PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features

Ergin Elmacioglu[1] **Yee Fan Tan**[2] **Su Yan**[1] **Min-Yen Kan**[2] **Dongwon Lee**[1]
[1]The Pennsylvania State University, USA
[2]National University of Singapore, Singapore
{ergin,syan,dongwon}@psu.edu, {tanyeefa,kanmy}@comp.nus.edu.sg

Abstract

We describe about the system description of the PSNUS team for the SemEval-2007 Web People Search Task. The system is based on the clustering of the web pages by using a variety of features extracted and generated from the data provided. This system achieves $F_{\alpha=0.5} = 0.75$ and $F_{\alpha=0.2} = 0.78$ for the final test data set of the task.

1 Introduction

We consider the problem of disambiguating person names in a Web searching scenario as described by the Web People Search Task in SemEval 2007 (Artiles et al., 2007). Here, the system receives as input a set of web pages retrieved from a search engine using a given person name as a query. The goal is to determine how many different people are represented for that name in the input web pages, and correctly assign each namesake to its corresponding subset of web pages.

There are many challenges towards an effective solution. We are to correctly estimate the number of namesakes for a given person name and group documents referring to the same individual. Moreover, the information sources to be processed are unstructured web pages and there is no certain way of correctly establishing a relation between any two web pages belonging to the same or different individuals.

We have taken several approaches to analyze different sources of information provided with the input data, and also compared strategies to combine these individual features together. The configuration that achieved the best performance (which were submitted for our run) used a single named entity feature as input to clustering. In the remainder of this paper, we first describe our system in terms of the clustering approach used and alternative features investigated. We then analyze the results on the training set before concluding the paper.

2 Clustering Algorithm

Clustering is the key part for such a task. We have chosen to view the problem as an unsupervised hard clustering problem. First, we view the problem as *unsupervised*, using the training data for parameter validation, to optimally tune the parameters in the clustering algorithm. Secondly, we observed that the majority of the input pages reference a single individual, although there are a few that reference multiple individuals sharing the same name. Hence, we view the problem as *hard* clustering, assigning input pages to exactly one individual, so that the produced clusters do not overlap.

Hard clustering algorithms can be classified as either partitive or hierarchical. Agglomerative hierarchical clustering generates a series of nested clusters by merging simple clusters into larger ones, while partitive methods try to find a pre-specified number of clusters that best capture the data. As the correct number of clusters is not given *a priori*, we chose a method from the second group. We use the *Hierarchical Agglomerative Clustering* (HAC) algorithm (Jain et al., 1999) for all experiments reported in this paper. HAC views each input web page as a separate cluster and iteratively combines the most similar pair of clusters to form a new cluster that re-

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 268–271,
Prague, June 2007. ©2007 Association for Computational Linguistics

places the pair.

3 Features

As input to the clustering, we consider several different representations of the input documents. Each representation views the input web pages as a vector of features. HAC then computes the cosine similarity between the feature vectors for each pair of clusters to determine which clusters to merge. We now review the inventory of features studied in our work.

Tokens (T). Identical to the task baseline by (Artiles et al., 2005), we stemmed the words in the web pages using the Porter stemmer (Porter, 1980), to conflate semantically similar English words with the stem. Each stemmed word is considered to be a feature and weighted by its Term Frequency $\times$ Inverse Document Frequency (TF$\times$IDF).

Named Entities (NE). We extract the named entities from the web pages using the Stanford Named Entity Recognizer (Finkel et al., 2005). This tagger identifies and labels names of places, organizations and people in the input. Each named entity token is treated as a separate feature, again weighted by TF$\times$IDF. We do not perform stemming for NE features.

We also consider a more target-centric form of the NE feature, motivated by the observation that person names can be differentiated using their middle names or titles. We first discard all named entities that do not contain any token of the search target, and then discard any token from the remaining named entities that appears in the search target. The remaining tokens are then used as features, and weighted by their TF$\times$IDF. For example, for the search target "Edward Fox", the features generated from the name "Edward Charles Morrice Fox" are "Charles" and "Morrice". We call this variation NE targeted (NE-T).

Hostnames and domains (H and D). If two web pages have links pointing to the exact same URL, then there is a good chance that these two web pages refer the same person. However, we find such exact matches of URLs are rare, so we relax the condition and consider their hostnames or domain names instead. For example, the URL http://portal.acm.org/guide.cfm has hostname portal.acm.org and domain name acm.org. As such, for each web page, we can extract the list of hostnames from the links in this page.

We observe that some host/domain names serve as more discriminative evidence than others (e.g., a link to a university homepage is more telling than a link to the list of publications page of Google Scholar when disambiguating computer science scholars). To model this, we weight each host/domain name by its IDF. Note that we do not use TF as web pages often contain multiple internal links in the form of menus or navigation bars. Using IDF and cosine similarity has been proven effective for disambiguating bibliographic citation records sharing a common author name (Tan et al., 2006).

We also considered a variant where we include the URL of the input web page itself as a "link". We tried this variation only with hostnames, calling this Host with Self URL (H-S).

Page URLs (U). Uniform resource locations (URLs) themselves contain a rich amount of information. For example, the URL http://www.cs.ualberta.ca/~lindek/ itself suggests a home page of "lindek" in the Computer Science department, University of Alberta, Canada.

We used the MeURLin system (Kan and Nguyen Thi, 2005) to segment the URL of each web page into tokens as well as to generate additional features. These features include (a) segmentation of tokens such as "www.allposters.com" to "www", "all", "posters" and "com"; (b) the parts in the URL where the tokens occur, e.g., protocol, domain name, and directory paths; (c) length of the tokens; (d) orthographic features; (e) sequential n-grams; and (f) sequential bigrams. As each of these features can be seen as a "token", the output of the MeURLin segmenter for a web page can be seen as a "document", and hence it is possible to compute the TF$\times$IDF cosine similarity between two such documents.

3.1 Feature Combination

The features described above represent largely orthogonal sources of information in the input: input content, hyperlinks, and source location. We hypothesize that by combining these different features we can obtain better performance. To combine these features for use with HAC, we consider simply concatenating individual feature vectors together to cre-

ate a single feature vector, and compute cosine similarity. We used this method in two configurations: namely, (T + NE + H-S), (T + D + NE + NE-T + U).

We also tried using the maximum and average component-wise similarities of individual features. (*max*(NE, H-S)) uses the maximum value of the Named Entity and Host with Self features. For the (*avg*(T, H-S)) and (*avg*(T, D, NE, NE-T, U)) runs, we compute the average similarity over the two and five sets of individual features, respectively.

4 Results

We present the clustering performances of the various methods in our system based on the different features that we extracted. Each experiment uses HAC with single linkage clustering. Since the number of clusters is not known, when to terminate the agglomeration process is a crucial point and significantly affects the quality of the clustering result. We empirically determine the best similarity thresholds to be 0.1 and 0.2 for all the experiments on the three different data sets provided. We found that larger values for these data sets do not allow the HAC algorithm to create enough clustering hierarchy by causing it to terminate early, and therefore result in many small clusters increasing purity but dramatically suffering from inverse purity performance.

Table 1 shows the results of our experiments on the training data sets (ECDL, Wikipedia and Census). Two different evaluation measures are reported as described by the task: $F_{\alpha=0.5}$ is a harmonic mean of purity and inverse purity of the clustering result, and $F_{\alpha=0.2}$ is a version of F that gives more importance to inverse purity (Artiles et al., 2007).

Among the individual features, Tokens and Named Entity features consistently show close to best performance for all training data sets. In most cases, NE is better than Tokens because some web pages contain lots of irrelevant text for this task (e.g., headers and footers, menus etc). Also, we found that the NEs have far more discriminative power than most other tokens in determining similarity between web pages. The NE variation, NE targeted, performs worse among the token based methods. Although NE targeted aims for highly precise disambiguation, it seems that it throws away too much information so that inverse purity is very much reduced. The other NEs, such as locations and organizations are also very helpful for this task. For example, the organization may indicate the affiliation of a particular name. This explains the superiority of NE over NE targeted for all three data sets.

Among the link based features, Domain gives better performance over Host as it leads to better inverse purity. The reason is that there are usually many pages on different hosts from a single domain for a given name (e.g., the web pages belonging to a researcher from university domain). This greatly helps in resolving the name while results in a slight drop in purity. Using a web page's URL itself in the features Host+Self and Domain+Self shows a larger increase in inverse purity at a smaller decrease in purity, hence these have improved F-measure in comparison to Domain and Host. Not surprisingly, these link based features perform very well for the ECDL data set, compared to the other two. A significant portion of the people in the ECDL data set are most likely present-day computer scientists, likely having extensive an web presence, which makes the task much easier. Although the other two data sets may have popular people with many web pages, their web presence are usually created by others and often scatter across many domains with little hyperlinkage between them. This explains why our link based methods are not very effective for such data sets.

Our final individual feature URL performs worst among all. Although highly precise, its resulting inverse purity is poor. While the features generated by MeURLin do improve the performance over pure host name and domain on the page URLs, its incorporation in a richer feature set does not lead to better results, as the other features which have richer information to process.

Each of the individual features has different degree of discriminative power in many different cases. By combining them, we expect to get better performance than individually. However, we do not obtain significant improvement in any of the data sets. Furthermore, in the Census data set, the combined features fail to outperform the individual NE and Tokens features. The relatively poor performance of the remaining features also degrades the performance of Tokens and NE when combined.

Considering the performances using the harmonic mean, we do not see any clear winner in all of three

Feature	ECDL		Wikipedia		Census	
	$F_{\alpha=0.5}$	$F_{\alpha=0.2}$	$F_{\alpha=0.5}$	$F_{\alpha=0.2}$	$F_{\alpha=0.5}$	$F_{\alpha=0.2}$
Tokens (T)	.72 / .77	.83 / .84	.72 / **.76**	.85 / .84	**.82 / .84**	.88 / .86
Named Entities (NE)	.75 / **.80**	.84 / .79	.75 / **.77**	.85 / .78	**.89 / .78**	.89 / .73
NE targeted (NE-T)	.54 / .55	.49 / .47	.66 / .64	.60 / .57	.64 / .64	.57 / .58
Host (H)	.72 / .57	.64 / .48	.67 / .51	.58 / .41	.67 / .63	.59 / .55
Host + Self (H-S)	.73 / .59	.66 / .49	.68 / .54	.60 / .43	.68 / .63	.60 / .56
Domain (D)	**.78** / .69	.72 / .60	.71 / .59	.66 / .50	.69 / .65	.61 / .58
Domain + Self (D-S)	**.79** / .70	.74 / .61	.72 / .62	.67 / .52	.70 / .66	.62 / .59
URL (U)	.50 / .43	.43 / .35	.56 / .42	.50 / .33	.64 / .58	.56 / .51
(T + NE + H-S)	.71 / .77	.83 / .83	.72 / **.76**	.85 / .83	.65 / .67	.78 / .76
(T + D + NE + NE-T + U)	.72 / .76	.83 / .80	.72 / **.77**	.84 / .83	.66 / .66	.78 / .74
(*max*(NE, H-S))	.74 / **.80**	.84 / .82	.74 / **.77**	.86 / .82	.71 / .66	.80 / .70
(*avg*(T, H-S))	.77 / **.81**	.86 / .76	.75 / **.77**	.86 / .76	.70 / .64	.80 / .67
(*avg*(T, D, NE, NE-T, U))	**.78** / .77	.86 / .73	.75 / **.78**	.86 / .76	.69 / .61	.77 / .62

Table 1: Experimental results for each training data set of the task: ECDL, Wikipedia and Census. Each experiment uses single link HAC with the similarity threshold values of 0.1 / 0.2. Best $F_{\alpha=0.5}$ performances are shown in bold.

training data sets. In addition, the method showing the best performance does not result in a win with a large margin in each data set. Relatively complicated methods do not always perform better over simpler, single featured based methods on all training data sets. Considering the results and Occam's razor (Thorburn, 1915), we conclude that a simple method should most likely work relatively well in many other different settings as well. Therefore, we selected the method based on the individual NE feature with the similarity threshold value of 0.2 for the final test submission run. We are able to achieve the following results for this submission run: purity = 0.73, inverse purity = 0.82, $F_{\alpha=0.5} = 0.75$, $F_{\alpha=0.2} = 0.78$.

5 Conclusion

We described our PSNUS system that disambiguates people mentions in web pages returned by a web search scenario, as defined in the inaugural Web People Search Task. As such, we mainly focus on extracting various kinds of information from web pages and utilizing them in the similarity computation of the clustering algorithm. The experimental results show that a simple Hierarchical Agglomerative Clustering approach using a single named entity feature seems promising as a robust solution for the various datasets.

References

Javier Artiles, Julio Gonzalo, and Felisa Verdejo. 2005. A testbed for people searching strategies in the WWW. In *ACM SIGIR*, pages 569–570, August.

Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2007. The SemEval-2007 WePS evaluation: Establishing a benchmark for the Web People Search Task. In *SemEval 2007, ACL*, June.

Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pages 363–370, June.

Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September.

Min-Yen Kan and Hoang Oanh Nguyen Thi. 2005. Fast webpage classification using URL features. In *CIKM*, pages 325–326, October/November.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137, July.

Yee Fan Tan, Min-Yen Kan, and Dongwon Lee. 2006. Search engine driven author disambiguation. In *ACM/IEEE JCDL*, pages 314–315, June.

William M. Thorburn. 1915. Occam's razor. *Mind*, 24:287–288.