Web People Search Based on Locality and Relative Similarity Measures

Fei Song
Dept. of Computing and Info. Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1
1-519-824-4120x58067
fsong@uoguelph.ca

Robin Cohen
Dept. of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
1-519-888-4567x34457
rcohen@uwaterloo.ca

Song Lin
Dept. of Computing and Info. Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1
slin@uoguelph.ca

ABSTRACT

In this paper, we describe our implementation for web people search 2. We emphasize two improvements for the process, including locality-based representation for feature vectors and relative similarity measures for hierarchically organizing web pages into different clusters. We achieved the results of 0.63 for Fmeasure_0.5_BEP_BER and 0.75 for Fmeasure_0.2_BEP_BER.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering.* I.5.3 [**Pattern Recognition**]: Clustering – *similarity measures.*

General Terms

Measurement, Performance, Design, Experimentation.

Keywords

People Search, Document Clustering, Locality-Based Weights, Relative Similarity Measures, Agglomerative Clustering.

1. INTRODUCTION

Search engines have dramatically changed the way people gather information. Instead of periodically scanning through different sources for relevant documents and organizing them into different folders for future use, we can simply type a query in a search engine and get most of the relevant results back in a matter of seconds. However, the users still need to play an active role in the process by formulating an appropriate query and sifting through the returned documents to extract the answers they are looking for. As the Web continues to grow at an accelerated speed, the burden is getting heavier on the user side, since we often get many search results for a typical query. Therefore, it is highly desirable to automate this process as much as possible so that the user can find the relevant information efficiently and effectively.

People search is intended to enhance the search results for queries that involve person names [1]. Many names are ambiguous, resulting in pages about different individuals with the same names returned from a search engine. By sorting these pages into different groups with one group corresponding to one individual, we can effectively reduce the workload for the users so that they can narrow down their focus to a particular group quickly.

As can be seen from the papers that describe the participating systems for Web People Search 2007 [1], most of the systems rely on a set of extracted features (or attributes such as occupation, location, nationality, phone number, email address, and URL) to describe a person and then based on the feature vectors of the related web pages and their similarities, they organize these web pages into different clusters for person name disambiguation ([3], [4], [5], [8], [11], and [12]).

In this paper, we examine two possible improvements on the representation of feature vectors and the clustering process. More specifically, we use a locality-based representation for feature vectors by adjusting the weights for each feature according to their minimum distance to all occurrences of the person name that is being disambiguated. The intention is to allocate higher weights to the features that are close to the person name being disambiguated. This is particularly true for list-based web pages where only the local words contain information about the given individual and the rest of the page may be about other individuals and something more general.

Another improvement we make is to explore the use of relative similarity measures for document clustering. Many of the current systems use the simple K-means ([8] and [11]) or single-link agglomerative hierarchical clustering ([3], [4], [5], and [12]) for organizing the search results into different clusters. Such methods require us to pre-determine the values of certain parameters (such as the number of clusters for K-means and the cut-off thresholds for separating different clusters in Single-link clustering), which hinder the discovery of natural groupings of the web pages. The advantages of the relative similarity measures as used in the Chameleon algorithm [7] are that we can identify clusters of different shapes, densities, and sizes, as well as scale up the implementation to handle a large number of documents.

For the rest of the paper, we describe the locality-based representation for feature vectors in section 2. Then, we introduce the Chameleon clustering algorithm along with a simplified relative similarity measure in section 3. After that, we discuss our experimental results for Web People Search 2008 in section 4, and end this paper with conclusions and future work in section 5.

2. LOCALITY-BASED DOCUMENT REPRESENTATION

Given a web page, we can try to extract values for certain attributes such as gender, birth-date, birth-place, and so on. Such values are useful for disambiguating a person name, since an individual is unlikely to have more than one birth-date or birth-place. Because of the importance of such attributes, Web People

WWW 2009, April 20-24, 2009, Madrid, Spain.

Search 2 creates a separate subtask for attribute extraction, which identifies a total of 18 different attributes that can potentially be used for person name disambiguation.

There is no doubt that these attribute values are helpful for any people search system. For some web pages, however, they are not always available, and in such cases, we still need to rely on certain keywords and their frequencies to distinguish between different kinds of documents. In fact, keywords and their frequencies are still commonly used for most text classification systems: Naïve Bayesian [9], Maximum Entropy Modeling [10] and Support Vector Machines [6].

We recognize the importance of extracting attribute values for person name disambiguation. Unfortunately, due to the time constraint, we are unable to incorporate such a mechanism in our current implementation. As a result, we focus exclusively on keywords and their frequencies and use the standard TF x IDF weights for document representation. Consequently, we can use the cosine measure to compute the similarity between any pair of documents, which forms the basis for any clustering algorithm.

However, as we examine the training data set for Web People Search 2, we notice that there are three different kinds of web pages: description-based, list-based, and mixed-mode documents. A description-based page often focuses on one individual and contains extended description about the person. A list-based page usually covers multiple individuals in a list or table form (e.g., lists of addresses, events, or references). For each individual, only the discussion that is physically close to the occurrences of his or her name is truly relevant; the discussion that is far away is either about another individual or something more general. A mixed-mode page is simply a combination of the description-based and list-based document: either a list is embedded in a long description or there are extended descriptions for each list item.

To capture and differentiate the keywords that are close to the occurrences of the person name being disambiguated, we introduce the notion of locality and locality-based TF x IDF weights for document representation. For each occurrence of a keyword, we first compute its minimum distance to all the occurrences of the given person name. After that, we adjust its count value according to the following formula:

$$\text{adjusted-count}(w_i) = \alpha + (1 - \alpha)\, \beta^{\text{min-distance}}$$

Here, both α and β are real numbers between [0, 1] and the "min-distance" is the minimum word distance between w_i and all occurrences of the given person name in a web page. When min-distance is 0, the adjusted-count will be 1.0, but when min-distance is approaching to infinity, the adjusted-count will be α. Thus, a count of 1.0 is the maximum value and a count of α is the minimum value. Normally, the adjusted-count is something in between the two extremes, as illustrated by the decreasing curve in Figure 1, where α is set to 0.2 and β is set to 0.9.

Based on the adjusted-count for each occurrence of a keyword, we can accumulate the adjusted-TF for all occurrences of the keyword, and then use the adjusted-TF x IDF weights for representing each document. Although we leave as future work the task of extracting the attribute values, the idea of locality and locality-based representation can be easily extended to attribute values so that the list-based pages can be accommodated.

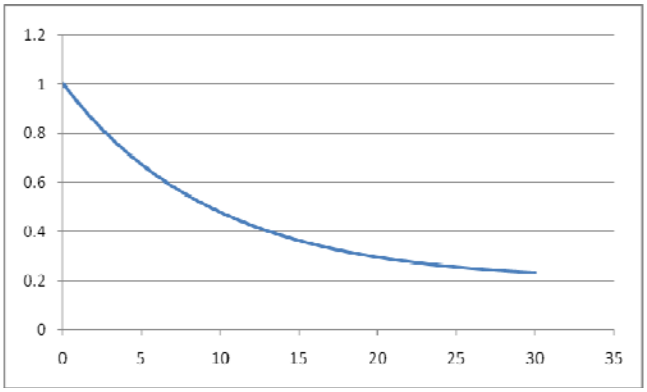


Figure 1. Differentiating contributions of different keywords based on locality.

3. RELATIVE SIMILARITY MEASURES FOR DOCUMENT CLUSTERING

Based on the feature vectors for each document and the related similarity measures, we can group documents into a set of clusters using a clustering algorithm. In the case of people search, we are hoping to organize all the web pages about one individual into one cluster so that different clusters provide a solution to the disambiguation of the same person name. A good clustering algorithm should help find the natural groupings of the documents, maximizing the connectivity within clusters but minimizing the connectivity between different clusters.

However, as mentioned in [7], the existing clustering algorithms, such as K-means and agglomerative hierarchical clustering algorithms using single-link or group-average, often require the results to fit into some static models rather than finding the natural groupings of the data. For example, we need to pre-determine the number of desired clusters for the K-means method. As a result, they tend to breakdown for the data that consists of clusters of different shapes, densities, and sizes.

In our implementation, we follow the Chameleon clustering process along with a simplification to relative similarity measures for merging clusters.

3.1 Chameleon Clustering Process

Chameleon is a three-step clustering process [7]. It first constructs a k-nearest neighbor graph for a collection of documents to be clustered. Then, it breaks the k-nearest neighbor graph into a set of small clusters, usually by repeatedly bisecting a graph so that the two sub-graphs more or less have the same number of nodes but the inter-connectivity between them is minimized. Finally, it re-builds a hierarchy of clusters by merging a pair of closest clusters at a time using relative similarity measures.

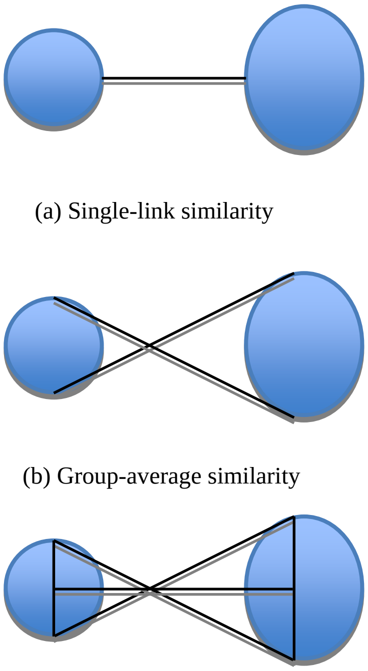
There are several advantages of the Chameleon clustering algorithm. First, a k-nearest neighbor graph only captures the strong links among a large two-dimensional similarity matrix and by controlling the size of k, we can keep the memory overhead to a reasonable level so that the method can be scaled up to handle a very large data set.

Second, we start with a set of small clusters for building a hierarchical structure. Unlike other agglomerative clustering methods such as single-link or group-average, we do not need to treat each document as the initial clusters and build a hierarchy of clusters from a scratch. Instead, we simply cut off the weak links in the k-nearest neighbor graph to get a set of sub-graphs, and if some sub-graphs are still too big, we then repeatedly bisect them into smaller sub-graphs.

Third, we use relative similarity measures for merging the nearest clusters. Compared with the absolute similarity measures such as single-link and group-average, relative similarity measures are more flexible and accurate in identifying clusters of different shapes, densities, and sizes. This should be well-suited for web people search, since there are considerably more pages for famous people than those ordinary people with the same name in the web space.

3.2 Chameleon Relative Similarity Measure

Given a pair of clusters, the single-link method uses the closest link between the two clusters as their similarity measure, while the group-average method uses the average of all the inter-links between the two clusters, as illustrated in figure 2.



(a) Single-link similarity

(b) Group-average similarity

(c) Chameleon's relative similarity

Figure 2. Illustrations of different similarity measures

The Chameleon measure is different from these absolute measures in that it uses both inter-connectivity and closeness between two clusters. Let $EC_{\{C_i,C_j\}}$ denotes all the edge cuts between two clusters C_i and C_j. Then, inter-connectivity $\left|EC_{\{C_i,C_j\}}\right|$ equals to the sum of all the edge weights (usually the cosine similarities) between two clusters, while closeness measures the average edge cut between the two clusters, denoted as $\bar{S}EC_{\{C_i,C_j\}}$. Similarly, Chameleon defines inner-cluster measures by bisecting each cluster into two sub-graphs and computing its inter-connectivity and closeness between the two sub-graphs. After that, the relative measures can be defined by dividing the between-cluster measures over the average of the two inner-cluster measures and we get a relative inter-connectivity (RI) and a relative closeness (RC) between two clusters:

$$RI(C_i,C_j)=\frac{\left|EC_{\{C_i,C_j\}}\right|}{(\left|EC_{Ci}\right|+\left|EC_{Cj}\right|)/2}$$

$$RC(C_i,C_j)=\frac{\bar{S}EC_{\{C_i,C_j\}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}EC_{C_i}+\frac{|C_j|}{|C_i|+|C_j|}\bar{S}EC_{C_j}}$$

Both RI and RC are important and are the keys for discovering clusters of different shapes, densities, and sizes. The two can be combined by a weighting factor α to get the final relative similarity between two clusters:

$$Sim(C_i,C_j)=RI(C_i,C_j)\times RC(C_i,C_j)^{\alpha}$$

Note that when α>1, we put more emphasis on RC, and when α<1, we give more weight on RI.

3.3 Simplified Relative Similarity Measure

Chameleon's relative similarity measure is flexible for data clustering. However, it needs to start with a set of small clusters, each of which has to contain a reasonable number of nodes in order to be bisected into two sub-graphs so that we can compute the relative measures for RI and RC. For web people search, some individuals may not have enough pages in the search results in order to form a meaningful small cluster for it. Consequently, it becomes infeasible to apply the Chameleon algorithm for web people search directly.

To reduce the need of a reasonable size for initial clusters, we propose a simplified relative similarity measure between two clusters. Instead of bisecting a cluster into two sub-graphs, we simply use the sum of the link weights within a cluster and its average as its inter-connectivity and closeness measures. Then, the relative RI and RC can be computed by dividing the between-cluster measures against the average of the two simplified inner-cluster measures. After that, the relative measures RI and RC can still be combined with α parameter to get a final relative similarity score. Note that such a simplified relative similarity measure also reduces the computational cost, since it is generally expensive in bisecting a graph, especially when it is large.

4. EXPERIMENTAL RESULTS

For Web People Search 2 Evaluations (WePS-2), we are given a total of 30 person names, each of which has about 100 web pages that contain the matched person name [2].

We use two different representations for the feature vectors: the standard TF x IDF weights and the locality-based TF x IDF weights. We further extend our implementation for the Chameleon clustering process so that we can support the relative similarity measures (Chameleon and Simplified Chameleon) as well as the absolute similarity measures between clusters (single-link, complete-link, and group-average). Once we obtain a hierarchy of clusters, we apply a flattening step to break the clusters with weak connections but at the same time merge the clusters with strong connections. This is done by trying different threshold values with the training data set and selecting the one that gives the best performance so that we can separate strong connections from the weak connections. The results are sets of flat clusters which can then be evaluated against the truth files provided for WePS-2 clustering task.

We send in five sets of results for the WePS-2 clustering task:

UGuelph_1: local_3_1.0@0.57

UGuelph_2: basic_0_1.0@0.13

UGuelph_3: local_3_2.0@0.54

UGuelph_4: basic_3_1.0@0.46

UGuelph_5: local_0_1.0@0.12

Here, "basic" refers to the standard TF x IDF weights, while "local" refers to the locality-based TF x IDF weights. The first number after the "basic" or "local" label indicates the similarity measure used for hierarchical clustering, with "0" for single-link and "3" for the simplified Chameleon. The second number after the "basic" or "local" label corresponds to the α parameter in the Chameleon formula to combine the relative measures RI and RC. Finally, the last number after the "@" symbol represents the cutoff threshold for flattening a hierarchy of clusters so that the results can be formally evaluated.

Listed in Table 1 and 2 are the evaluation numbers for the five sets of results we submitted.

Table 1: Results in BEP and BER measures

	BEP	BER	F_0.5	F_0.2
UGuelph_1	.54	.93	.63	.75
UGuelph_3	.54	.93	.63	.75
UGuelph_4	.53	.91	.60	.71
UGuelph_5	.55	.90	.60	.70
UGuelph_2	.51	.91	.57	.68

Table 2: Results in P and IP measures

	P	IP	F_0.5	F_0.2
UGuelph_1	.64	.95	.74	.84
UGuelph_3	.65	.96	.74	.84
UGuelph_4	.64	.95	.72	.82
UGuelph_5	.65	.94	.72	.81
UGuelph_2	.62	.94	.69	.79

As can be seen from the tables above, our system achieved better recall values (BER – Bcubed Recall) than the precision values (BEP – Bcubed Precision). For the combined F-measures, we achieved 0.63 for Fmeasure_0.5_BEP_BER, and 0.75 for Fmeasure_0.2_BEP_BER. The BCubed measures are introduced for WePS-2 in order to accommodate the overlapped clusters for some individuals. Using the metrics used in SemEval-2007, we achieved 0.74 for Fmeasure_0.5_P_IP and 0.84 for Fmeasure_0.2_P_IP. These results positioned us a little over the median rank among the participating systems for WePS-2: the clustering subtask.

Also seen from the tables is that the locality-based representation (UGuelph_1, UGuelph_3, and UGuelph_5) seem to perform better than the basic TF x IDF representation (UGuelph_2 and UGuelph_4). Furthermore, the simplified Chameleon measure helps further improve the clustering results (UGuelph_1, UGuelph_3, and UGuelph_4). Finally, for the simplified Chameleon measure, the α parameter does not seem to have a big impact on the results, since the results from UGuelph_1 and UGuelph_3 are basically the same.

5. CONCLUSIONS AND FUTURE WORK

We described in detail our implementation for Web People Search 2: the clustering subtask. We proposed a locality-based TF x IDF scheme for document representation, which can be easily extended for feature vectors once the attribute extraction is incorporated into our implementation. We also explored the use of relative similarity measures for web page clustering as used in the Chameleon algorithm along with a simplified relative similarity measure which is well-suited for clustering web people search results. Our system achieved the results of 0.63 for Fmeasure_0.5_BEP_BER and 0.75 for Fmeasure_0.2_BEP_BER, or 0.74 for Fmeasure_0.5_P_IP and 0.84 for Fmeasure_0.2_P_IP. This positions our implementation a little above the median rank among the all implementations for WePS-2 Clustering Subtask Evaluations.

Our implementation can be extended in several different ways. First, we would like to incorporate an attribute extraction mechanism into our system so that we can extract the relevant feature values about an individual and enhance our document representation. Second, we need to differentiate the contributions of different attributes and find effective ways of combining all the feature values, including the basic keywords and their frequencies, since the attribute values may not always be available for some web pages. Finally, we would like to explore the acquaintance information in a web page and use it to enhance the disambiguation of person name, since people are naturally associated with individuals they know and/or having similar interests.

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support from OCE (Ontario Centres of Excellence) and NSERC (Natural Sciences and Engineering Research Council of Canada).

6. REFERENCES

[1] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. Proceedings of the 4th International Workshop on Semantic Evalutions (SemEval-2007), 2007.

[2] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. WePS 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. Proceedings of the 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[3] Ying Chen and James Martin. CU-COMSEM: Exploring rich features for unsupervised web personal name disambiguation. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 125-128, 2007.

[4] Ergin Elmacioglu, Yee Fan Tan, Su Yan, Min-Yen Kan, and Dongwon Lee. PSNUS: Web people name disambiguation by simple clustering with rich features. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 268-271, 2007.

[5] Andrea Heyl and Günter Neumann. DFKI2: An information extraction based approach to people disambiguation. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 137-140, 2007.

[6] Thorsten Joachims. Learning to Classify Text Using Support Vector Machines. Kluwer Academic Publishers, 2002.

[7] G. Karypis, Eui-Hong Han, and V. Kumar. Chameleon: Hierarchical Clustering Using Dynamic Modeling. Computer, 32(8): 68-75, 1999.

[8] Els Lefever, Véronique Hoste, and Timur Fayruzov. AUG: A combined classification and clustering approach for web people disambiguation. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 105-108, 2007.

[9] David Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. Third Annual Symposium on Document Analysis and Information Retrieval, 81-93, 1994.

[10] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. IJCAI-99 Workshop on Machine Learning for Information Filtering, 61-67, 1999.

[11] Delip Rao, Nikesh Garera, and David Yarowsky. JHU1: An unsupervised approach to person name disambiguation using web snippets. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 199-202, 2007.

[12] Horacio Saggion. SHEF: Semantic tagging and summarization techniques applied to cross-document coreference. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 292-295, 2007.