ABSTRACT

Some of the main ranking features of today’s search engines reflect result popularity and are based on ranking models, such as PageRank, implicit feedback aggregation, and more. While such features yield satisfactory results for a wide range of queries, they aggravate the problem of search for ambiguous entities: Searching for a person yields satisfactory results only if the person we are looking for is represented by a high-ranked Web page and all required information are contained in this page. Otherwise, the user has to either reformulate/refine the query or manually inspect low-ranked results to find the person in question. A possible approach to solve this problem is to cluster the results, so that each cluster represents one of the persons occurring in the answer set. However clustering search results has proven to be a difficult endeavor by itself, where the clusters are typically of moderate quality.

A wealth of useful information about persons occurs in Web 2.0 platforms, such as LinkedIn, Wikipedia, Facebook, etc. Being human-generated, the information on these platforms is clean, focused, and already disambiguated. We show that when searching for ambiguous person names the information from such platforms can be bootstrapped to group the results according to the individuals occurring in them. We have evaluated our methods on a hand-labeled dataset of around 5,000 Web pages retrieved from Google queries on 50 ambiguous person names.

1. INTRODUCTION

With ever more information being placed on the Web, established retrieval techniques are undergoing a stress test. Although search engines have matured by integrating different relevance criteria, e.g., query-based and social relevance, result freshness, user interests, etc., they still lack the ability to effectively respond to ambiguous queries for specific entities, such as people, products, locations, etc. The common way by which modern search engines approach the ambiguity problem is by diversifying search results and hoping that at least one of the top-10 results satisfies the user’s information need. However, the employed ranking strategies mostly rely on authority- and popularity-based measures, e.g., PageRank scores [26], models for aggregating implicit relevance feedback (e.g., in terms of user clicks), etc. As a consequence, while the diversification approach works well for popular searches, for which there exist authoritative Web pages and plenty of user feedback, there is a long tail of results to ambiguous queries, which does not fulfill the mentioned criteria. The severeness of this problem becomes especially obvious in search tasks involving ambiguous person names. In such cases, the returned results are only satisfactory if the person in question is represented by a high-ranked Web page with the required information. Otherwise, the user has to either refine the query (through additional terms) or manually inspect low-ranked results.

For example, suppose that you have recently attended a conference talk by computer scientist Michael Jordan and you are interested in more background information about the speaker and his research. Searching for the name “Michael Jordan” on Google yields top-10 results entirely about the former basketball player. In fact, the first page about the well-known researcher from the U.C. Berkeley is ranked 18th in the result list. For the average user, who aims at a top-10 hit for his search, this is unacceptable.

For such queries, it would be useful to present the user with clusters of results where each cluster represents one of the individuals occurring in the answer set. Typical approaches to this problem retrieve salient text snippets for the ambiguous query and cluster results based on textual similarity measures, using predefined or learned thresholds. Other features, such as links or word senses (concepts), can also be taken into account. Obviously, such techniques have to handle a lot of noise and it is questionable whether they can handle highly ambiguous person-related queries (e.g.,

1The second hit about the researcher is ranked 47th.
“Michael Jordan”), with different persons of the same name and the same category, say “computer scientists” (DBLP alone lists 3 Michael Jordan). Under such noisy conditions typical clustering techniques, such as K-Means or Hierarchical Agglomerative Clustering (HAC) [24], are shown to perform rather moderately [37]. We have evaluated the performance of well-known clustering methods, such as HAC and K-Means on our dataset (see Section 4.2), and found that although they perform well in terms of purity, but the resulting clusters yield low normalized mutual information (NMI) scores. Our approaches outperformed both methods by approximately 10% in terms of purity, respectively up to 95% in terms of NMI.

A study on the distribution of query types on AlltheWeb and AltaVista shows that, already in 2003, person-related queries made up a non-negligible portion of the Web search queries [31]. In today’s Web ever more information centers around people, be it in social networks, encyclopedic sources, or professional homepages. LinkedIn, Facebook, Wikipedia, Twitter, and Xing are only part of a wide and continuously growing range of Web 2.0 platforms centering around people and other entities. Therefore, we believe that person-related queries will continue to gain importance. At the same time, being human-generated, the information on such platforms is focused and already disambiguated [8].

Our approach to the ambiguous person search problem can bootstrap the information of a knowledge base to identify groups of results that represent unique individuals. In this way the original clustering problem is cast into a classification problem, where the classes are given by the different same-name individuals occurring in the knowledge base. We are aware of the existing coverage problem: there remain many people who may not appear in the knowledge base. However, as shown in our evaluation (i.e., Section 4.2) the classification into knowledge base entities leads to a quality enhancement for pages regarding these particular entities. Furthermore, the remaining pages might be clustered traditionally or further linked to entities of other knowledge bases in a separate step, but this is not in the scope of this work.

Note that for any practical relevance, approaches to ambiguous person search on the Web need to be effective (i.e., the returned groups need to have high quality) and efficient (i.e., the grouping of results should happen in an online fashion, avoiding long waiting times for the user). Other non-obvious requirements are that the system should handle noise and uncertainty in the grouping process. In this paper, we analyze the result quality of different efficient Information Retrieval and Machine Learning strategies to solve the above problem and show that information bootstrapped from Web 2.0 sources can considerably improve the result of the disambiguation process.

In summary, our contributions are the following:

1. We propose a framework for transforming the task of clustering results to ambiguous person name queries into a classification task that builds on bootstrapping knowledge base entities.

2. We propose and investigate different strategies for mitigating bias and noise during the disambiguation process. While bias inevitably arises from a document ranking (which, in case of PageRank, yields top results of highly linked Web pages about few more “Web-popular” individuals and many low-rank ones about many others), noise may arise from single result pages containing information about different people or about people who are not represented in the knowledge base (open world assumption).

3. We investigate the quality of different efficient Information Retrieval and Machine Learning algorithms with respect to the result disambiguation problem.

4. We demonstrate the viability of our approach in experiments on a hand-labeled dataset of around 5,000 Web pages retrieved from Google queries on 50 ambiguous person names appearing on “Wikipedia’s Human Disambiguation Pages”.

The remainder of this document is organized as follows: Section 2 discusses related research. Section 3 introduces our Web page classification approaches and Section 4 discusses their experimental evaluation. Finally, we discuss future work and conclude in Section 5.

2. BACKGROUND AND RELATED WORK

Entity disambiguation is a broad topic and spans several well-studied research fields in computer science [16]. Due to the large amount of scientific publications in this area, we can discuss only the most relevant fraction of the existing related work with no claim for completeness. In the following we discuss three different research streams: Entity Resolution, Entity Linking, and Text Clustering.

The field of Entity Resolution (ER) (also referred to as record linkage, deduplication, or data matching by the database community) is already summarized in several survey works [10][11][14][20][25]. In this realm, the focus is on comparing sets of entities and identifying convincing/correct mappings between them. However, ER methods typically assume structured entity data, such as database entries with a defined set of attributes (commonly with a value range), which we cannot assume.

Entity Linking (EL) is the task of linking mentions of named entities in Web text with their referent entities in a knowledge base. These knowledge bases might be extracted from various sources, such as Wikipedia, DBLP, IMDb, etc. For instance, Bunescu and Pasca transform the named entity disambiguation problem to a ranking problem of Wikipedia entities [9]. They derive an entity dictionary from Wikipedia and rank these entities for a given mention according to a scoring function that is based on the cosine similarity between the textual context of the mention and Wikipedia’s text and categories of the candidate entity. Cucerzan [12] extends the feature set of a candidate Wikipedia entity by information from other articles linking to the candidate’s article, but instead of using the whole article text only some key phrases and immediate categories are included. Fader et al. [15] extend the previous works by factoring a prominence prior into the disambiguation decision.

Hassell et al. disambiguate the names of academic researchers included in a collection of DBWorld posts by analyzing relationship information between research articles, computer scientists, journals, and proceedings that are extracted from DBLP [17]. All these techniques address a problem similar to ours. However, our problem has to be
solved by more efficient methods, because they have to be applicable to online scenarios and process complete Web pages, which are results to a Web search query.

Another related research field is Text Clustering. The focus is on grouping texts about ambiguous named entities, such that every group uniquely represents an individual entity. Early works in this realm use clustering for cross-document co-reference resolution, i.e., to find references across multiple documents. Subsequent works cluster Web appearances of ambiguous named entities and try to identify the underlying real-world entity (e.g., [23]). One of the most challenging tasks here is the disambiguation of search results to ambiguous person names [1–3], also referred to as personal name resolution [6].

For text clustering, a wide variety of feature selection strategies can be observed: A common representation of Web documents is given by the vector space model of the document terms [5, 7, 8, 22, 30, 33, 34]. The features are typically weighted by their tf-idf weights. Further, more advanced features can be considered, such as named entities [18, 19, 22, 34], noun phrases [18, 34], intrinsic hyper-link relationships among Web pages [7, 18, 19, 30], email address references [19, 34], and detailed personal or biographical features, such as occupation and title, phone number, or birth year [23, 34]. Such features typically lead to a high precision but a low recall of results, because “they are not observed frequently, but work as strong evidence” [18].

Balog et al. compare the performance of a simple bag-of-words based clustering approaches for the personal name resolution task and showed comparable results to state-of-the-art approaches that base on more sophisticated features [6]. Furthermore, Sekine and Artiles state that the extraction of some of these “advanced features” (such as birth date, spouse name, occupation, …) is especially hard and that existing solutions generally provide low F1 measure scores [23]. This poor performance occurs, because Web documents (in contrast to scientific texts, or encyclopedic articles) can be very noisy.

Once the features are selected, the approaches employ various clustering methods to achieve the final grouping. A commonly used clustering method is Hierarchical Agglomerative Clustering (HAC) [8, 22, 23, 30, 33, 34]. Based on this technique, Ikkeda et al. introduce a two-stage clustering [18]. In the first stage, HAC creates clusters that are used to extract refined features for the second stage clustering. Bekkerman and McCallum provide an agglomerative/conglomerative double clustering method [7]. It can be shown that this technique is related to the information bottleneck method, which is known to perform well for text clustering tasks [29]. As one can see, hierarchical algorithms are commonly used in this research field. It is a widespread belief that hierarchical algorithms have a higher clustering quality than partitional methods (e.g., K-Means), which typically provide a better run-time behavior in high dimensional feature spaces. However, Zhao and Karypis showed that partitional methods can outperform hierarchical methods in terms of clustering quality [39]. Pedersen et al. apply the “Repeated Bisections” of [39], a hybrid approach that combines K-Means and hierarchical divisive clustering [27]. They cluster documents based on second order context vectors. These vectors are based on a bigram matrix that is reduced through Singular Value Decomposition. The authors of [8] introduce a three-step clustering algorithm that makes use of social network profiles from various networks and is in this respect in the spirit of our approach. In the first step the profiles are clustered and in the second step the result documents are clustered. Finally, the profile clusters are merged with the document clusters. In contrast to this approach, we avoid the noise of clustering by bootstrapping social profiles as clean seeds against which Web pages have to be matched.

In the evaluation section, we compare our methods with traditional Text Clustering techniques. The results show that the bootstrapping of the knowledge bases leads to results that are superior to those returned by unsupervised techniques.

3. WEB PAGE CLASSIFICATION

As stated earlier, our goal is to transform the clustering of search results to ambiguous person name queries into a classification task, by bootstrapping knowledge base entities about people with the same name. To this end we define the disambiguation task as follows:

For an ambiguous person name x, let $D_x = \{d_1, \ldots, d_n\}$ denote the set of retrieved documents to the query x, e.g., Google search results to the query x. Furthermore, for an entity source S let $E_x(S) = \{e_1, \ldots, e_m\}$ be the set of entity profiles in S that are referred to by the same name x, e.g., those entities could be retrieved from Wikipedia disambiguation pages or from a name search API in case of other entity sources, such as LinkedIn or Facebook. The task we address is the construction of a surjective mapping $m_x : D_x \rightarrow E_x(S) \cup \{e_{\text{noise}}\}$ such that a document $d \in D_x$ is mapped to $e \in E_x(S)$ if and only if $d$ is about $e$, and to a “noise entity” $e_{\text{noise}}$ if $d$ does not describe any of the entities in $E_x(S)$.

In the following, for the sake of a simpler notation, we omit the index x and the symbol S of the entity source and assume them to be implicitly given by the context. Also, for compactness, we use the term $E'$ to denote the extended entity profile set $(E' = E \cup \{e_{\text{noise}}\})$. The definition and construction of these noise profiles is explained in Section 3.3.

The above problem could be modeled as a graph partitioning problem, where entities and result pages would be connected by weighted edges representing their similarities. Figure 1 shows an example graph, where edges indicate differently weighted similarity relationships. Note that there is no edge between $e_1$ and $e_2$ ($e_1, e_2 \in E'$), because we apply a disjointness constraint between all entities from $E'$. The objective then would be to partition the graph into $|E'|$ components while minimizing the total weight of the edges between separate components. This problem is known to be NP-hard [49], hence we follow a relaxed version of this model, where we assign each document to the entity partition for which it exhibits the largest similarity or probability score. To this end, we propose a suite of vector space and probabilistic models, which can be analogously applied to address the above problem.

Note that Web page clustering enables the user to browse
the results in a more efficient way \cite{25}. However, it remains to mention that in general, a clustering of search results bears significant risks, since, if a result document was assigned to the wrong cluster, it would be difficult for the user to find it. To mitigate this risk, we propose to show the clusters in addition to the ranked search results as in \cite{19}.

If the user explores a specific entity cluster, corresponding to an entity \( e \in E' \), we first present all result documents \( d_j \in D \) for which \( m(d_j) = e \) holds and list subsequently all other result documents (i.e., sorted by a similarity score). Thus, the user has the chance to examine all related Web pages.

### 3.1 Vector Space Model

Let \( C' = D \cup E' \) denote the corpus of documents and profiles about entities that are referred to by the same name. Let \( F_c \) denote the features of a document \( c \in C' \). The feature space

\[
F = \bigcup_{c \in C'} F_c
\]

contains all features from \( C' \).

Being relatively succinct, an entity profile might miss many features that could also be salient for the corresponding entity. These missing features, however, could be found in other Web documents about the same entity. Hence, in our model the similarity score between an entity profile \( e \in E' \) and a result document \( d \in D \) is captured not only by the features that they actually have in common but also by the features that they might have in common if the profile was fully extended. More specifically, we define

\[
score(d, e) = \sum_{f \in F} w(f, e) \cdot w(f, d)
\]

where \( w \) should reflect the importance of feature \( f \) for an element \( c \in C' \) and can be modeled in different ways, e.g., as a metric distance, an information-theoretic similarity measure, a probabilistic measure, etc. We have analyzed the result quality for many different options and present the ones that turned out to be most promising in our experiments. Thereby, we used the well known cosine similarity \( (\text{sim}_{\text{cos}}) \) as a baseline scoring function.

Note that in general, the features could be of various types; they could be source dependent (e.g., semi-structured or multimodal sources suggest other features than unstructured sources), and they could also be more complex in nature by involving inter-document-links, n-grams \((n > 1)\), named entities, factual phrases, compound nouns, etc. However, for most online scenarios, the challenge is to exploit simple features and yet return high-quality clusters of results. The evaluation of our bootstrapping approach (see Section 4.3) shows that it is possible to rely on word-unigrams and still provide high-quality results. However, especially semi-structured information sources might additionally provide richer features like factual phrases regarding particular attributes (i.e., research topics and working places in LinkedIn profiles) and thus enhance the result quality \cite{25}.

Finally, the mapping function \( m : D \rightarrow E' \) assigns a document \( d \in D \) to the entity profile \( e \in E' \) that maximizes the similarity score \( score(d, e) \) \( (\text{sim}_{\text{cos}}(d, e)) \), respectively.

### Measures for the Weight Function \( w \)

As mentioned above, the similarity between an entity profile and a result document is measured over the features they have or might not have in common. These kinds of explicit and implicit similarities are incorporated into our model by different choices of the weight function \( w \).

A popular measure for the importance of a feature for a given document is the \( \text{tf-idf} \) measure. For the explicit similarity between entity profiles and result documents we define the weight measure for a feature \( f \in F \) and an element \( c \in C' \) simply as

\[
w(f, c) = \text{tfidf}(f, c)
\]

where implementation-wise we have chosen

\[
\text{tfidf}(f, c) = \frac{\text{freq}(f, c)}{\text{max}_{f' \in F_c} \text{freq}(f', c)} \cdot \log \left( \frac{|F_c|}{|\{c' \in C : f \in F_{c'}\}|} \right).
\]

For measuring the implicit similarity between entity profiles and result documents, we define the smoothed weight function for the importance of a feature \( f \in F \) for an entity profile \( e \in E' \) as

\[
w'(f, e) = |\text{tfidf}(f, e)| + \sum_{d \in D} \text{sim}_{\text{cos}}(e, d) \cdot |\text{tfidf}(f, d)|
\]

where \( \text{sim}_{\text{cos}} \) stands for the cosine similarity between the weighted feature vector of the entity profile \( e \) and that of the document \( d \), and \( |\text{tfidf}| \) represents the L1-normalized \( \text{tfidf} \) score. The final smoothed similarity score is given by:

\[
score'(d, e) = \sum_{f \in F} w'(f, e) \cdot w(f, d)
\]

The intuition behind this measure is that the normalized \( \text{tfidf} \)-based feature vector representing the entity profile is first ‘pulled’ towards similar document vectors and then its (inner-product-based) similarity to the document vector is computed. This method is similar in spirit to the Rocchio algorithm for modifying the original query vector when relevance feedback is available \cite{23}. Note that the L1-normalization allows us to move the entity profile vector fraction-wise towards similar document vectors. This accounts for a careful (and rather conservative) modification of the original entity profile vector. The outcome of the final inner-product is proportional to the cosine similarity between the modified vector and the document vector.

### 3.2 Probabilistic Models

We now discuss the application of probabilistic models to our classification problem. The mapping \( m : D \rightarrow E' \) maps a document \( d \in D \) to the entity profile \( e \in E' \) that maximizes the joint probability \( p(d, e) \).

By applying the chain rule and assuming conditional feature independence for a given entity-profile, we can derive a Bernoulli Naïve Bayes model \( \hat{p}_B \)

\[
\hat{p}_B(e, d) = p(e) \cdot \prod_{f \in F_d} p(f|e)
\]

where \( p(e) \) is a prior describing the prominence of the entity represented by the profile \( e \), and \( p(d|e) \) captures the plausibility of the document \( d \) being generated from the entity profile \( e \). As an alternative, we consider a Multinomial
Naïve Bayes model \((\hat{p}_M)\), which takes feature occurrence frequencies into account:

\[
\hat{p}_M(e, d) = \frac{|d| \cdot p(e) \cdot \prod_{f \in F_d} \frac{p(f|e)^{\text{freq}(f, d)}}{\text{freq}(f, d)!}}
\]

where \(\text{freq}(f, d)\) represents the absolute frequency of the feature \(f\) in the document \(d\), so that \(\sum_{f \in F_d} \text{freq}(f, d) = |d|\).

**Parameter Estimation**

Due to the confined nature of the feature set given a specific entity profile, simple maximum likelihood estimation of the conditionals \(p(f|e)\) would not be appropriate and lead to underestimations. Furthermore, the model would be prone to numerical effects, especially for cases where \(f \notin F_e\), i.e., the feature \(f\) does not occur in the entity profile \(e\). A possible solution to this problem is the extension of the feature set of \(e\) with features from the documents similar to the actual entity \([13, 21, 32]\). We implemented models that extended the feature sets with features from the top-\(k\) documents that are “closest” to \(e\) (e.g., in terms of cosine similarity, Jaccard distance, etc.) and experimented with different values for \(k\). However, the results were fairly disappointing (both in terms of precision and recall measures). Seemingly, the model could not handle the noise introduced by the extension of the feature set. Much better results were achieved by following the simple smoothing techniques.

**Laplace-smoothing** (also referred to as additive smoothing) adds a smoothing factor \(\alpha\) to the actual relative frequency of each feature. Thus, the prior is estimated by:

\[
\hat{p}_L(e) = \frac{\sum_{f \in F_e} w(f, e)}{\sum_{e_j \in E} \sum_{f_j \in F_e} w(f_j, e_j)} + \alpha
\]

and the likelihood is defined as:

\[
\hat{p}_L(f|e) = \frac{\sum_{f \in F_e} w(f, e)}{\sum_{f \in F_e} w(f, e)} + \alpha
\]

In our experiments, a smoothing parameter \(\alpha = 0.01\) empirically showed best results.

Another popular smoothing method, the Jelinek-Mercer-smoothing, uses a background model (based on corpus frequencies) to estimate the likelihood of non-occurring features. It is defined as:

\[
\hat{p}_\lambda(f|e) = (1 - \lambda) \hat{p}_{\text{M}}(f|F_e) + \lambda p(f|F)
\]

It can be shown that by setting \(\lambda \) to 0.5, one can derive a \(\text{tfidf}\)-style smoothing, which we used in our implementation. In our experiments we found that the Bernoulli Naïve Bayes model worked best with Laplace smoothing \((\hat{p}_L)\), while the Multinomial model worked best with the Jelinek-Mercer smoothing \((\hat{p}_\lambda)\). We report the corresponding results in Section 4.3.

### 3.3 Modeling the Noise Entity Profile

In the definition of our mapping \(m\) we introduced an artificial entity profile \(e_{\text{noise}}\), to which documents should be mapped if they do not match any of the entity profiles in \(E\). This definition accounts for the fact that the set of unique entities having the same name is limited by the underlying boosting source. Hence, result documents that do not correspond to any of the entity profiles from \(E\) are assigned to the artificial profile \(e_{\text{noise}}\). There are different ways to model such a noise profile; in general, however, it should contain rather uninformative features, e.g., features with low expected information gain or features with high \(df\) values. We tested various approaches, but for the sake of brevity, we present the two best performing and most robust ones.

As a first approach we consider the **union-noise entity profile**, denoted by \(e_{\text{noise}}\). The profile is generated in a straightforward manner by equally weighting all entity features.

\[
F_{\text{union-noise}} = \{f|f \in F_e, e \in E\}
\]

This method aims at maximizing the feature noise in the artificial profile.

In addition to \(e_{\text{union-noise}}\), we introduce the **intersection-noise entity profile**, denoted by \(e_{\text{intersection-noise}}\). It contains all features (equally weighted) occurring in the intersection of any entity profile \(e\) with any document \(c\) from the corpus:

\[
F_{\text{intersection-noise}} = \{f|\forall e \in E, \forall c \in C, e \neq c : f \in F_e \cap F_c\}
\]

where \(C = C' \setminus \{e_{\text{noise}}\}\). Note that the above definition of the noise entity is biased towards features with high \(df\) values (i.e., non-specific features), but may still contain slightly informative features, thus mitigating a rigid discrimination between the noise entity and the other entity profiles.

We have evaluated the effect of both approaches on the mapping quality and show the results in the next section.

### 4. EVALUATION

The problem of clustering search results to ambiguous person name queries has been studied in prior work and it is no surprise that there are various publicly available evaluation datasets, e.g., SIGIR’05 [4], WWW’05 [7], WePS-2 [8], etc. However, we found that available datasets are relatively small for conclusive statements on clustering quality. Furthermore, for the quality evaluation of our approach we needed a manual alignment of the search results with entity profiles from a given entity source. Such an alignment was not available in any of the datasets. Hence we decided to create a larger dataset, which would provide the required alignments. While an obvious application for our techniques is to use profile pages from social networks, e.g., LinkedIn, the general terms of agreement of those networks currently do not allow such usage. Therefore, we extracted Wikipedia entities with ambiguous names; Web search results for those names were manually aligned with the corresponding articles (which we viewed as entity profiles). To this end we tested more than 85k possible entity – Web page combinations for correctness.

#### 4.1 Dataset

This subsection describes the content of the evaluation dataset, which we make available[1].

In Wikipedia, articles about persons with ambiguous names are manually maintained and organized in so-called “Human Name Disambiguation Pages” (HDPs for short). Each of these pages is dedicated to an ambiguous person name and contains links to Wikipedia articles about persons of that name. Some HDPs contain more links than others, reflecting

[1] http://hpi-web.de/naumann/projekte/repeatability/datasets/wpsd.html
the fact that on Wikipedia some names are more ambiguous than others. Figure 2 depicts the distribution of links over the HDPs. As of July 2012 there were 38,692 HDPs on Wikipedia containing a total of 213,564 links referring to Wikipedia articles. Interestingly, there were 28 HDPs that did not contain any link (e.g., wiki:Sharon_Allen) and 328 HDPs contained only one valid link (e.g., see wiki:Luciano_Buonaparte). These articles were ignored. A careful stratified random sampling procedure on the remaining 38,366 HDPs, gave us the final dataset \( X \) with HDPs sampled from three different regions of the distribution (i.e., A, B, C shown in Figure 2). The resulting sample contained HDPs that were evenly spread across the three regions. In this way, a total of 50 HDPs were extracted: 17 from region A (each with 24 linked articles or more), 17 from region B (containing 10+ linked articles), and 16 from the remaining long tail C of the distribution.

![Figure 2: Distribution of links from HDPs on a log-log scale. The area under the curve is divided into three regions: (A) HDPs containing highly ambiguous names with more than 24 linked articles each; this region contains \( \sim 10\% \) of all article links in all HDPs. (B) HDPs containing ambiguous names with at least 10 linked articles each, corresponding to \( \sim 23\% \) of all article links, and (C) HDPs with moderately ambiguous names, containing \( \sim 66\% \) of all article links.](http://en.wikipedia.org/w/api.php)

Given an HDP from the sample \( (X) \), the ambiguous person name \( x \) can be directly inferred from the url, e.g., wiki:John_Campbell is translated to \( x \leftarrow \) "John Campbell", ignoring Wikipedia notations, such as `(disambiguation)`. Additionally, the following five steps were performed:

1. The content of the HDP was extracted using Wikipedia’s MediaWiki RESTful API.
2. The set \( E \) of Wikipedia articles referenced from the HDP was extracted.
3. The textual content of each article in \( E \) was parsed, tokenized, and cleaned (from any markups) to derive unigram features.
4. A Google Custom Search (using the Google API) was performed for the person name \( x \) derived from the HDP’s url, whereas Wikipedia articles in the results were excluded, e.g., for \( x \leftarrow \) "John Campbell" the Google query “John Campbell -site:wikipedia.org” was issued.

Table 1: Overview of the manually labeled evaluation dataset.

| HDP title | #wiki articles | #result pages | alignments |
|-----------|---------------|---------------|------------|
| A         | John Campbell | 100           | 96         | 33 → 6   |
|           | James White   | 53            | 97         | 36 → 6   |
|           | John Clark    | 42            | 96         | 4 → 4    |
|           | John Rogers   | 42            | 93         | 28 → 7   |
|           | William Murray| 38            | 91         | 24 → 8   |
|           | David Young   | 34            | 96         | 11 → 4   |
|           | Robert Miller | 34            | 97         | 16 → 6   |
|           | James Baker   | 33            | 97         | 3 → 3    |
|           | John Mason    | 32            | 98         | 12 → 5   |
|           | Thomas Baker  | 31            | 93         | 1 → 1    |
|           | James Ross    | 30            | 96         | 7 → 5    |
|           | Michael Kelly | 29            | 92         | 14 → 2   |
|           | David Thompson| 28            | 97         | 26 → 5   |
|           | James Gordon  | 28            | 90         | 31 → 4   |
|           | John Richards | 26            | 94         | 0 → 0    |
|           | Thomas Bell   | 26            | 89         | 6 → 5    |
|           | William Collins| 24          | 87         | 27 → 5   |
| B         | Maria Theresa | 19            | 91         | 31 → 1   |
|           | Mark Robinson | 16            | 94         | 6 → 3    |
|           | Michael Walsh | 16            | 95         | 44 → 4   |
|           | Danny Williams| 13            | 92         | 37 → 4   |
|           | David Brooks  | 13            | 97         | 86 → 1   |
|           | Fred Johnson  | 13            | 84         | 1 → 2    |
|           | George Stephens| 13           | 90         | 1 → 1    |
|           | Gustav of Sweden| 13      | 81         | 27 → 6   |
|           | Henry Robinson| 13            | 95         | 1 → 1    |
|           | John MacArthur| 13            | 99         | 97 → 5   |
|           | Billy Hayes   | 11            | 91         | 49 → 4   |
|           | Drogó         | 11            | 93         | 77 → 5   |
|           | Kimberly Smith| 11            | 97         | 14 → 4   |
|           | William Hammond| 11         | 95         | 20 → 6   |
|           | Bob Hope      | 10            | 90         | 78 → 5   |
|           | John Peppin   | 10            | 95         | 31 → 4   |
|           | Robert Dean   | 10            | 92         | 20 → 2   |
| C         | Michael Myers | 8             | 74         | 44 → 3   |
|           | Edward Cole   | 6             | 94         | 5 → 2    |
|           | Peter Donnelly| 6             | 98         | 31 → 5   |
|           | Siemowit of Masovia| 5       | 99         | 10 → 3   |
|           | William McKee | 5             | 98         | 2 → 1    |
|           | Brian Reynolds| 4             | 95         | 14 → 1   |
|           | Leslie Baker  | 4             | 93         | 9 → 2    |
|           | George Donnelly| 3           | 99         | 4 → 2    |
|           | Peter Robertson| 3           | 97         | 2 → 1    |
|           | Timothy Allen | 3             | 92         | 29 → 2   |
|           | Daniel Terra  | 2             | 100        | 15 → 1   |
|           | Frida Snevson | 2             | 98         | 54 → 1   |
|           | Jack Renner   | 2             | 95         | 24 → 2   |
|           | Lasse Nielsen | 2             | 91         | 8 → 2    |
|           | William McFaddin| 2           | 96         | 8 → 2    |
|           | William Nicol | 2             | 97         | 3 → 1    |

5. The top-100 results were retrieved (ignoring empty, non-textual, and HTTP error pages), e.g.:
   (1) [http://yelp.com/biz/john-campbells-irish-bakery-san-francisco-4](http://yelp.com/biz/john-campbells-irish-bakery-san-francisco-4)
   (2) [http://en.wikipedia.org/w/api.php](http://en.wikipedia.org/w/api.php)
   (3) [http://economics.ag.utk.edu/campbell.html](http://economics.ag.utk.edu/campbell.html)
   (4) [http://www.facebook.com/pages/John-Campbell-Yoga/125627657452140](http://www.facebook.com/pages/John-Campbell-Yoga/125627657452140)

Subsequently the same procedure as in Step 3 was applied to derive features from each of the documents in the result set.

To create the gold standard for the evaluation, for each query the alignment of search results with Wikipedia articles was carefully performed by human labelers, namely students from our department. Each document in the result set was either assigned to exactly one Wikipedia article (referenced from the HDP from which the query was derived) or labeled as noise document if it was not about any of the entities described by the Wikipedia articles. Also, in cases in which
multiple Wikipedia entities occurred in the result document, the document was labeled as noise document. This process added up to more than 85k possible Web page to entity (non-noise) combinations (i.e., \(\sum |E_x| \times |D_x|\)) that had to be checked manually.

Table 1 gives an overview of the manually aligned dataset. One alignment task is identified by the HDP title, the Wikipedia articles referenced from the HDP (#wiki articles), and the remaining Web pages from Google’s top-100 results after excluding empty, non-textual, and HTTP-error pages (#result pages). The rightmost column in Table 1 (alignments) shows the number of result pages (#pages) that could be assigned to any Wikipedia article referenced from the HDP and the number of different Wikipedia articles (#articles), i.e., ground-truth classes, to which result pages were assigned. The remaining search results that could not be aligned with any Wikipedia article referenced from the HDP as well as the ones that were about multiple individuals were assigned to a designated class \(e_{\text{noise}}\).

A common but misleading assumption that is based on the notability of Wikipedia entities is that searching for Wikipedia entity names yields many top results related to the corresponding Wikipedia entities. This assumption holds when the documents about Wikipedia entities are also popular on the Web/Google (which is often the case), but for niche Wikipedia entities, which are known to few scholars, this assumption leads astray. For instance, “John Campbell” refers to 100 different individuals in Wikipedia, but only 6 of them actually occurred in Google’s top-100 results (after excluding Wikipedia-related results). In fact, this skew is the case for the majority of the ambiguous names in our dataset. Also note that the classification problem is extremely difficult: for the ambiguous name “John Campbell” the problem is to automatically classify 96 Web pages into one of 100 entities (+\(e_{\text{noise}}\)), while the ground truth tells us that only 33 of 96 results are assigned to 6 out of 100 Wikipedia articles and the rest to \(e_{\text{noise}}\). A more extreme example for this observation is “John Richards”, for which no valid mapping between the top-100 search results and any of Wikipedia articles referenced from the corresponding HDP was found. Hence using Wikipedia to bootstrap the grouping of search results to ambiguous person names can be quite challenging and also covers the problem of clustering documents about less famous people on the Web. As stated earlier, the entities of \(e_{\text{noise}}\) are not classified with respect to the entity source and are thus unclustered.

4.2 Comparison with clustering techniques
This section provides a comparison of state-of-the-art clustering algorithms to our techniques. The results demonstrate the advantages of the proposed bootstrapping approach, which exploits prior knowledge to perform the disambiguation task.

For the comparison, we selected two of the most popular clustering methods: Hierarchical Agglomerative Clustering (HAC), and K-Means, as provided by the Weka machine learning toolkit\(^6\). For each of the methods, we tested many different configurations. For example, for fairness reasons, one of the configurations of K-Means used the Wikipedia entities as initial centroids similarly as proposed in \(^6\). However, random seeds\(^7\) led to better results than the above configuration.

For the HAC method we tried different linkage configurations, e.g., single-link, average-link, complete-link \(^2\). The configuration with the complete linkage criterion worked best and was chosen as a competitor. For both methods we tested configurations in which the parameter \(k\) of resulting clusters was set to the number of different Wikipedia entities for the corresponding ambiguous name.

For comparison with the above methods we used an implementation of \(\text{score}'\) and \(\hat{p}_{\text{X}}^J\). Both methods were applied using the intersection-noise entity \((e_{\text{noise}})\) to derive a mapping as proposed in Section 3. The mapping results were transformed to anonymous clusters by omitting the assigned Wikipedia entity label.

To be fair, for the quality evaluation of the groups returned by the clustering methods, result documents that were not related to any of the Wikipedia entities (these would be noise documents for our bootstrapping approach) were not taken into account. The reason is that the clustering algorithms treat such documents equally to all the others, thus missing the task of creating a coherent “noise cluster” (i.e., with documents assigned to \(e_{\text{noise}}\)). This led to 1,095 Web documents used for the clustering evaluation.

To compare the clustering performance of the approaches we applied two established measures: purity and normalized mutual information (NMI) \(^{24}\). Table 2 shows the average purity and NMI values over all 50 names from \(X\). As can be seen, all three bootstrapping-based classification models outperform the clustering approaches with respect to both measures. However, the differences for the purity values are smaller than for the NMI values.

\begin{table}[h]
\centering
\begin{tabular}{|c|cc|}
\hline
\textbf{Method} & \textbf{Purity} & \textbf{NMI} \\
\hline
\textbf{score}' & 0.913 & 0.560 \\
\hat{p}_{\text{X}}^J & 0.890 & 0.492 \\
HAC & 0.829 & 0.321 \\
K-Means & 0.814 & 0.287 \\
\hline
\end{tabular}
\caption{Clustering evaluation of documents related to a Wikipedia article.}
\end{table}

The purity is a commonly used evaluation measure in the area of personal name resolution. However, a high purity is easy to achieve with a large number of clusters (i.e., a score of 1 can be achieved by turning every document into one cluster). The results for the clustering methods align with the evaluation of Balog et al. \(^6\). This shows that even the subset of our evaluation dataset (i.e., Web pages related to Wikipedia entities) is representative for the evaluation of a personal name resolution task.

The NMI measure shows larger deviations, because it is normalized by the overall entropy across clusters (which requires clusters not only to possibly contain elements from only one class, but also to possibly contain all the elements from that class). In terms of NMI score, all two bootstrapping approaches outperform the unsupervised methods by a large margin. This shows that our approach is able to balance between quality and size of clusters.

However, note that although this evaluation shows promising results for the grouping of this subset of documents, the actual problem covered in this work also depends on another difficult subtask, namely the identification of documents not related to any entity of the bootstrapping source. The fol-
lowing section discusses the evaluation of our approaches for the task of linking Web documents to ambiguous Wikipedia entities and thus covers the complete problem.

4.3 Probabilistic vs. vector space models

A commonly used measure to evaluate the performance of binary categorization methods is the $F_1$ measure. It is defined as the harmonic mean of precision and recall and mitigates the influence of large precision or recall values. For our multi-class classification problem, we apply the micro- and macro-averaged $F_1$ ($\text{micro}(F_1)$ and $\text{macro}(F_1)$, respectively) \cite{6}, which also range between 0 and 1. By the $\text{micro}(F_1)$ all documents evenly weighted, thus giving large classes a higher chance to dominate the overall score. On the other hand, the values of $\text{macro}(F_1)$ tends to be biased towards the performance of the classifier with respect to rare classes, because in this case the classes are weighted uniformly.

|          | no noise | union-noise | intersection-noise |
|----------|----------|-------------|--------------------|
| $\text{sim}_{\text{cos}}$ | 0.233    | 0.438       | 0.601              |
| score    | 0.210    | 0.704       | 0.701              |
| score'   | 0.322    | 0.569       | 0.734              |
| $\hat{p}_B^C$ | 0.251    | 0.634       | 0.642              |
| $\hat{p}_M^C$ | 0.257    | 0.486       | 0.730              |

Table 3: Performance comparison between different mapping functions for different noise entity profiles ($\mathcal{F}_1$ scores)

Table 3 compares the performance of different mapping functions and configurations for the noise entity. Each row shows the performance of one of the membership scoring functions described in Sections 3.1 ($\text{sim}_{\text{cos}}, \text{score}, \text{score}'$) and 3.2 ($\hat{p}_B, \hat{p}_M^C$). The scoring function $\text{sim}_{\text{cos}}$ serves as a baseline and refers to a simple application of the cosine similarity on the weighted features defined in Section 3.1. Each column shows the influence of different configurations for the noise entity: a union-noise entity ($\epsilon_{\text{union}}$), an intersection-noise entity ($\epsilon_{\text{intersection}}$) (see Section 3.3), and a “no noise entity” configuration. The “no noise entity” column stands for a configuration without any noise entity (i.e., $E' = E$). The values in the cells represent the micro and macro $F_1$ scores averaged over all 50 name tasks from the previously introduced sample $\mathcal{X}$ of the evaluation dataset. More specifically:

$$\mathcal{F}_1 = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \frac{\text{micro}(F_1) + \text{macro}(F_1)}{2}$$

The influence of the noise entity profiles on the performance is notable. The configurations with the union-noise entity and the intersection-noise entity considerably outperform the “no noise” configurations. This finding highlights the importance of mechanisms that can deal with the presence of search results that are not related to any of the source entities. Especially in our approach the noise entities are crucial for the final grouping of search results, since the bootstrapped source is of limited scope and the algorithms have to handle the open-world assumption (i.e., with results about individuals that do not occur in the underlying entity source, Wikipedia).

However, the configurations with the union-noise entity performed worse than the model represented by the mapping function $\text{score}$ (i.e., the model that quantifies the dot-product-based similarity between a result document and an entity profile). We hypothesize that this is due to the fact that the larger union-noise entity introduces too much noise for the smoothed scoring function ($\text{score}'$, which modifies the original vectors to capture the implicit similarity between them) and also leads to a degraded performance of the Multinomial Naïve Bayes model ($\hat{p}_M$), which relies on multiple occurrences of features) since in the union-noise entity every feature occurs only once. This effect is lower for the more carefully constructed and generally smaller intersection-noise entity.

A more detailed analysis of the performance of different mapping functions and configurations for the noise entity profile is illustrated in Figure 3. Again, each row shows the performance of a membership scoring functions described in Sections 3.1 ($\text{sim}_{\text{cos}}, \text{score}, \text{score}'$) and 3.2 ($\hat{p}_B, \hat{p}_M^C$). The columns show the influence of the different noise entity profiles. Moreover the charts in each cell illustrate the $\text{micro}(F_1)$ (y-axis) and $\text{macro}(F_1)$ (x-axis) of a configuration given all 50 names from the previously introduced evaluation dataset.

As one can see, the influence of the noise entity profiles on the performance is notable. Both configurations with noise profile ($\epsilon_{\text{union}}$ and $\epsilon_{\text{intersection}}$) considerably outperform the configurations without noise profiles. Specifically the differences in the $\text{micro}(F_1)$ values are remarkable. This is due to the fact that the micro averaged $F_1$ favors larger classes, which, given the limited scope of our bootstrapping source (i.e., Wikipedia), are often the noise profiles. Indeed Wikipedia contains only information about notable people; hence, there are many result documents about people having the same names as the Wikipedia entities, which have to be assigned to the noise profiles.

The configurations with the intersection noise profile entities ($\epsilon_{\text{intersection}}$) seem to outperform the union noise profile ($\epsilon_{\text{union}}$) in all cases but for $\text{score}$ and $\hat{p}_B$ where both noise profiles perform equally good.

Next, the influence of the proposed feature smoothing solutions on the results is compared. To this end, the basic vector-space similarity model $\text{score}$ is compared to its more advanced version $\text{score}'$ (which captures implicit similarities between documents and entity profiles, see Section 3.1). First of all, both models outperform the simple cosine similarity version ($\text{sim}_{\text{cos}}$) when a noise profile is available. This means that both versions are better suited to bootstrapping sources of limited scope. Furthermore, $\text{score}'$ outperforms $\text{score}$ for intersection noise profile, whereas $\text{score}$ seems to perform slightly better with the union noise profile (i.e., $\text{micro}(F_1)$). This is because the combination of the union noise profile with the implicit similarity function $\text{score}'$ introduces too much noise.

Next, the performance of the Bernoulli Naïve Bayes model with Laplace smoothing ($\hat{p}_B$) is compared to the Multinomial Naïve Bayes model with Jelinek-Mercer smoothing ($\hat{p}_M^C$) (see Section 3.2). Again, both approaches are better than $\text{sim}_{\text{cos}}$ in case of a provided noise profile entity and are on a par for “no noise entity”. One can also see, that the $\hat{p}_B$-estimator works better than $\hat{p}_M^C$ in combination with $\epsilon_{\text{union}}$. However, $\hat{p}_M^C$ clearly outperforms $\hat{p}_B$ in combination with $\epsilon_{\text{intersection}}$, thus achieving the best overall performance among the probabilistic models. We hypothesize that the advantage of the Multinomial model (i.e., taking multiple occurrences
Figure 3: Performance comparison between different algorithm configurations. In row (a) \( \text{sim}_\text{cos} \) – cosine similarity, (b) \( \text{score} \) – scoring function, (c) \( \text{score}' \) – smoothed scoring function, (d) \( \hat{p}_L \) – Bernoulli Na"ive Bayes model with Laplace smoothing, and (e) \( \hat{p}_M \) – Multinomial Na"ive Bayes model with Jelinek-Mercer smoothing; in the left-most column the configurations with no noise entity, in the middle column, the union noise profile (\( \varepsilon \cup \text{noise} \)), and in the right column the intersection noise profile configurations (\( \varepsilon \cap \text{noise} \)).

of features into account) is diminished by the larger \( \varepsilon \cup \text{noise} \), in which all the features occur only once. This impact is much smaller in the case of \( \varepsilon \cap \text{noise} \).

The best results are thus achieved by \( \text{score}' \) and \( \hat{p}_M \) estimator in combination with the \( \varepsilon \cap \text{noise} \). Furthermore, the baseline algorithm \( \text{sim}_\text{cos} \) performs best with \( \varepsilon \cap \text{noise} \) too.

5. CONCLUSION

The focus of this work has been on the design of efficient and high-quality approaches to the problem of cluster-
ing search results to ambiguous person-name queries. The proposed methods build on the idea that Web 2.0 profiles, can be bootstrapped to cast the above problem into a classification problem, where results are mapped to the most similar profile. The proposed method does not influence the grouping of entities not represented in the knowledge base, which might be clustered subsequently. With a steady increase of information about people and other entities in social networks and other Web 2.0 sources, we are confident that methods like the ones presented in this paper will gain importance for reliably disambiguating search results.

The suite of presented and evaluated methods covers a wide range of advanced vector-space and probabilistic models. Efficient similarity and likelihood estimations that do not compromise the classification quality have been a primary goal of this work. Furthermore, dealing with noisy and biased data from the Web documents as well as the knowledge base was essential for the introduced approach. Specifically, handling the incompleteness of the entity source (open world assumption) was as well in the focus of this work.

The provided experiments were based on a hand-labeled dataset over more than $5k$ alignment candidates of around 5,000 Web pages on ambiguous person names that we have made publicly available. Although all methods deliver satisfactory results, in light of the experimental outcome, we would favor the smoothed vector space model implementing score$^*$. For a definitive answer all methods would have to be evaluated on multiple datasets.

Note that all presented methods can be analogously applied to a broader online scenario, where for a given ambiguous query, Web results and appropriate entity profiles would be retrieved in parallel and search results mapped to the most similar profile. Although there would be a moderate increase in latency due to online indexing of unigrams in documents and profiles, we expect the result quality to outweigh the runtime latencies.

As part of our future work, we are aiming to aggregate the results for different Web 2.0 sources (e.g., LinkedIn, Facebook, etc.) to improve the grouping of search results to ambiguous queries. To this end, more complex features (e.g., multigrams, structured attribute-value pairs, etc.) could boost our methods further. An incremental classification of search results is an additional point on our agenda. One could first map only those result documents to profiles for which the similarity exceeds some threshold and then given the gained information, hopefully a better classification of the remaining documents would be achieved. Obviously, there is high value in the information accessible through Web 2.0 APIs and future Web search should aim at better exploiting this information.

6. REFERENCES

[1] J. Artiles, A. Borthwick, J. Gonzalez, S. Sekine, and E. Amigo. WePS-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In CLEF (Notebook Papers/LABs/Workshops), 2010.

[2] J. Artiles, J. Gonzalez, and S. Sekine. The senseval-2007 weps evaluation: Establishing a benchmark for the web people search task. In Proceedings of the International Workshop on Semantic Evaluations (SemEval), pages 64–69, 2007.

[3] J. Artiles, J. Gonzalez, and S. Sekine. WePS-2 evaluation campaign: overview of the web people search clustering task. In Web People Search Evaluation Workshop (WePS), International World Wide Web Conference (WWW), 2009.

[4] J. Artiles, J. Gonzalez, and F. Verdejo. A testbed for people searching strategies in the WWW. In Proceedings of the International Conference on Information Retrieval (SIGIR), pages 569–570, 2005.

[5] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In Proceedings of the Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics (ACL), pages 79–85, 1998.

[6] K. Balog, L. Azzopardi, and M. de Rijke. Resolving person names in web people search. In Wearing Services and People on the World Wide Web, International World Wide Web Conference (WWW), pages 301–323, 2008.

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

[8] R. Berendtsen, B. Kovachev, E.-P. Nastou, M. de Rijke, and W. Weerkamp. Result disambiguation in web people search. In Proceedings of the European Conference on Advances in Information Retrieval (ECIR), pages 146–157, 2012.

[9] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 9–16, 2006.

[10] P. Christen. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer Publishing Company, Incorporated, 2012.

[11] P. Christen. A survey of indexing techniques for scalable record linkage and deduplication. IEEE Transactions on Knowledge and Data Engineering (TKDE), 24(9):1537–1555, 2012.

[12] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 708–716, 2007.

[13] M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short texts through document expansion. In Proceedings of the International Conference on Information Retrieval (SIGIR), pages 911–920, 2012.

[14] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering (TKDE), 19(1):1–16, 2007.

[15] A. Fader, S. Soderland, and O. Etzioni. Scaling Wikipedia-based named entity disambiguation to arbitrary web text. In Proceedings of the WikiAI - IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy, International Joint Conference on Artificial Intelligence (IJCAI), 2009.
[16] I. P. Fellegi and A. B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[17] J. Hassell, B. Aleman-Meza, and I. B. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 44–57, 2006.

[18] M. Ikeda, S. Ono, I. Sato, M. Yoshida, and H. Nakagawa. Person Name Disambiguation on the Web by Two-Stage Clustering. In *Proceedings of the International World Wide Web Conference (WWW)*, 2009.

[19] D. V. Kalashnikov, S. Mehrotra, Z. Chen, R. Nuray-Turan, and N. Ashish. Disambiguation algorithm for people search on the web. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1258–1260, 2007.

[20] H. Köpcke, A. Thor, and E. Rahm. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2):484–493, 2010.

[21] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of the International Conference on Information Retrieval (SIGIR)*, pages 186–193, 2004.

[22] C. Long and L. Shi. Web person name disambiguation by relevance weighting of extended feature sets. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[23] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the Conference on Natural language learning (CONLL)*, pages 33–40, 2003.

[24] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, 2008.

[25] F. Naumann and M. Herschel. *An Introduction to Duplicate Detection*. Morgan and Claypool Publishers, 2010.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

[27] T. Pedersen, A. Purandare, and A. Kulkarni. Name discrimination by clustering similar contexts. In *Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 3406 of *Lecture Notes in Computer Science*, pages 226–237. Springer-Verlag, 2005.

[28] S. Sekine and J. Artiles. WePS2 attribute extraction task. In *Web People Search Evaluation Workshop (WePS)*, *International World Wide Web Conference (WWW)*, 2009.

[29] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the International Conference on Information Retrieval (SIGIR)*, pages 208–215, 2000.

[30] E. Smirnova, K. Avrachenkov, and B. Trousse. Using web graph structure for person name disambiguation. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[31] A. Spink, B. Jansen, and J. Pedersen. Searching for people on web search engines. *Journal of Documentation*, 60:266 – 278, 2004.

[32] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 407–414, 2006.

[33] Q. M. Vu, T. Masada, A. Takasu, and J. Adachi. Disambiguation of people in web search using a knowledge base. In *Proceedings of the International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies (RIVF)*, pages 185–191, 2007.

[34] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 163–170, 2005.

[35] R. Watrigant, M. Bougeret, R. Giroudeau, and J.-C. König. On the Approximability of the Sum-Max Graph Partitioning Problem. In *International Workshop on Approximation, Parameterized and EXact algorithms (APEX)*, page 3, Paris, France, 2012.

[36] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999.

[37] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of the International Conference on Information Retrieval (SIGIR)*, pages 46–54, 1998.

[38] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks*, 31(11-16):1361–1374, 1999.

[39] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 515–524, 2002.