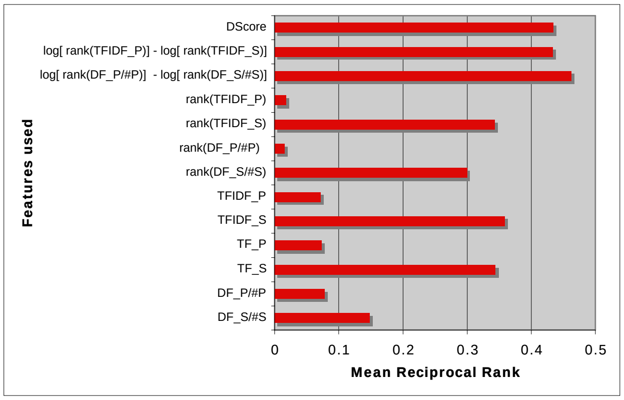

Figure 2. shows MRR for different features set.

In summary, we propose a simple linear model that considers the structure of the hierarchy when automatically assigning labels to document clusters in a hierarchy. We conducted a study to show the effectiveness of different statistical features in selecting cluster labels. We also showed that such a simple model is likely to tolerate the type of noise in the cluster hierarchy that is normally generated by clustering algorithms.

Table 1. MRR at different noise levels.

% of docs swapped	0%	7%	14.5%	22.5%	30%
MRR	0.51	0.51	0.51	0.48	0.50

Acknowledgements

This research was supported by NSF grant IIS-0429102. Any opinions, findings, conclusions, or recommendations are the authors', and do not necessarily reflect those of the sponsor.

4. REFERENCES

[1] Chuang S., and Chien L. A practical web-based approach to generating topic hierarchy for text segments. CIKM 2004.

[2] Cutting D. R., Karger D. R., and Pederson J. O. Constant interaction-time Scatter/Gather browsing of very large document collections. SIGIR 1993.

[3] Glover, E., Pennock, D., Lawrence, S. and Krovetz, R. Inferring hierarchical descriptions. CIKM 2002.

[4] Popescul, A., and Ungar, L. Automatic labeling of document clusters. Unpublished manuscript, available at http://citeseer.nj.nec.com/popescul00automatic.html, 2000.

[5] Zeng, H., He, Q., Chen Z., Ma, W., and Ma J. Learning to cluster web search results, SIGIR 2004.