Bootstrapping Wikipedia to Answer Ambiguous Person Name Queries

Toni Gruetze, Gjergji Kasneci, Zhe Zuo, Felix Naumann
Hasso Plattner Institute
Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany
Email: {firstname.lastname}@hpi.uni-potsdam.de

***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 in question 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 Wikipedia, LinkedIn, Facebook, etc. Being human-generated, the information on these platforms is clean, focused, and already disambiguated. We show that when searching with ambiguous person names the information from Wikipedia 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.

I. 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, or locations. The common way by which modern search engines approach the ambiguity problem is to diversify search results and hope 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 [1], 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 U.C. Berkeley, is ranked 18^{th} in the result list[1]. 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 ambiguous person-related queries (e.g., "Michael Jordan"), with different persons of the same name and the same category, say "computer scientists" (DBLP alone lists 3 Michael Jordans). Under such noisy conditions typical clustering techniques, such as K-Means or Hierarchical Agglomerative Clustering (HAC), are shown to perform rather moderately [2]. We have evaluated the performance of these methods on our dataset (see Section IV-A), and found that although they perform well in terms of purity, 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.

Our approach to the ambiguous person search problem can bootstrap the information of a knowledge base (i.e., Wikipedia) 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, we also observe a continuous growth of Web 2.0 sources, which might render them adequate for addressing many ambiguity problems on the Web.

In this paper, we analyze the result quality of different

[1] The second hit about the researcher is ranked 47^{th}.

efficient information retrieval and machine learning strategies to solve the above problem and show that information bootstrapped from Wikipedia 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, noise may arise from single result pages containing information about different people or about entities 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[2] retrieved from Google queries on 50 ambiguous person names.

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

II. Background and Related Work

Entity disambiguation is a broad topic and spans several well-studied research fields in computer science. 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.

The field of **entity resolution** (ER) (also referred to as record linkage or duplicate detection) is already summarized elsewhere [3], [4]. 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, Cucerzan transforms the named entity disambiguation problem to a ranking problem of Wikipedia articles [5]. He ranks these Wikipedia entities for a given mention according to a the cosine similarity between the textual context of the mention and the article content as well as its categories. 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 [6]. These techniques address a problem similar to ours. However, in contrast we try to find the main entity of a text instead of dozens of entity mentions per document.

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 coreference resolution, i.e., to find referents across multiple documents [7]. One of the most challenging tasks here is the disambiguation of search results to ambiguous person names [8], also referred to as personal name resolution.

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 [7], [9], [10]. The features are typically weighted by their *tf·idf* scores. More advanced features can be considered, such as named entities, noun phrases, intrinsic hyper-link relationships among Web pages, and detailed personal or biographical features [11], [12]. However, Balog et al. compare the performance of a simple bag-of-word 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 [13].

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) [9], [12]. Bekkerman and McCallum provide an agglomerative/conglomerative double clustering method [10]. It can be shown that this technique is related to the information bottleneck method, which is known to perform well for text clustering tasks [14]. It is a widespread belief hat 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 [15]. The authors of [9] introduce a three-step clustering algorithm, which 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 that 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 base Wikipedia leads to results that are superior to those returned by unsupervised techniques.

III. Web page classification

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

[2] See http://hpi-web.de/naumann/projekte/repeatability/datasets/wpsd.html

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 mapping $m_x : D_x \rightarrow E_x(S) \cup \{e_{noise}\}$ 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_{noise} if d does not describe any of the entities in $E_x(S)$.

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 notion of E' to denote the extended entity profile set ($E' = E \cup \{e_{noise}\}$). The definition and construction of these noise profiles is explained in Section III-C.

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 entities 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 [16]; hence we follow a relaxed version of this problem, 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.

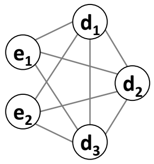

Fig. 1. Example of a graph partitioning

A. 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 contains the union of all features from all documents within he corpus $c \in C'$. 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 ($sim_{\cos}$) 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, compound nouns, etc. The evaluation of our Wikipedia-based bootstrapping approach (see Section IV-B) 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.

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 $score(d, e)$ ($sim_{\cos}(d, e)$, 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. For the explicit similarity between entity profiles and result documents we employ the well known *tf-idf* measure as w.

To measure 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

$$w'(f, e) = |\textit{tf-idf}(f, e)| + \sum_{d \in D} sim_{\cos}(e, d) \cdot |\textit{tf-idf}(f, d)|,$$

where $sim_{\cos}$ stands for the cosine similarity between the weighted feature vector of the entity profile e and that of the document d, and $|\textit{tf-idf}|$ represents the L1-normalized *tf-idf* score vector. 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 *tf-idf*-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. 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.

B. 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 ($\hat{p}_{\mathcal{B}}$) as follows:

$$\hat{p}_{\mathcal{B}}(e,d) = p(e) \cdot \prod_{f \in F_d} p(f|e),$$

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 ($\hat{p}_{\mathcal{M}}$), which takes feature occurrence frequencies into account:

$$\hat{p}_{\mathcal{M}}(e,d) = |d|! \cdot p(e) \cdot \prod_{f \in F_d} \frac{p(f|e)^{\mathit{freq}(f,d)}}{\mathit{freq}(f,d)!},$$

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

Parameter Estimation: Due to the confined nature of the feature set of a given 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 [17], [18], but different experiments showed fairly dissapointing results. Much better results were achieved by the following simple smoothing techniques

Laplace-smoothing (also referred to as additive smoothing) adds a smoothing factor α to the actual relative frequency of each feature. Thus, the prior and likelihood are estimated by:

$$\hat{p}^{\mathcal{L}}_{w}(e) = \frac{\sum\limits_{f \in F_e} w(f,e) + \alpha}{\sum\limits_{e_j \in E} \sum\limits_{f_i \in F_{e_j}} w(f_i,e_j) + \alpha} \quad \hat{p}^{\mathcal{L}}_{w}(f|e) = \frac{w(f,e) + \alpha}{\sum\limits_{f_i \in F_e} w(f_i,e) + \alpha}$$

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:

$$\hat{p}_{\lambda}(f|e) = (1-\lambda)\hat{p}_{ML}(f|F_e) + \lambda p(f|F)$$

It can be shown that by setting λ to 0.5, one can derive a *tf-idf*-style smoothing, which we used in our implementation. In our experiments we found that the Bernoulli Naïve Bayes model worked best with Laplace smoothing ($\hat{p}^{\mathcal{L}}_{\mathcal{B}}$), while the Multinomial model worked best with the Jelinek-Mercer smoothing ($\hat{p}^{\mathcal{J}}_{\mathcal{M}}$). We report the concrete results in Section IV-B.

C. Modeling the Noise Entity Profile

In the definition of our mapping m we introduced an artificial entity profile e_{noise}, to which documents should be mapped if they do not match any of the entity profiles in E. This addition accounts for the fact that the set of unique entities having the same name is limited by the underlying bootstrapping source. Hence, result documents that do not correspond to any of the entity profiles from E are assigned to the artificial profile e_{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 and, for the sake of brevity, present the two best performing and most robust ones.

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

$$F_{e_{\bigcup \mathit{noise}}} = \{f | f \in F_e, e \in E\}$$

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

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

$$F_{e_{\bigcap \mathit{noise}}} = \{f | \forall e \in E, \forall c \in C, e \neq c : f \in F_e \cap F_c\},$$

where $C = C' \setminus \{e_{\mathit{noise}}\}$. 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.

IV. 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 [19], WWW'05 [10], and WePS-2 [8]. 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 do not allow such usage. Therefore, we extracted over 900 Wikipedia articles about persons of 50 different groups of ambiguous names; following, we extracted the top-100 Web Google search results (excluding Wikipedia pages).

To create the gold standard for the dataset, for each ambiguous name 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 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 that had to be checked manually. We refer to [20] for further details about the data set. Furthermore, we provide online access to the evaluation dataset[3].

[3] http://hpi-web.de/naumann/projekte/repeatability/datasets/wpsd.html

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 the very many 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 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 ($+e_{noise}$), while the ground truth tells us that only 33 out of 96 results are assigned to 6 out of 100 Wikipedia articles and the rest to e_{noise}. Hence, using Wikipedia to bootstrap the grouping of search results to ambiguous person names is already challenging and also covers the problem of clustering documents about less famous people on the Web.

A. *Comparison with clustering techniques*

This section provides a comparison of baseline 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 toolkit[4]. For each of the methods, we tested many different configurations. However, due to the limited space, we refer to [20] for further details of the baseline configuration.

For comparison with the above methods we used an implementation of $score'$ and $\hat{p}_{\mathcal{M}}^{\mathcal{J}}$. Both methods were applied using the intersection-noise entity ($e_{\bigcap noise}$) to derive a mapping as proposed in Section III. 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_{noise}). This led to 1,095 Web documents used for the clustering evaluation.

TABLE I. CLUSTERING EVALUATION OF DOCUMENTS RELATED TO A WIKIPEDIA ARTICLE.

	purity	**NMI**
$score'$	**0.913**	**0.560**
$\hat{p}_{\mathcal{M}}^{\mathcal{J}}$	0.890	0.492
HAC	0.829	0.321
K-Means	0.814	0.287

To provide the clustering performance comparison, Table I shows the average purity and NMI values over all 50 names from the dataset. As can be seen, both 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.

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 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, both 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 following section discusses the evaluation of our approaches for the task of linking Web documents to ambiguous Wikipedia entities and thus covers the complete problem.

B. *Probabilistic vs. vector space models*

A commonly used measure to evaluate the performance of binary categorization methods is the F_1 measure. For a multiclass classification problem, typically applied measures are the micro- and macro-averaged F_1 ($micro(F_1)$ and $macro(F_1)$, respectively). Due to space restrictions, we try to combine both values by averaging over all 50 name tasks from the previously introduced evaluation dataset ($\mathcal{X}$). More specifically:

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

A more detailed analysis including $micro(F_1)$ and $macro(F_1)$ values for all approaches can be found in [20].

Table II 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 III-A ($sim_{\cos}$, $score$, $score'$) and III-B ($\hat{p}_{\mathcal{B}}^{\mathcal{L}}$, $\hat{p}_{\mathcal{M}}^{\mathcal{J}}$). Each column shows the influence of different configurations for the noise entity: a union-noise entity ($e_{\bigcup noise}$), an intersection-noise entity ($e_{\bigcap noise}$) (see Section III-C), and a "no noise entity" configuration. The "no noise entity" column stands for a configuration without any noise entity (i.e., $E' = E$). 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. 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).

[4] http://www.cs.waikato.ac.nz/ml/weka/

TABLE II. Performance comparison between different mapping functions and noise entity profiles ($\overline{F_1}$ scores)

	no noise	**union-noise**	**intersection-noise**
$sim_{\cos}$	0.233	0.438	0.601
$score$	0.210	0.704	0.701
$score'$	0.322	0.569	**0.734**
$\hat{p}_{\mathcal{B}}^{\mathcal{L}}$	0.251	0.634	0.642
$\hat{p}_{\mathcal{M}}^{\mathcal{J}}$	0.257	0.486	**0.730**

The configurations with the intersection-noise entity and $score'$, respectively $\hat{p}_{\mathcal{M}}^{\mathcal{J}}$ performed best in our evaluation. However, applying the union-noise entity profile, the mapping function $score$ (i.e., the model that quantifies the dot-product-based similarity between a result document and an entity profile) outperforms these scoring functions. We hypothesize that this is due to the fact that the larger union-noise entity introduces too much noise for the smoothed scoring function ($score'$, 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 ($\hat{p}_{\mathcal{M}}^{\mathcal{J}}$, 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.

V. Conclusion

The focus of this work has been on the design of approaches to the problem of clustering search results to ambiguous person-name queries. The proposed methods build on the idea that entity profiles, such as Wikipedia pages about persons, can be bootstrapped to cast the above problem into a classification problem, where results are mapped to the most similar profile.

The suite of presented and evaluated methods covers vector-space and probabilistic models. Dealing with noisy and biased data from the Web documents as well as Wikipedia was essential for the introduced approach. In particular, the incompleteness of the entity source (open world assumption) was in the focus of this work.

The provided experiments were based on a hand-labeled dataset over more than $85k$ 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.

As part of our future work, we are aiming to aggregate profiles from multiple other Web 2.0 sources besides Wikipedia to improve the grouping of search results to ambiguous queries. Furthermore, we are planing to test the proposed models for other use cases besides the person name disambiguation problem (e.g., clustering of places, organizations, or products). To this end, more complex features (e.g., multigrams, structured attribute-value pairs, etc.) could boost our methods further. The final goal is an efficient, incremental disambiguation of search results.

References

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

[2] O. Zamir and O. Etzioni, "Web document clustering: a feasibility demonstration," in *Proceedings of the International ACM SIGIR Conference on Research and development in Information Retrieval*, 1998.

[3] H. Köpcke, A. Thor, and E. Rahm, "Evaluation of entity resolution approaches on real-world match problems," *Proceedings of the VLDB Endowment*, 2010.

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

[5] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[6] 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)*, 2006.

[7] 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 (ACL)*, 1998.

[8] J. Artiles, J. Gonzalo, and S. Sekine, "WePS-2 evaluation campaign: overview of the web people search clustering task," in *Web People Search Evaluation Workshop (WePS)*, 2009.

[9] R. Berendsen, B. Kovachev, E.-P. Nastou, M. de Rijke, and W. Weerkamp, "Result disambiguation in web people search," in *Proceedings of the European Conference on IR Research (ECIR)*, 2012.

[10] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *Proceedings of the International Conference on World Wide Web (WWW)*, 2005.

[11] 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 IEEE International Conference on Data Engineering (ICDE)*, 2007.

[12] 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)*, 2005.

[13] K. Balog, L. Azzopardi, and M. de Rijke, "Resolving person names in web people search," in *Weaving Services and People on the World Wide Web*, 2008.

[14] N. Slonim and N. Tishby, "Document clustering using word clusters via the information bottleneck method," in *Proceedings of the International ACM SIGIR Conference on Research and development in Information Retrieval*, 2000.

[15] 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)*, 2002.

[16] 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)*, Paris, France, 2012.

[17] M. Efron, P. Organisciak, and K. Fenlon, "Improving retrieval of short texts through document expansion," in *Proceedings of the International ACM SIGIR Conference on Research and development in Information Retrieval*, 2012.

[18] X. Liu and W. B. Croft, "Cluster-based retrieval using language models," in *Proceedings of the International ACM SIGIR Conference on Research and development in Information Retrieval*, 2004.

[19] J. Artiles, J. Gonzalo, and F. Verdejo, "A testbed for people searching strategies in the WWW," in *Proceedings of the International ACM SIGIR Conference on Research and development in Information Retrieval*, 2005.

[20] T. Gruetze, G. Kasneci, Z. Zuo, and F. Naumann, "Bootstrapped grouping of results to ambiguous person name queries," *CoRR*, 2013, arXiv:1312.1897 [cs.IR].