Determine the Entity Number in Hierarchical Clustering for Web Personal Name Disambiguation

Jun Gong
Department of Information System
Beihang University
No.37 XueYuan Road HaiDian District, Beijing, China
jungong@ymail.com

Douglas W. Oard
College of Information Studies/UMIACS
University of Maryland
College Park, Maryland USA
oard@glue.umd.edu

ABSTRACT
An internet user is often frustrated by the ambiguous names in the web search results when the user is trying to find information about some person. Hierarchical clustering methods are often used to cluster the personal names referred to the same entities. As the correct number of the entities for a given personal name can not be accessed, we are required to determine the cut points in the dendrogram to gain high disambiguation accuracy. In this paper, we explore the appropriate cut points in hierarchical clustering for web personal name disambiguation. We first measure the similarity and density distribution of the search result pages, and then we propose an approach that combines the global distribution features and local features from cut points to explore the appropriate cut points. Finally, we perform experiments on real-world datasets and the results show that our method is effective.

Categories and Subject Descriptors
H.3.3 [Information Systems]: Information Search and Retrieval – *search process, clustering.* I.2.7 [Computing methodologies]: Natural Language Processing – *language models, language parsing and understanding, Text analysis.* I.7.5 [Computing methodologies]: Document and Tex Processing – *document analysis.*

General Terms
Algorithms, Performance, Languages, Experimentation

Keywords
Personal Name Disambiguation, Support Vector Machine, Clustering Algorithms.

1. INTRODUCTION
In the real-world, we often face a problem called name disambiguation, such as name variations, identical names, pseudonyms and name misspellings. Moreover, this problem is more serious in web due to the amount of web information. A internet user is often frustrated by the ambiguous names in the web search results when the user is trying to find information about some person. For example, if you search in Google for "James Martin", you will get about 47,300,000 result pages relevant to this personal name query; furthermore among the top 100 results you will find at least twenty different persons sharing this name. Additionally, it's a hard task for an internet user to distinguish so many ambiguous names and locate the one he really wants. In an ideal system the user would simply type a person name, and receive search results clustered according to the different people sharing that name.

In the WePS workshop 2007, many methods and rich features are presented [1]. Among these approaches, *Hierarchical Agglomerative Clustering* (HAC) algorithms [2] are widely used to cluster the personal names referred to the same entities. HAC views each web page containing personal names as a separate cluster and iteratively combines the most similar pair of clusters to form a cluster that replace the pair. After the dendrogram are constructed, the hierarchy is cut at some point to get a partition of disjoint clusters, and the pages in the same disjoint cluster are regarded to refer the same entity. As we do not know the correct number of the entities for a given personal name query, we are required to determine the cut points to gain high disambiguation accuracy. It is therefore crucial to develop methods that can efficiently detect the cut points that achieve the highest name disambiguation accuracy.

In this paper, we present an approach that combines the global dendrogram features and local features from the cut points to determine the entity number of these ambiguous names. In order to verify our methods, we implement single-link clustering and complete-link clustering algorithms in our experiments, and the results show that our methods perform well in both of the two clustering algorithms. In addition, we extract five different features to measure the similarity between the two web pages, and some features are firstly introduced in the WePS task.

2. Previous Work
In the WePS workshop 2007, hierarchical clustering methods combined with rich features achieved the highest scores [3][4][5] in all of the results. Most of these works are focused on feature extraction and integration, and few of them discuss how to determine the entity numbers or the appropriate cut points in hierarchical clustering of their experiments. On the other hand, common approaches to estimating the number of clusters such as cross-validation, penalized likelihood estimation, permutation tests, resampling, and finding the knee of an error curve [6] do not work very well in our task, because most of these methods only work well for clusters that are well separate and the inefficiency is their common drawback. In our study, we aim to explore the real entity numbers in the hierarchical clustering for web personal name disambiguation.

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

3. Methodology

In web personal name search results, we regard the top N result pages as our ambiguous objects. Here we adopt the policy of "one person per document" [7], and we assume that all mentions of the ambiguous personal name in one web page are referred to the same personal entity.

3.1 Feature Extraction and Page Similarity

Rich features were proposed in WePS 2007 to measure the page similarities in personal name search results. In our study, we follow the results of the previous researchers, and also make some extensions. Five features are extracted in our system and they are Token (T), Name Entities (NE), Full Personal Names matched (FPN), Hyperlinks in the page (L) and Page Structure (PS). Among the five features, T and NE have been widely used in the previous studies [3][4][5], and we do not discuss them anymore here. Now we review the new features we introduced in this pager.

Full Personal Name matched (FPN). We extract the personal full names that match the name query exactly, For example, we can locate "William A. Dickson", "William H Dickson", "William Henry Dickson", "William GLEN Dickson" and so on in different result pages for the given name query "William Dickson". In order to use this information, we take the average Jaro-Winkler distance [8] of first name, middle name and last name as the similarity between the two names.

Hyperlinks (L). If links in two web pages are pointing to the same URL, the two pages are more likely to refer the same person. We therefore extract links in the result pages and compare these links in the result pages. The similarity between the two pages is measured with Jaccard similarity coefficient [9].

Page Structure (PS). Some pages in the results are very large and mixed with different contents; hence we parse the HTML pages[1] in order to get the paragraphs or sentences that are relevant the personal name query. Additionally, we filter out the banners, navigation bars and advertisements using the HTML tree structure and content.

In order to integrate the five features in clustering, we give different weight to each feature and combine these features linearly.

3.2 Similarity and Density Distribution

For the result pages of a given name query, if we take each result page (p_i) as a node, and the similarity (s_{ij}) between two pages as the weighted edge, we will get an undirected graph (G).

$$G=(P,S),\ p_i \in P,\ s_{ij} \in S.$$

In this graph, the similarities are distributed unevenly. In order to describe this distribution feature, we define the density (d_i) for each result page.

$$d_i = \sum_j s_{ij} \Big/ (N-1)$$

If we take the similarity (S) and page's density (D) as the random variables, we will get different cumulative distribution functions for different personal name queries. Figure 1 illustrates the distribution differences between the name "Amanda Lentz" and "Cheng Niu".

[1]http://search.cpan.org/~petek/HTML-Tree-3.23/

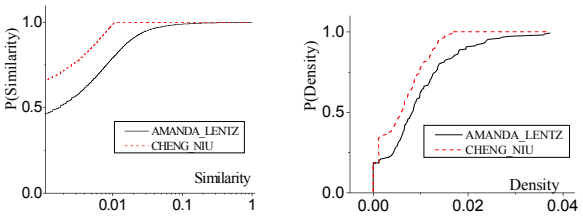

Figure 1. The similarity and density cumulative distributions for two personal names

In Figure 1, the two names share the similar curve shapes in both similarity and density distribution, but there is some distance between the two curves. This figure illustrates the global similarity and density distribution features of the result pages for a given name query. In next subsection, we will discuss how to combine the global distribution features with local features extracted at the cut points to enhance the personal name disambiguation accuracy.

3.3 Determining the Entity Number through Learning

In our application, hierarchical clustering views each web page containing personal names as a separate cluster and iteratively combines the most similar pair of clusters to form a cluster that replace the pair. After the dendrogram are constructed, the hierarchy is cut at some point to get a partition of disjoint clusters, and the pages in the same disjoint cluster are regarded to refer the same entity. As we do not know the correct number of the entities for a given personal name query, we are required to determine the cut point.

The problem of determining the entity number can be turned into a binary classification problem. For each cut point (C_i) in a dendrogram, we are required to make a decision whether or not to cut at this point according to its attributes. We use a decision table to illustrate this process.

Table 1. Decision table for exploring appropriate cut points

	Attributes		C_1	C_2	C_3	C_i
Conditions	Global	Mean(S)	0.01	0.01	0.04	…
		Stddev(S)	0.04	0.04	0.02	…
		Stddev(D)	0.07	0.08	0.04	…
	Local	Merging Similarity	0.87	0.11	0.05	…
		Cluster Number	148	23	5	…
Actions	Yes			√		…
	No		√		√	…

The attributes we use in the decision table are extracted both globally and locally. We take the mean of similarity, standard deviation of similarity and standard deviation of density as the global distribution attributes from whole result set. For the attributes from the local cut point, we focus on the similarity and cluster number when two clusters are combined at this point

In the training process, we calculate F-measure (F_i) at each cut point (C_i), and label the best F-measure as F_{best}. In order to make the train process more flexible, we define the acceptance range (r). If F_i is larger than $F_{best} * r$, we assume this cut point C_i is acceptable, otherwise is not. In the testing process, there may be multiple appropriate cut points for a given personal name query, and we pick up the one whose cluster number is closest to the mean cluster number of these candidate points.

Support Vector Machine (SVM) [10] and Maximum Entropy Modeling (ME) [11] are useful techniques in machine learning and achieve good results in many fields. Hence, we leverage the two techniques in our application separately to explore the appropriate cut points. For the SVM, we use the libsvm[2] in our experiments, and we use the Radial Basis Function (RBF) as the kernel function.

$$K(x_i, x_j)=\exp(-r \| x_i - x_j \|^2),\ r>0$$

Here, r is the kernel parameters. The kernel parameter r and penalty parameter C are tuned in our experiments following the instruction of the libsvm. For the Maximum Entropy Modeling, we use the classifier from Stanford Natural Language Processing Group directly[3].

4. Experiment

We use the WePS-1 dataset for learning and WePS-2 dataset for testing in our experiments. As the clustering algorithms play an important role in our application, hence we implement both single-link and complete-link clustering algorithms to verified our method.

4.1 Results with Fixed Threshold

We firstly cut the clustering dendrogram with fixed similarity threshold, which means that we give the different query results with the same threshold. We cautiously select five thresholds in our experiments, and the official results are shown as below.

Table 2. Clustering results using single-link with fixed threshold

	Similarity Threshold	BEP	BER	FMeasure_ 0.5
UMD_1	0.15	0.91	0.60	0.69
UMD_2	0.14	0.93	0.57	0.67
UMD_3	0.13	0.92	0.58	0.67
UMD_4	0.12	0.94	0.60	0.70
UMD_5	0.11	0.94	0.56	0.67

In the table, we can see that the BEP and PER are not well balanced. In our next experiment, we are trying to solve the problem.

4.2 Results with SVM

In order to measure the effectiveness of the method with learning, we define the *Closeness* as below.

Closeness= F-measure Reached / Best F-measure

Firstly, we try to scale how close our method can reach the best F-measure (a=0.5). Table 3 and Table 4 illustrate the results of the thirty name queries. In this experiment, we fixed the acceptance range at 0.4, and perform our method using SVM. The average closenesses for single-link and complete-link are 84.01% and 81.59%, which is pretty close to the best F-measure.

Table 3. Clustering results using complete-link algorithm with SVM

	Best F_0.5	F_0.5	Closeness
AMANDA_LENTZ	1	1	1
BENJAMIN_SNYDER	0.7904	0.6743	0.853
BERTRAM_BROOKER	0.7742	0.3093	0.4
CHENG_NIU	0.9704	0.8997	0.927
DAVID_TUA	0.6973	0.6973	1
DAVID_WEIR	0.7377	0.6034	0.818
EMILY_BENDER	0.9797	0.9797	1
FRANZ_MASEREEL	0.8127	0.3481	0.428
GIDEON_MANN	0.9686	0.9488	0.98
HAO_ZHANG	0.8077	0.6343	0.785
HELEN_THOMAS	0.679	0.6703	0.987
HERB_RITTS	0.5055	0.3671	0.726
HUI_FANG	0.8865	0.7048	0.795
IVAN_TITOV	0.7717	0.5655	0.733
JAMES_PATTERSON	0.9621	0.5779	0.601
JANELLE_LEE	0.8837	0.8837	1
JASON_HART	0.7768	0.6389	0.822
JONATHAN_SHAW	0.9333	0.7832	0.839
JUDITH_SCHWARTZ	1	0.9481	0.948
LOUIS_LOWE	1	0.9475	0.948
MIKE_ROBERTSON	0.7425	0.6722	0.905
MIRELLA_LAPATA	0.9698	0.833	0.859
NICHOLAS_MAW	0.928	0.4388	0.473
OTIS_LEE	0.9895	0.9681	0.978
RITA_FISHER	0.9888	0.5628	0.569
SHARON_CUMMINGS	0.9103	0.8879	0.975
SUSAN_JONES	0.8564	0.5416	0.632
TAMER_ELSAYED	0.8312	0.7815	0.94
THEODORE_SMITH	0.7991	0.7006	0.877
TOM_LINTON	0.8387	0.5698	0.679
AVERAGE	**0.85972**	**0.7046**	**0.8159**

Table 4. Clustering results using single-link algorithm with SVM

	Best F_0.5	F_0.5	Closeness
AMANDA_LENTZ	0.6212	0.6146	0.989
BENJAMIN_SNYDER	0.5385	0.5385	1
BERTRAM_BROOKER	1	0.8349	0.835
CHENG_NIU	0.9602	0.8824	0.919
DAVID_TUA	1	0.7999	0.8
DAVID_WEIR	0.6478	0.6187	0.955
EMILY_BENDER	0.8359	0.7688	0.92
FRANZ_MASEREEL	0.7524	0.6245	0.83
GIDEON_MANN	0.9443	0.7074	0.749
HAO_ZHANG	0.7639	0.6882	0.901
HELEN_THOMAS	0.9797	0.795	0.811

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[3] http://nlp.stanford.edu/downloads/classifier.shtml

HERB_RITTS	0.9895	0.8898	0.899
HUI_FANG	0.7569	0.7179	0.948
IVAN_TITOV	0.9795	0.8483	0.866
JAMES_PATTERSON	0.9488	0.9038	0.953
JANELLE_LEE	0.9167	0.7801	0.851
JASON_HART	0.6927	0.6351	0.917
JONATHAN_SHAW	0.6988	0.6375	0.912
JUDITH_SCHWARTZ	0.6148	0.4935	0.803
LOUIS_LOWE	0.7711	0.5573	0.723
MIKE_ROBERTSON	0.8385	0.8319	0.992
MIRELLA_LAPATA	1	0.6678	0.668
NICHOLAS_MAW	1	0.9156	0.916
OTIS_LEE	0.7011	0.564	0.804
RITA_FISHER	0.8181	0.5293	0.647
SHARON_CUMMINGS	0.9012	0.8556	0.949
SUSAN_JONES	0.5641	0.3432	0.608
TAMER_ELSAYED	0.6221	0.4407	0.708
THEODORE_SMITH	0.6486	0.2593	0.4
TOM_LINTON	0.7653	0.7106	0.929
AVERAGE	**0.8090**	**0.6818**	**0.840**

4.3 Sensitivity to Acceptance Range

Next we use Figure 2 to illustrate the sensitivity of the average closeness to the acceptance range (r). In the figure, the acceptance range is not fixed, but variates from 0 to 1.

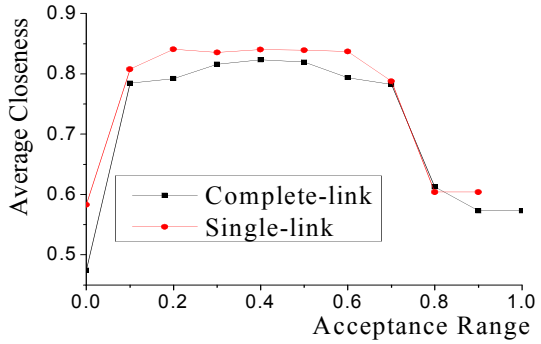

Figure 2. The average closeness variates according to the acceptance rate using SVM

In the Figure 2, the two average closeness curves keep high closeness when acceptance range ranging from 0.1 to 0.7, and drop down when the rate is out of the range. The curves show us that the average closeness is insensitive to the acceptance rate for both single-link and complete-link clustering algorithms. Practically, we can set the acceptance range in a wide rang in the training process and get high closeness to best F-measure in the testing process.

4.4 Results using Maximum Entropy Modeling

We also perform the experiments with Maximum Entropy Modeling (ME) and Figure 3 illustrates the result.

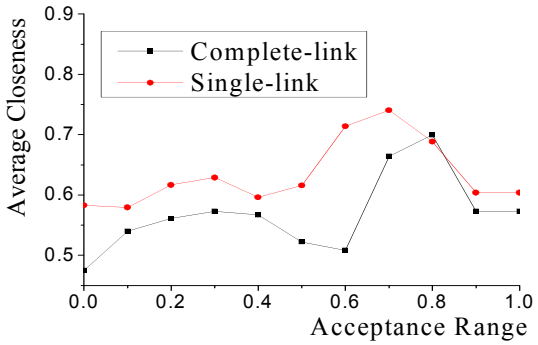

Figure 3. The average closeness variates according to the acceptance rate using Maximum Entropy Modeling

The max average closeness in Figure 3 is 0.74, which is much lower than the closeness using SVM. Moreover, the two curves are very sensitive to the acceptance range, and they only achieve higher closeness when the acceptance range is near 0.7.

5. Conclusion

In this paper, we present an approach that can efficiently explore the appropriate cut points in the hierarchical clustering for web personal name disambiguation. Our work is primarily concerned with the global distribution features and local features from the cut points in the dendrogram. We also integrate Support Vector Machine and Maximum Entropy Modeling separately into our approach for learning in the training process. In order to verify our methods, we implement single-link clustering and complete-link clustering algorithms in experiments. The results show that our method combined with SVM performs quite well with both of the two clustering algorithms, which not only achieves high closeness to the best F-measure but also keeps the high accuracy steadily when acceptance range variates in a wide range.

Our research can be extended in different applications. For example, we only use five features in the decision table, and new feature can be added to our model in order to represent additional global or local features in other applications.

6. REFERENCES

[1] J. Artiles, J. Gonzalo, and S. Sekine. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. Proceedings of Semeval, 2007.

[2] C. S. Manning and H. Schutze. Foundations of Statistical Natural Language Processing, The MIT Press, 500-512

[3] E. Elmacioglu, Y.F. Tan, S. Yan, M.Y. Kan and D. Lee. PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features. Proceedings of Semeval, 2007.

[4] Y. Chen and J. H. Martin. CU-COMSEM: Exploring Rich Features for Unsupervised Web Personal Name Disambiguation. Proceedings of Semeval, 2007.

[5] E. Lefever, V. Hoste and T. Fayruzov. AUG: A combined classification and clustering approach for web people disambiguation. Proceedings of Semeval, 2007.

[6] S. Salvador and P. Chan. Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation

Algorithms Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI) 2004.

[7] A. Bagga and B. Baldwin. Entity–based Cross–document Co–referencing Using the Vector Space Model. In 17th COLING. 1998.

[8] Winkler, W. E. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication 1999/04.

[9] P. N. Tan, M. Steinbach and V. Kumar. Introduction to Data Mining 2005.

[10] Cristianini, N. and Shawe T. J. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press. 2000.

[11] A. Berger, S. D. Pietra, and V. D. Pietra. A maximum entropy approach to natural language processing. Computational Linguistics, 22 (1):39-71, 1996.