Resolving Person Names in Web People Search[*]

Krisztian Balog and Leif Azzopardi and Maarten de Rijke

Abstract Disambiguating person names in a set of documents (such as a set of web pages returned in response to a person name) is a key task for the presentation of results and the automatic profiling of experts. With largely unstructured document and an unknown number of people with the same name the problem presents many difficulties and challenges. This chapter treats the task of person name disambiguation as a document clustering problem, where it is assumed that the documents represent particular people. This leads to the person cluster hypothesis, which states that similar documents tend to represent the same person. Single Pass Clustering, k-Means Clustering, Agglomerative Clustering and Probabilistic Latent Semantic Analysis are employed and empirically evaluated in this context. On the SemEval 2007 Web People Search it is shown that the person cluster hypothesis holds reasonably well and that the Single Pass Clustering and Agglomerative Clustering methods provide best performance.

1 Introduction

A field of growing importance and popularity is the profiling and searching of people. For instance, searching for expertise within an organization is a rapidly growing research area (also known as expert finding), and its importance is underlined by the introduction of an expert finding task at TREC in 2005 [10]. However, there are

Krisztian Balog
ISLA, University of Amsterdam, e-mail: kbalog@science.uva.nl

Leif Azzopardi
DCS, University of Glasgow, e-mail: leif@dcs.gla.ac.uk

Maarten de Rijke
ISLA, University of Amsterdam, e-mail: mdr@science.uva.nl

[*] This chapter is a revised and expanded version of [7].

many other related people search tasks (such as entity extraction, building descriptions of expertise, creating biographies, identifying social networks, etc). A more general task that helps facilitate such tasks is, what we call people-document associations [5, 6]. This is the task of associating documents with particular people. For instance, within an organization in order to build profiles of employees the collection of documents needs to processed for such associations.

One particular case of this people-document association task is referred to as personal name resolution [26, 27, 23] (also referred to as personal name disambiguation/discrimination [9, 22], and cross-document co-reference [4, 12]). The task is as follows: given a set of documents all of which refer to a particular person name but not necessarily a single individual (usually called *referent*), identify which documents are associated with each referent by that name. Recently, a test collection has been developed [3] to study this problem in a web setting; the scenario is this: given a list of documents retrieved by a web search engine using a person's name as a query, group documents that are associated to the same referent. This is a particularly relevant task because searching for people is one of the most popular types of web searches (around 5–10% of searches contain person names [27]). Given the popularity of people names in web queries, the problem of ambiguous person names is encountered frequently as a person name may have hundreds of distinct referents. Indeed, according to U.S. Census Bureau figures approximately 90,000 different names are shared by around 100 million people (as cited by [2]). On the web, a query for a common name often yields thousands of pages referring to different namesakes [9]. Grouping the documents together by referent has been shown to be particularly useful in this scenario as a means of reducing the burden on the user to sort through the results [27].

In this chapter, we focus on the task of personal name resolution, a problem that has generally been considered as a clustering task: cluster the extracted representations of referents from the source documents so that each cluster contains all the documents associated with each referent. Essentially all work on the personal name resolution task has framed the problem in this way. However, we consider the problem from an Information Retrieval point of view, in the context of the cluster hypothesis [16]. The cluster hypothesis states that similar documents tend to be relevant to the same request. Re-stated in the context of the personal name resolution task, similar documents tend to represent the same person (referent). And thus, the task is reduced to document clustering.

Here, we explicitly examine the "person cluster hypothesis," making no assumptions about the underlying documents, i.e., their structure, format, style, type, and so forth (unlike the bulk of past work). While we recognize that considering the semantic attributes and features within documents can help in the disambiguation of names, it is the purpose of this chapter to examine the extent to which the hypothesis holds under the most general conditions using only the distribution of terms in a document as features. To this end, we consider several clustering algorithms: Single Pass, k-Means, and Agglomerative Clustering along with Probabilistic Latent Semantic Analysis, where we address a number of research questions based on the assumption of the "person cluster hypothesis." Since, under this view the person name resolution task boils down to clustering the document space where each cluster is assumed to be a particular person (referent), there is an obvious limitation. If the same person is described or involved in very disparate ways or things, then similarity-based methods will suffer. But, how much of a limitation is this? More generally, how good are clustering techniques for this task? And to what extent does the assumption/hypothesis hold?

In addition to these high-level questions, we also have a set of more low level issues that we aim to make progress on: What factors affect performance? How stable is the performance? When is the best performance obtained? And, what is the best number of clusters to use? Also, we are interested in more contrastive and reflective questions: Is term-based clustering better than semantic-based clustering, or vice versa? And, how can we improve the current methods?

The remainder of the chapter is organized as follows. In Section 2 we review related work. Then, in Section 3, we discuss ways of modelling the personal name disambiguation problem. Section 4 is devoted to a discussion of our evaluation platform, and we present the results of our experimental evaluation in Section 5. Finally, we conclude by zooming out to discuss our more general research questions surrounding the person cluster hypothesis in Section 6.

2 Related work

The task of personal name resolution has been considered in many different ways; as (personal) name disambiguation, cross document co-reference and name resolution.

Name discrimination or name disambiguation is similar to word sense discrimination and generally relies upon the contextual hypothesis [21]: words with similar meaning are often used in similar contexts. Importantly, in word sense disambiguation the number of possible senses are known and limited to around 2–20; moreover, they are typically all known *a priori*—in name disambiguation the situation can be considerably more difficult as the numbers quoted in Section 1 suggest.

Cross document co-reference refers to the situation where an entity such as a person, place, event, etc. is discussed across a number of source documents [1]: if there are two instances of the same name from different documents, determine whether they refer to the same individual or not [9]. Essentially, cross document co-reference and personal name resolution/disambiguation are two sides of the same coin, where cross document personal name resolution is the process of identifying whether or not a person name mentioned in different documents refers to the same individual [23]. The problem can be broken down into two distinct sub-problems resulting from the types of ambiguity that manifest themselves in resolving person names [26]:

- multi-referent ambiguity: there are many people that share the same name; and
- multi-morphic ambiguity: one name may be referred to in different forms.

Past work has largely concentrated on the former problem, which has been addressed by clustering different types of representations extracted from the documents using different clustering techniques [4, 20, 12, 11, 23]. Different methods have been used to represent documents that mention a candidate, including snippets, text around the person name, entire documents, extracted phrases, etc. For instance, Bagga and Baldwin [4] first produce a summary of each person within each document (local person resolution). This summary is produced by extracting the text surrounding the person's name, which forms a bag of words representation. These, then, are clustered, using the cosine distance to determine similarity. Gooi and Allan [12] try a similar approach using snippets and perform agglomerative clustering with different similarity measures (cosine, KL-divergence). A possible criticism of such approaches is that the simplicity of the representation may not provide a rich enough representation of the person as the semantic relationships present within the document are ignored. However, in Information Retrieval, using a bag of words representation is common practice, because it is simple and effective. And it is a very powerful representation because it makes no specific assumptions about the underlying document structure and the content that it contains. It is more likely that the sparseness of the representations in the aforementioned techniques is more problematic.

An alternative approach that makes specific assumptions about the data was pursued by Mann and Yarowsky [20] who build a profile from each document based on learned and hand-coded patterns which are designed to extract (where present) the birth year, occupation, birth location, spouse, nationality, etc. Documents are matched based on matching the extracted factoids. A similar approach is taken by Phan et al. [23] who first create personal summaries consisting of a series of sentences; each summary is assigned a semantic label (such as *birthdate*, *nationality*, *parent*, etc); corresponding facts of each personal summary are then clustered using a notion of semantic similarity that is based on the relatedness of words. It should be noted that the approaches just outlined are limited as they make very strong assumptions about the data—which in web search, can not also be met or guaranteed.

Fleischman and Hovy [11] use a maximum entropy classifier trained on the ACL data set to give the probability that two names refer to the same referent. This technique requires large amounts of training data which are usually not available in practical settings.

Another approach which uses an alternative representation and approach to the problem of personal name resolution is based on social networks and co-citations to group/cluster the documents. Bekkerman and McCallum [8] use the link structure in web pages as a way to disambiguate the referents, while Malin [19] use actor co-citations within the Internet Movie DB.

Several semantics-based approaches have been proposed in the literature. E.g., Pedersen et al. [22] propose a method based on clustering using second-order context vectors derived from singular value decomposition (SVD) on a bigram-document co-occurrence matrix. And Al-Kamha and Embley [1] study combinations of three different representation methods—attribute (factoid) based representations like those used in [20, 23], link/citation-based, and content-based.

In this chapter, we use a standard IR representation of each document (i.e., bag of words) because we want to examine the person cluster hypothesis and make as few assumptions about the data as possible. Then, we examine this hypothesis using multiple clustering approaches, ranging from naive, but intuitive, methods such as single pass clustering, agglomerative clustering, k-means clustering, that focuses on term similarity, to a more sophisticated approach, Probabilistic Latent Semantic Analysis (similar to performing SVD) which focuses on semantic similarity. Since our focus is on evaluating the person clustering hypothesis in a very general setting, we have selected these clustering methods because they are representative of the types already tried. For instance, Artiles et al. [2] use a similar representation of documents with agglomerative clustering technique to obtain a baseline for a pilot test collection for this task. However, our work differs because we focus on exploring how document clustering performs for this task.

While there has been growing interest in studying the person name disambiguation task, past work has used different test collections with significantly different characteristics (i.e., web pages or Internet Movie DB data or journal publications), which makes it hard to compare previous approaches. An important recent development has been the introduction of a common and publicly available test collection for testing personal name resolution [3]. Consequently, this is one of the first studies conducted of this nature using such a resource (see Section 4 for details).

3 Data Modeling and Clustering Approaches

In this section we describe the clustering approaches that we shall use in order to evaluate the person clustering hypothesis. Before doing so, it is important to explicitly state the assumptions we have about the data, which will allow us to contextualize how well the clustering methods fit the task of resolving person names. The main modeling assumptions engaged along with the person cluster hypothesis are as follows:

1. One document is associated with one referent. While, this may not always be the case in practice, i.e., a page might contain several senses of the same personal name, there are only few instances of this within the test collection. (Note: this is a simplifying assumption often employed.)
2. The distribution of documents assigned to referents follows a power law, i.e., many referents have few documents associated with them, while few referents have many documents associated with them.
3. Every document refers to a distinct person sense, unless there is evidence to the contrary.
4. The number of distinct person senses is not known *a priori*. However, the number of possible person senses is limited by the number of documents available as a result of assumption (1) and (3).

5. The documents are assumed to be textual, but unstructured in nature with no predefined format, i.e., there are no guarantees about the format or structure within the documents.

Now, given these assumptions about the data, we evaluate how well the person cluster hypothesis holds under these conditions using four different clustering methods: Single Pass, k-Means and Agglomerative clustering along with Probabilistic Latent Semantic Analysis. The first three methods provide different variations of traditional clustering methods (varying in terms of efficiency and quality). These methods also explicitly rely on the documents associated to a particular referent sharing common terms to describe the individual. We also contrast these clustering methods with PLSA, which does not have such an explicit reliance on the terms because transitive connections between terms can be identified through the latent space, although sharing common terms would certainly improve the method's effectiveness. Each of the chosen methods are described in more detail below before we perform the evaluation.

3.1 Single Pass Clustering

The first and simplest clustering method we employed is single pass clustering (SPC) [14] to automatically assign pages to clusters. SPC provides not only an efficient clustering algorithm, but also mimics a reasonable heuristic that a user might employ (i.e., start at the top and work down the list associating documents to different person senses). SPC does exactly this: each document is considered in turn starting with the top ranked document, if a cluster representing that person already exists, then the document is assigned to that cluster, otherwise the document is assigned to a new cluster, to represent the new person sense. In fact, this is very similar to the process taken by the annotators of the collection we use for evaluation; see [3] and Section 4. Also, since web search results are often ranked proportional to the number of in-links which represents the "popularity" of the page, it is reasonable to assume that the most dominant (popular) senses of the person name are highly ranked. So by starting with the highest rank document, the SPC algorithm may capitalize on this external but implicit, knowledge. Finally, SPC is a very efficient algorithm and classification/clustering can be performed online, i.e., as the documents are downloaded.

The process for assignment is performed as follows: The first document is taken and assigned to the first cluster. Then each subsequent document is compared against each cluster with a similarity measure. A document is assigned to the most likely cluster, as long as the similarity score is higher than a threshold γ (this implements assumption 3); otherwise, the document is assigned to a new cluster, unless the maximum number of desired clusters η has been reached; in that case the document is assigned to the last cluster (i.e., the left overs).

We employ two similarity measures ($\mathrm{sim}(D,C)$): Naive Bayes and a standard cosine measure using a TF.IDF weighting scheme.

Naive Bayes

The Naive Bayes similarity measure uses the log odds ratio to decide whether the document is more likely to be generated from that cluster or not ($\mathrm{sim}(D,C) = O(D,C)$). This approach follows Kalt [17]'s work on document classification using the document likelihood by representing the cluster as a multinomial term distribution (i.e., a cluster language model) and predicting the probability of a document D, given the cluster language model, i.e., $p(D|\theta_C)$. It is assumed that the terms t in a document are sampled *independently and identically*, so the odds ratio is calculated as follows:

$$O(D,C) = \frac{p(D|\theta_C)}{p(D|\theta_{\bar{C}})} = \frac{\prod_{t \in D} p(t|\theta_C)^{n(t,D)}}{\prod_{t \in D} p(t|\theta_{\bar{C}})^{n(t,D)}}, \tag{1}$$

where $n(t,D)$ is the number of times term t appears in document D, and $\theta_{\overline{C}}$ is the language model that represents "not being in the cluster." Note that this is similar to the well-known relevance modeling approach [18] except, here, it is applied in the context of classification, as done in [17]. The cluster language model is estimated by performing a linear interpolation between the empirical probability of a term occurring in the cluster $p(t|C)$ and the background model $p(t)$, the probability of a term occurring at random in the collection, i.e., $p(t|\theta_C) = \lambda \cdot p(t|C) + (1-\lambda) \cdot p(t)$. The "not in the cluster" language model $\theta_{\overline{C}}$ is approximated by using the background model $p(t)$.

Cosine Similarity with TF.IDF

The other similarity measure we consider for single pass clustering is the cosine distance. Let $\mathbf{t}(D)$ and $\mathbf{t}(C)$ be term frequency vectors, weighted by the TF.IDF formula, representing document D and cluster C, respectively. Similarity is then estimated using the cosine distance of the two vectors:

$$\mathrm{sim}(D,C) = \cos(\mathbf{t}(D), \mathbf{t}(C)) = \frac{\mathbf{t}(D) \cdot \mathbf{t}(C)}{\|\mathbf{t}(D)\| \cdot \|\mathbf{t}(C)\|}. \tag{2}$$

3.2 K-Means Clustering

K-means is a clustering technique [13], that creates a partitioning of data (i.e., the documents) given a desired number of clusters, K. K-means is based on the idea that a center point (centroid) can adequately represent a cluster. The basic K-means clustering algorithm for finding K clusters is as follows:

1. Select K points as the initial centroids
2. Assign all points to the closest centroid
3. Recompute the centroid of each cluster

4. Repeat steps 2 and 3 until centroids do not change

Again, we use the cosine similarity measure to compute which cluster centroid is closest to the given document. Following standard practice we calculate the centroid of clusters using the mean of the documents within the cluster. As advocated in [25] we use an incremental version of k-means clustering, i.e., centroids are updated as each point is assigned to a cluster, rather than the end of an assignment pass as in the basic version. We set K based on the actual number of person name senses; see Section 5.2 for details.

3.3 Agglomerative Clustering

Agglomerative clustering [16] (AGGLOM) starts with documents as individual clusters and, at each step, merges the most similar pair of clusters. This is repeated until all clusters have been merged into a single cluster that contains all documents (thus, it is a hierarchical bottom-up approach). However, we want a partition of disjoint clusters, therefore, the hierarchy needs to be cut at some point. We use a pre-specified threshold γ for the level of similarity used to determine the cutting point.

The distance between two clusters C_1 and C_2 is determined based on the maximum distance between elements of each cluster (also called complete linkage clustering). We use the cosine similarity to estimate the similarity of two pages. Formally:

$$sim(C_1, C_2) = \max\{\cos(\mathbf{t}(D_1), \mathbf{t}(D_2)) : D_1 \in C_1, D_2 \in C_2\}, \tag{3}$$

where $\mathbf{t}(D_1)$ and $\mathbf{t}(D_2)$ are TF.IDF weighted term frequency vectors representing documents D_1 and D_2, respectively.

Note that each agglomeration step occurs at a greater distance between clusters than the previous agglomeration. We decide to stop clustering when the clusters are too far apart to be merged, i.e., the distance criterion $sim(C_1, C_2) > \gamma$ is not met (and this implements assumption 3).

3.4 Probabilistic Latent Semantic Analysis

The final method for disambiguation we employ is probabilistic latent semantic analysis (PLSA) [15]. PLSA can be used to clusters documents based on the semantic decomposition of the term document matrix into a lower dimensional latent space. Formally, PLSA can be defined as:

$$p(t,d) = p(d) \sum_z p(t|z) p(z|d), \tag{4}$$

where $p(t,d)$ is the probability of term t and document d co-occurring, $p(t|z)$ is the probability of a term given a latent topic z and $p(z|d)$ is the probability of a latent topic in a document. The prior probability of the document, $p(d)$, is assumed to be uniform. This decomposition can be obtained automatically using the EM algorithm [15]. Once estimated, we make the simplifying assumption that each latent topic represents one of the different senses of the person name. The document d is assigned to one of the person-topics z if (i) $p(z|d)$ is the maximum argument, and (ii) the odds of the document given z, i.e., $O(z,d)$, is greater than a threshold γ, where

$$O(z,d) = \frac{p(z|d)}{p(\bar{z}|d)} = \frac{p(z|d)}{\sum_{z',z'\neq z} p(z'|d)}. \tag{5}$$

Note that the requirement $O(z,d) > \gamma$ implements assumption 3: sufficient evidence must be found before assignment to a cluster can be made. All documents un-assigned are placed into their own cluster as per assumption 3.

In order to automatically select the number of person senses using PLSA, we perform the following process to decide when the appropriate number of person name-senses (defined by z) have been identified: (1) we set $z = 2$ and compute the log-likelihood of the decomposition on a held out sample of data; (2) we increment z and compute the log-likelihood again; if the log-likelihood has increased (by an amount larger than 0.001), then we repeat step 2, else (3) we stop as we have now maximized the log-likelihood of the decompositions with respect to the number of person name-senses. This point is assumed to be optimal with respect to the number of person name senses. Since we are focusing on identifying the true number of referents, this should result in higher inverse purity, whereas with the single pass and agglomerative clustering the number of clusters is not restricted, and so we would expect single pass and agglomerative clustering to produce more clusters but with a higher purity.

4 Evaluation platform

In this section we describe the data set used, the evaluation metrics and methodology along with details concerning the preprocessing and representation of documents and estimation of PLSA.

4.1 Data Set

The data set we used for our experiments is from the Web People Search track at the Semantic Evaluation 2007 Workshop [3]. This data set consists of pages obtained from the top 100 results for a person name query to a web search engine.[2] Each web

[2] Note that 100 is an upper bound, for some person names there are fewer documents.

page from the result list is stored, as well as metadata, including the original URL, title, position in the ranking, and the snippet generated by the search engine. Annotators manually classified each web page to create a ground truth for evaluation. It is important to note that the original task statement allows a document to be assigned to multiple clusters, if it has multiple referents mentioned. However, because this was quite rare, we engaged a simplifying assumption and only perform hard classification (and leave fuzzy/soft classification for further work). Another caveat in this data set is that some web pages did not contain enough information about the person to make a decision (usually because the URL was out of date). These documents were discarded from the evaluation process (but not from the data set) and so we accept that a small amount of noise is introduced by the inclusion of these documents.

The collection is divided into training and test sets, comprising 49 and 30 person names, respectively.[3] In order to provide different ambiguity scenarios, the data set is made up of person names from different sources (the source of names was known in advance only for the training data):

US Census 42 names (32/10 in training/test set) picked randomly from the Web03 corpus [20];

Wikipedia 17 names (7/10 in training/test set) sampled from a list of ambiguous person names in the English Wikipedia;

ECDL06 10 names (training set only) randomly selected from the Program Committee listing of a Computer Science conference (ECDL 2006); and

ACL06 10 names (test set only) randomly selected from participants of a Computer Science conference (ACL 2006).

Table 1 Statistics of the data collection. The columns of the table are: data set (or source), number of person names (queries), average number of documents (per person name), average number of discarded documents (per person name), average number of referents (per person name).

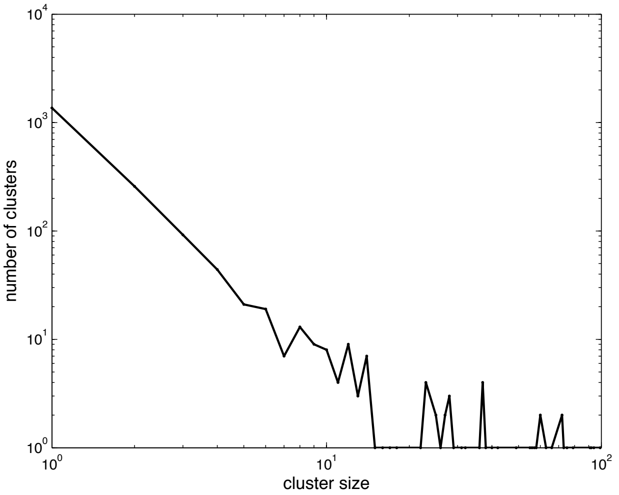
Data set/source	#names	docs	discarded	referents
Training set	49	71.02	26.00	10.76
US Census	32	47.20	18.00	5.90
Wikipedia	7	99.00	8.29	23.14
ECDL06	10	99.20	30.30	15.30
Test set	30	98.93	15.07	45.93
US Census	10	99.10	14.90	50.30
Wikipedia	10	99.30	17.50	56.50
ACL06	10	98.40	12.80	31.00

The statistics of the training/test sets and the different sources are shown in Table 1. Despite the fact that both were sampled from the same sources, the ambiguity in the

[3] The WePS organizers also released a trial data set, consisting of an adapted version of WePS corpus, described in [2]. We did not use this corpus in our experiments, but limited ourselves to the official SemEval training and test collections.

test data (45.93 referents per person name, on average) is much higher than in the training data (where it is only 10.76). According to Artiles et al. [3], this shows that "there is a high (and unpredictable) variability, which would require much larger data sets to have reliable population samples." In order to measure performance as reliably as possible given the SemEval test suite, we conduct our experiments using all names from both the training and the test sets; we will refer to it as *all names*. Unless stated otherwise results are reported on all names.

Fig. 1 Number of clusters for each cluster size on test+train data (log-log plot shown)



While there appears to be high ambiguity due to the large number of person name senses, we assumed that the distribution of documents to person name senses would follow a power law. Figure 1 shows that the size of the clusters follows a power law with an exponent of approximately 1.31 estimated used linear regression of the log-log plot. This confirms the second assumption of the data and is a novel finding regarding this task/data.

4.2 Performance Measures

Evaluation of the SemEval WePS task is performed using standard clustering measures: purity and inverse purity. Purity is related to the precision measure, well known in IR, and rewards methods that introduce less noise in each cluster. The overall purity of a clustering solution is expressed as a weighted average of maximal precision values:

$$purity = \sum_i \frac{|C_i|}{n} \max precision(C_i, L_j), \tag{6}$$

where n denotes the number of documents, and the precision of a cluster $|C_i|$ for a given category L_j is defined as:

$$precision(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}. \tag{7}$$

Inverse purity focuses on recall, i.e., rewards a clustering solution that gathers more elements of each class into a corresponding single cluster. Inverse purity is given by:

$$inv.purity = \sum_i \frac{|L_i|}{n} \max precision(L_i, C_j). \tag{8}$$

We get a weighted version of the F-measure by computing a weighted average of the purity and inverse purity scores:

$$F = \frac{1}{\alpha \frac{1}{purity} + (1-\alpha)\frac{1}{inv.purity}}, \tag{9}$$

where $\alpha \in [0..1]$ is a parameter to set the ratios between purity and inverse purity.

The harmonic mean ($\alpha = 0.5$) was used for the final ranking of systems at SemEval, and $F_{0.2}$ was also reported as an additional measure, which gives more importance to the inverse purity aspect ($\alpha = 0.2$). Artiles et al. [2] argue that the rationale for using $F_{0.2}$, from a user's point of view, is that "it is easier to discard a few incorrect web pages in a cluster which has all the information needed, than having to collect the relevant information across many different clusters." We decided to also report on $F_{0.8}$, a measure which gives more importance to the purity aspect ($\alpha = 0.8$). Our motivation for also reporting $F_{0.8}$ is that from a machine point of view, it is more important to ensure that the precision/purity of the clusters are high (so that any subsequent task involving their use, like building a profile, does not contain any unnecessary noise).

4.3 Document Representation

A separate index was built for each person, using the Lemur toolkit[4]. We used a standard (English) stopword list but did not apply stemming. A document was represented using the *title* and *snippet* text from the search engine's output, and the *body text* of the page, extracted from the crawled HTML pages, using the method described below.

[4] URL: http://www.lemurproject.org

4.3.1 Acquiring Plain-Text Content from HTML

Our aim here is to extract the plain-text content from HTML pages and to leave out blocks or segments that contain little or no useful textual information (headers, footers, navigation menus, adverts, etc.). To this end, we exploit the fact that most web pages consist of blocks of text content with relatively little markup, interspersed with navigation links, images with captions, etc. These segments of a page are usually separated by block-level HTML tags. Our extractor first generates a syntax tree from the HTML document. We then traverse this tree while bookkeeping the stretch of uninterrupted non-HTML text we have seen. Each time we encounter a block-level HTML tag we examine the buffer of text we have collected, and if it is longer than a threshold, we output it. The threshold for the minimal length of buffer text was empirically set to 10. In other words, we only consider segments of the page, separated by block-level HTML tags, that contain 10 or more words.

4.4 PLSA Estimation

We used the Lemur toolkit and the PennAspect implementation of PLSA [24] for our experiments, where the parameters for PLSA were set as follows. For each k we perform 10 initializations where the best initialization in terms of log-likelihood is selected. The EM algorithm is run using tempering with up to 100 EM Steps. For tempering, the setting suggested in [15] is used. The models are estimated on 90% of the data and 10% of the data is held out in order to compute the log-likelihood of the decompositions.

5 Experiments and Results

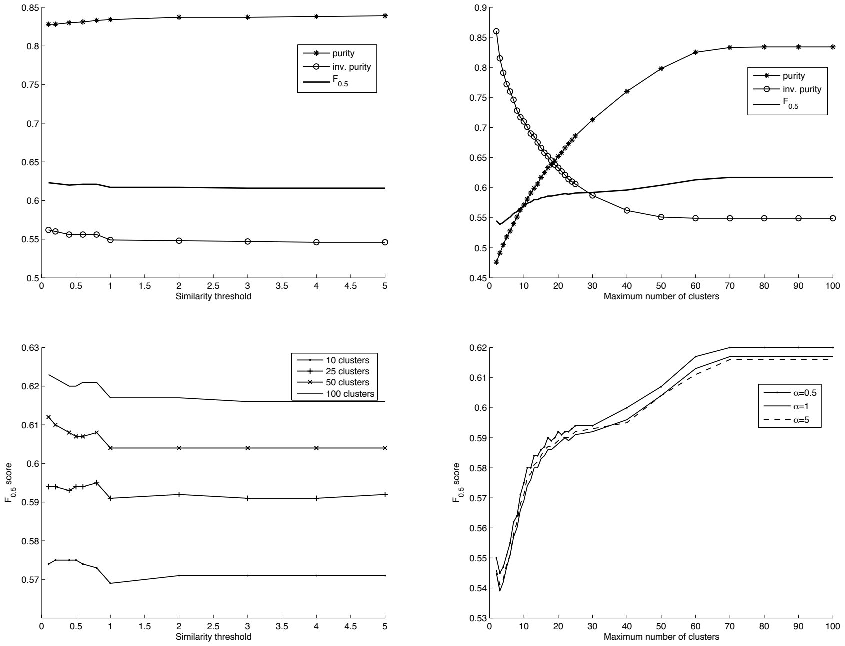
In this section, we present an experimental evaluation of the four clustering approaches. We address the following specific research questions, leaving the more general ones surrounding the person cluster hypothesis to Section 6:

- What factors affect performance? I.e., number of clusters, similarity threshold, similarity metric, etc.
- How stable is the performance?
- When is the best performance obtained?
- What is the best number of clusters to use? Can we determine this automatically?
- How do the different clustering approaches compare to each other?

We start by exploring the performance and behavior of the Single Pass Clustering, k-Means Clustering, Agglomerative Clustering, and Probabilistic Latent Semantic Analysis methods, separately. Then, we compare and contrast the various methods, before providing an analysis over different groups (as opposed to aggregated over all topics).

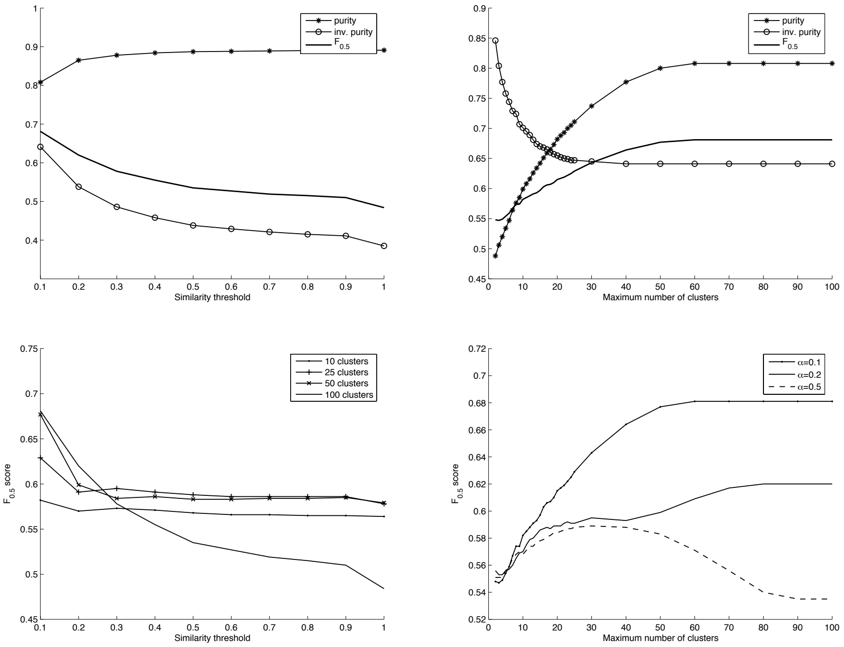
5.1 Single Pass Clustering

Fig. 2 Single Pass Clustering using the Naive Bayes similarity measure. (Top Left): varying the similarity threshold, maximum number of clusters is fixed. (Top Right): varying the maximum number of clusters, similarity threshold is fixed. (Bottom Left): varying the similarity threshold, using different cluster size configurations. (Bottom Right): varying the cluster sizes, using different similarity threshold configurations.



Figures 2 and 3 present the results of the SPC method along a number of dimensions, using the Naive Bayes (SPC-NB) and cosine similarity measures (SPC-COS), respectively. Results are aggregated over all names (including both the training and test sets); the best scoring configurations are summarized in Tables 3 and 4. The top left plot shows performance given the maximum number of clusters is fixed ($\eta = 100$) as the similarity threshold is varied. The bottom left plots shows the harmonic F-score displayed for various similarity thresholds for η equal to a maximum of 10, 25, 50, and 100 clusters. These two plots across the similarity threshold show that the performance of either SPC is very stable w.r.t. the threshold, but the best performance obtained is with a lower threshold. This implies that the similarity between documents need not be very high (i.e., evidence to the contrary for assignment can be quite low, for assumption 3.).

Fig. 3 Single Pass Clustering using the cosine similarity measure. (Top Left): varying the similarity threshold, maximum number of clusters is fixed. (Top Right): varying the maximum number of clusters, similarity threshold is fixed. (Bottom Left): varying the similarity threshold, using different cluster size configurations. (Bottom Right): varying the cluster sizes, using different similarity threshold configurations.



In the top right plots, the similarity threshold is fixed ($\gamma = 0.1$) and performance is measured against different maximal cluster size limits. In the bottom right plots, the F-score is explored against the possible cluster sizes, using different similarity threshold configurations. In these two plots across the maximum number of clusters, we can see that enforcing a limit on the clustering is not appropriate—and actually violates the third assumption. However, this appears to be in contrast with the similarity threshold, which from above, does not need to be very high.

Consequently, the best performance was achieved when the maximum number of clusters (η) was set to 100, and this was independent of the similarity measure. And while performance was quite stable given the similarity threshold, γ set to 0.1 was the threshold which delivered the highest F-score.

5.2 K-Means Clustering

Note that in order to perform k-Means Clustering, the number of desired clusters (K) has to be specified. We set K to be the actual number of person-senses based on the ground truth. This is a special—and arguably unrealistic—experimental condition, to determine the performance that could be achieved with this clustering method if the number of person-senses were known. We view this as an upper bound for the capability of k-Means clustering. As the performance of this method may vary depending on the initial assignments, the algorithm is run 100 times, and the scores are averaged over all 100 runs. The results are presented in Tables 3 and 4.

5.3 Agglomerative Clustering

Figure 4 shows the performance of Agglomerative Clustering as the similarity threshold (γ) is varied. We see from the plot that performance in terms of inverse purity and $F_{0.5}$ score decreases rapidly as the threshold hold increases (and conversely for purity). The best overall result (in terms of $F_{0.5}$ score) is obtained with a low γ value. This is in accordance with what we have seen for SPC; the evidence to the contrary for assignment can be quite low, for assumption 3. Results of the best scoring configuration are detailed in Tables 3 and 4.

Fig. 4 The effect of varying the similarity threshold for agglomerative clustering.

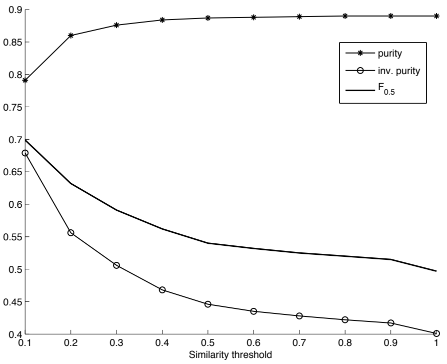


Table 2 Performance of PLSA. Best scores are in boldface.

Experimental condition	pur.	invp.	$F_{0.5}$	$F_{0.2}$	$F_{0.8}$
Manual (truth)	0.530	0.647	0.547	0.591	0.530
Auto ($\gamma = 0.5$)	0.495	**0.800**	0.536	**0.624**	0.501
Auto ($\gamma = 1.0$)	0.517	0.782	0.543	0.622	0.515
Auto ($\gamma = 5.0$)	**0.662**	0.647	**0.561**	0.583	**0.584**

5.4 PLSA

Table 2 reports on the results we obtained from PLSA under two different experimental conditions. The first is a manual configuration of the number of latent topics which is set to be the actual number of person name-senses, based on the ground truth files. This is to determine an upper bound, which could be achieved if the number of latent topics could be identified, and assuming that each latent topic is actually representative of each person name-sense. The other, more realistic experimental setting uses unsupervised learning to determine the number of latent topics within the set of documents (as explained in Subsection 3.4). For this setting, we varied the similarity threshold. Surprisingly, the manual setting did not perform very well at all, and shows that the latent topics are not really that representative of the individual person name senses. We suspect this is because the distribution over the latent topics is dominated by only a few (“principal”) components, so to speak; and so the number of resulting clusters is quite low (as we shall see in a following subsection); the automatic methods stop, in theory, when the overriding latent factors have been identified (because using any more would just introduce noise). Consequently, we find that the best performance for PLSA is obtained when the number of clusters is automatically estimated. In contrast to the SPC and AGGLOM methods, the number of clusters identified is very low, which results in high inverse purity scores, but lower purity (as we anticipated). Interestingly, for PLSA increasing the threshold means that more clusters are created, but at the expense of inverse purity.

5.5 Comparing Methods

Table 3 presents the results achieved by the best performing configuration of the different clustering approaches, while Table 4 presents a breakdown of scores to training and test sets. The parameters used are: $\lambda = 0.5$, $\eta = 100$, $\gamma = 0.1$ for SPC with Naive Bayes similarity (SPC-NB), $\eta = 100$, $\gamma = 0.1$ for SPC with cosine similarity (SPC-COS), $\gamma = 0.1$ for AGGLOM, and $\gamma = 1.0$ for PLSA.[5] Note that K is set to the actual number of person senses for k-Means Clustering.

[5] In the case of PLSA, there is no ‘best’ γ, the setting we use is the one that performs well across the board.

Table 3 Results achieved by the best performing configurations of the different approaches. Best scores are in boldface.

Method	pur.	invp.	$F_{0.5}$	$F_{0.2}$	$F_{0.8}$
SPC-NB	**0.828**	0.562	0.623	0.579	0.705
SPC-COS	0.808	0.641	0.681	0.651	0.736
K-MEANS	0.742	0.658	0.678	0.660	0.710
AGGLOM	0.791	0.679	**0.699**	**0.681**	**0.739**
PLSA	0.517	**0.782**	0.543	0.622	0.515

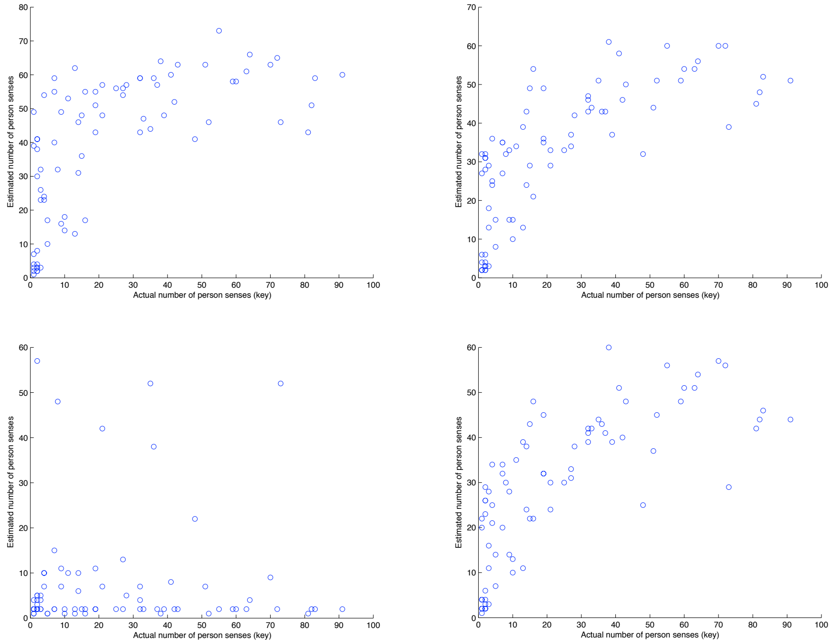
Table 4 Breakdown of results achieved by the best performing configurations of the different approaches to training and test sets. Best scores are in boldface.

Method	Training set					Test set				
	pur.	invp.	$F_{0.5}$	$F_{0.2}$	$F_{0.8}$	pur.	invp.	$F_{0.5}$	$F_{0.2}$	$F_{0.8}$
SPC-NB	**0.793**	0.484	0.547	0.501	0.641	**0.884**	0.688	0.747	0.707	0.809
SPC-COS	0.782	0.557	0.613	0.572	0.688	0.850	0.777	0.791	0.780	**0.815**
K-MEANS	0.694	0.632	0.634	0.625	0.662	0.820	0.701	0.750	0.718	0.789
AGGLOM	0.775	0.597	**0.640**	0.608	**0.700**	0.818	0.812	**0.796**	**0.802**	0.803
PLSA	0.607	**0.719**	0.605	**0.647**	0.596	0.370	**0.885**	0.442	0.581	0.382

We can see the contrast between the methods when we consider the number of clusters each method creates against the actual number of person name-senses. Figure 5 plots the number of estimated clusters against the actual number of clusters, extracted from the truth files for each of the clustering methods. Clearly, the SPC and AGGLOM methods are providing a good estimate of the number of person name-senses. This is reflected by the strong correlation between clusters and person name senses. The Pearson's Correlation coefficients for SPC-NB, SPC-COS, and AGGLOM are $r = 0.736$, $r = 0.634$, and $r = 0.729$, respectively—where $r = 1$ would indicate that the method correctly identifies the true number of person name senses. On the other hand, the assignment to clusters based on the max $p(z|d)$ given the PLSA decomposition completely underestimates the number of person senses and the correlation is very weak ($r = 0.045$). Alternative assignment methods (such as clustering the latent space) could provide improvements but is left for future work.

These results clearly demonstrate the difference in the behaviors of the term-based approaches (SPC, K-MEANS, AGGLOM) and the semantic-based approach (PLSA), with the former outperforming the latter. Out of the term-based approaches, AGGLOM and SPC-COS deliver the best performance. We identify AGGLOM as the preferred method, as it gives a better estimation of the actual number of person name senses. The performance of K-MEANS is somewhat disappointing, as it delivers the worst results among the term-based methods. Overall, the term-based approaches assign people to the same cluster with high precision, as is reflected by the high purity scores. In contrast, the PLSA method produces far fewer clusters per person. These clusters may cover multiple referents of a name, as is witnessed by the low purity scores. On the other hand, inverse purity scores are very high, which means referents are usually not dispersed among clusters.

Fig. 5 Estimated versus actual number of person senses. (Top Left) SPC-NB, (Top Right) SPC-COS, (Bottom Left) PLSA, (Bottom Right) AGGLOM. The Pearson correlation coefficient *r* is 0.736, 0.634, 0.045, and 0.729 respectively.



5.6 Group-Level Analysis

The results on which we have reported so far were aggregated over all people. Since the data is not homogeneous, it is interesting to see how performance varies on different groups of people. More specifically, we seek to answer: what is the performance of the methods like over (i) different data sources and (ii) different numbers of person name senses?

Figure 6 (Left) shows the performance of SPC, AGGLOM and PLSA across the different data sources. Note that we report only on the better performing SPC variation (SPC-COS) and we exclude K-MEANS (as it would not be a fair comparison given that information from the ground truth was used to set the value of *K*). All sources display high levels of variability, which seems independent of the size of the source. In case of SPC-COS and AGGLOM, the level of variance is more prominent for the US Census data than for the other three sets. The median F-scores of US Census and ECDL are in the same range (0.66–0.73), as are Wikipedia and ACL06 (0.78–0.81). However, for PLSA, the deviation is very high for all sources. The median F-scores of Wikipedia and ECDL are in the same range (0.64–0.66), but US Census and ACL06 are significantly lower (0.51 and 0.43, respectively).

Fig. 6 Performance of (Top) SPC-COS, (Middle) AGGLOM, and (Bottom) PLSA. (Left) across different data sources, (Right) against different cluster sizes. The parameter settings for these methods correspond to the configurations reported in Table 3.

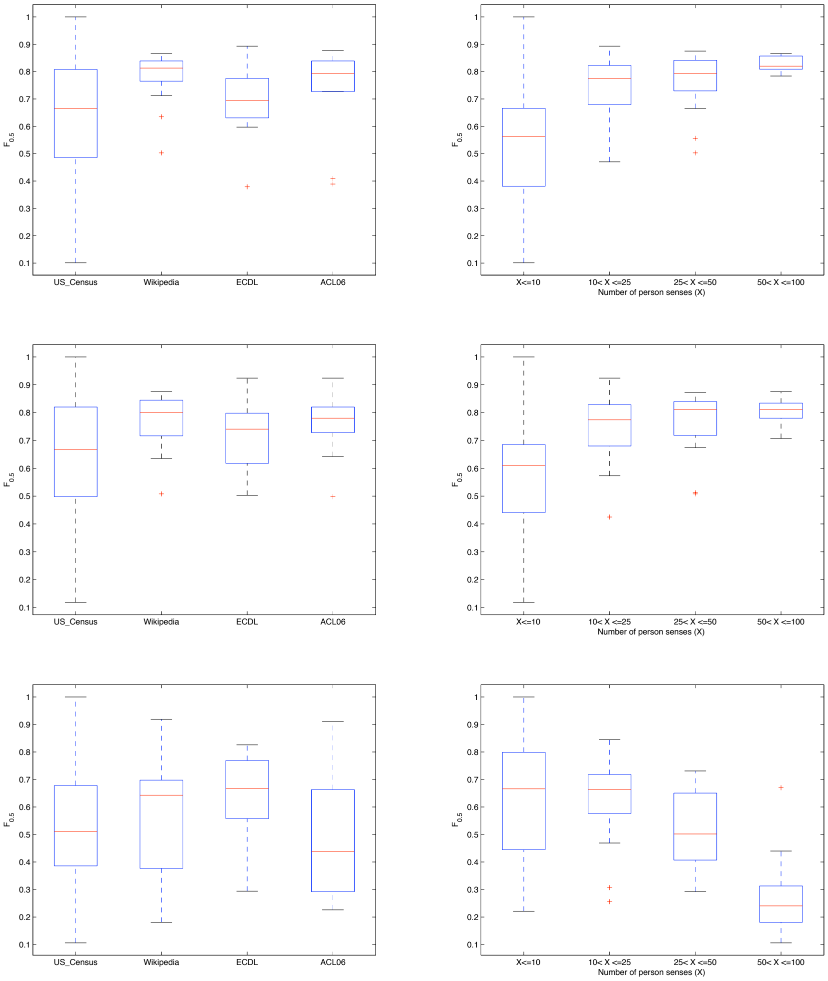


Figure 6 (Right) shows the performance of the methods across the different cluster sizes, where the cluster size is the number of senses of a person name, based on the ground truth. Interestingly, as the number of senses goes up, so does the F-score achieved by the SPC-COS and AGGLOM algorithms. On the other hand, PLSA seems to have an orthogonal effect, the best F-score is achieved when the number of senses is low (≤ 10), and performance is gradually decreasing, as the number of senses increases. This behavior confirms our intuition, that the distribution of latent topics may be dominated by a few principal components, which are easier to

associate with prominent person name senses, or when there are only a few referents. When only limited examples of the other referents are available (i.e., one or two documents, which is often the case according to assumption 2), PLSA seems unable to specifically identify such cases.

6 Discussion and Conclusion

Table 5 Comparison of results to baselines and top performing systems at the SemEval 2007 WePS task [3]. Results are reported on the test set only.

Method	pur.	invp.	$F_{0.5}$	$F_{0.2}$
SPC-NB	**0.884**	0.688	0.747	0.707
SPC-COS	0.850	0.777	0.791	0.780
K-MEANS	0.820	0.701	0.750	0.718
AGGLOM	0.818	0.812	**0.796**	**0.802**
PLSA	0.370	**0.885**	0.442	0.581
CU_COMSTEM	0.720	**0.880**	**0.780**	**0.830**
IRST-BP	**0.750**	0.800	0.750	0.770
PSNUS	0.730	0.820	0.750	0.780
ONE-IN-ONE	1.000	0.470	0.610	0.520
ALL-IN-ONE	0.290	1.000	0.400	0.580

In this chapter, we have explored the person cluster hypothesis for the person name resolution task in a web setting. As we have seen, SPC and AGGLOM with a standard bag of words representation provide excellent performance on this task. To put our results in context, Table 5 reports the results of our best performing methods, along with the top performing systems from SemEval [3] and two naive baselines: ONE-IN-ONE, which assumes that each document is a different referent (i.e., the worst case scenario of assumption 3, if we had no evidence), and ALL-IN-ONE, which assumes that all documents are associated with a single referent. While two of the top performing systems use richer features and more sophisticated clustering methods than we do, the performance of SPC-COS and AGGLOM are comparable to, if not better than the state of the art, and provides a strong baseline for this task. This is truly remarkable, and demonstrates that viewing the task of person name resolution as document clustering is quite effective. Furthermore, we contend that this result provides strong evidence to support the "person cluster hypothesis".

While the way in which we used PLSA for this task has not performed as well as we expected, we have identified a number of possible reasons for this failure. We also noted that when there are only few person name senses, PLSA is more effective than the term-based approaches. An interesting line of future work would be to consider how the advantages of both methods could be combined in order to gain greater improvements. Other areas for future research where improvements could be gained include employing a richer feature set which includes named entities, etc., and pre-processing the documents to remove irrelevant content before the disambiguation process.

7 Acknowledgments

We thank the participants in the NLPIX 2008 workshop at WWW 2008 for their questions and suggestions. This research was supported by the E.U. IST programme of the 6th FP for RTD under project MultiMatch contract IST-033104, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 220-80-001, 017.001.190, 640.001.501, 640.002.501, 612.066.512, STE-07-012, 612.061.814, 612.061.815.

8 References

[1] R. Al-Kamha and D. W. Embley. Grouping search-engine returned citations for person-name queries. In *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 96–103, New York, NY, USA, 2004. ACM Press.

[2] J. Artiles, J. Gonzalo, and F. Verdejo. A testbed for people searching strategies in the www. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 569–570, New York, NY, USA, 2005. ACM Press.

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

[4] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proc. 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th Conf. on Computational Linguistics (COLING)*, pages 79–85, 1998.

[5] K. Balog. *People Search in the Enterprise*. PhD thesis, University of Amsterdam, June 2008.

[6] K. Balog and M. de Rijke. Associating people and documents. In C. Macdonald et al., editor, *30th European Conference on Information Retrieval (ECIR 2008)*, pages 296–308, 2008.

[7] K. Balog, L. Azzopardi, and M. de Rijke. Personal name resolution of web people search. In *WWW2008 Workshop: NLP Challenges in the Information Explosion Era (NLPIX 2008)*, April 2008.

[8] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th International World Wide Web (WWW) Conference*, pages 463–470, 2005.

[9] D. Bollegala, Y. Matsuo, and M. Ishizuka. Extracting key phrases to disambiguate personal name queries in web search. In *Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval? at ACL'06*, pages 17–24, 2006.

[10] N. Craswell, A. de Vries, and I. Soboroff. Overview of the TREC-2005 Enterprise Track. *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2006.

[11] M. Fleischman and E. Hovy. Multi-document person name resolution. In *Proc. 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, 2004.

[12] C. Gooi and J. Allan. Cross-document coreference on a large scale corpus. In *Proc. Human Language Technology/North American chapter of Association for Computational Linguistics annual meeting (HLT/NAACL),*, 2004.

[13] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28: 100–108, 1979.

[14] D. R. Hill. A vector clustering technique. In Samuelson, editor, *Mechanised Information Storage, Retrieval and Dissemination*, North-Holland, Amsterdam, 1968.

[15] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999. URL `citeseer.ist.psu.edu/hofmann99probabilistic.html`.

[16] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7:217–240, 1971.

[17] T. Kalt. A new probabilistic model of text classification and retrieval. Technical Report CIIR TR98-18, University of Massachusetts, January 1996.

[18] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the 24th annual international ACM SIGIR conference*, pages 120–127, New Orleans, LA, 2001. ACM Press.

[19] B. Malin. Unsupervised name disambiguation via social network similarity. In *Proceedings of the SIAM Workshop on Link Analysis, Counterterrorism, and Security*, pages 93–102,, 2005.

[20] G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Conference on Computational Natural Language Learning (CoNLL)*, 2003.

[21] G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6:1–28, 1991.

[22] T. Pedersen, A. Purandare, and A. Kulkarni. Name discrimination by clustering similar contexts. In *Computational Linguistics and Intelligent Text Processing*, pages 226–237. Springer Berlin - Heidelberg, 2005.

[23] X. Phan, L. Nguyen, and S. Horiguchi. Personal name resolution crossover documents by a semantics-based approach. *IEICE Transactions on Information and Systems*, E89-D(2): 825–836, 2006.

[24] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, New York, NY, USA, 2002. ACM Press. See `http://www.cis.upenn.edu/datamining/software_dist/PennAspect/`.

[25] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. *In Proceedings of Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00)*, pages 109–110, 2000.

[26] M. Taffet. Looking ahead to person resolution. In *Proceedings of the 4th Annual Workshop on Technology for Family History and Genealogical Research*, pages 11–15, 2004.

[27] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 163–170, New York, NY, USA, 2005. ACM Press.