Information Sciences 261 (2014) 89–100

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Toward detection of aliases without string similarity

Ning An [a,b,*], Lili Jiang [c], Jianyong Wang [d], Ping Luo [e], Min Wang [e], Bing Nan Li [a,f,*]

[a] Gerontechnology Lab, Hefei University of Technology, Hefei, China
[b] School of Computer and Information, Hefei University of Technology, Hefei, China
[c] Max Planck Institute for Informatics, Saarbrucken, Germany
[d] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[e] HP Labs China, Beijing, China
[f] Department of Biomedical Engineering, Hefei University of Technology, Hefei, China

ARTICLE INFO

Article history:
Received 20 June 2012
Received in revised form 24 September 2013
Accepted 8 November 2013
Available online 18 November 2013

Keywords:
Alias detection
Entity subset
Active learning
Supervised classification

ABSTRACT

Entity aliases commonly exist. Accurately detecting these aliases plays a vital role in various applications. In particular, it is critical to detect the aliases that are intentionally hidden from the real identities, such as those of terrorists and frauds. Most existing work does not pay close attention to the aliases that have low/no string similarity to the given entities. In this paper, we propose a classifier that is based on active learning for detecting this type of aliasing. To minimize the cost of pair-wise comparison, a subset-based method is designed to restrict the selection within entity subsets. An active learning classifier is then employed in each entity subset to find the probability of whether a candidate is the alias of a given entity within the subset. After all of the results from the classifier are integrated, a list of aliases is returned for each given entity. For evaluation, we implemented four state-of-the-art methods and compared them with our proposed approach on three datasets. The results clearly demonstrate that this new active learning classifier is superior to those existing methods.

1. Introduction

Solving the problem of entity alias detection [20,22] is important for a large number of applications. Typically, there are two types of aliases: one type of alias can be roughly detected through string similarity, for example, "John Smith" and "J.M. Smith"; the second type of alias, such as nicknames, fraud, and terrorist aliases, has a low string similarity, and certain number of these aliases are intentionally hidden from their real identities. It is obvious that a pure string similarity search could fail to handle semantically identical entities. For example, "Wisconsin state" has the nickname "dairy state", and "Abu Abdallah" was used as an alias of "Osama bin Laden". It is appropriate to call this type of alias a semantic entity alias. Detecting semantic entity aliases is very useful in the real world, and the aim of this paper is to detect such semantic entity aliases.

Researchers have investigated this issue with respect to various domains, including people alias extraction [5,14], fraud detection [6,23], medical alias extraction [10,18], and terrorist recognition [16,22]. Their solutions focus on a special domain (e.g., peoples' names, terrorism or fraud detection) but fail to correctly obtain the true aliases under a broad range of circumstances. Compared to detecting an entity alias with string similarity, it is more challenging to discover a semantic entity alias. First, there is no or quite low string similarity between a given entity and its semantic aliases. In particular, certain types of

* Corresponding authors. Address: Hefei University of Technology, P.O. Box 112, Tunxi Road 193, Hefei 230009, China. Tel.: +86 139 65089390; fax: +86 551 62901760.
E-mail addresses: ning.g.an@acm.org (N. An), bingoon@ieee.org (B.N. Li).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.11.010

entities (e.g., terrorists or frauds) are intentionally hidden from their real identities; hence, the commonly used rules (e.g., "aka", "as well known as", and "also called") [5] do not work. Second, with the increasingly growing volume of information and data, the number of an entity's aliases is rather small compared to billions of strings/entities. Consequently, it is difficult to accurately detect the true aliases for a concerned entity.

To achieve this goal, we decided to employ a supervised method that is based on active learning to solve this problem. Then, the following two issues arise: first, because there is low/no string similarity, it is expensive to choose potential alias candidates for each given entity from a large-scale document corpus; second, classification features and training samples play important roles in the supervised learning methods [19]. To address these two issues, we propose a probabilistic classification method that is based on active learning. Initially, to reduce the cost of pair-wise comparisons for selected alias candidates, a subset-based method is introduced to divide the extracted entities into subsets. Next, three informative features are employed to train a probabilistic classifier. In particular, the strategy of active learning is used to choose high-quality training samples. Finally, the classifier assigns a probability to each pair of a concerned entity and its corresponding extracted entity (alias candidate) in the same subset.

The contributions of this study include:

- Proposing a subset-based method to decrease the cost of pair-wise comparisons and to improve the overall precision of alias detection.
- Designing a classifier that is based on active learning, for which the training samples are selected for high-quality classification.
- Conducting extensive experiments on three types of datasets.

It is noteworthy that a preliminary version of this paper was published as a poster [17]. We have since then made significant enhancements to this paper, as follows:

- Explaining the proposed method in more detail, with additional background knowledge.
- Enhancing the proposed method by adding graph-based features and user-selected samples during the training.
- Adding state-of-the-art indexing measures to make a better evaluation of the proposed method.
- Presenting more experiments and related analyses.

The remainder of this paper is organized as follows. After Section 2 introduces the related work, Section 3 formulates the problem and describes the overall framework. We then present the proposed methods in Section 4 and show the experiment results in Section 5. Conclusions and future work are summarized in Section 6.

2. Related work

The problem of entity alias detection has a close connection with the data matching problem [8], including deduplication [12,24], record linkage [3] and entity resolution [4]. Here, deduplication is the process of removing duplicate records, i.e., records that refer to the same entity, in one data set; record linkage is the process of finding related records and creating a linkage between them [7]; and entity resolution is the process of finding duplicate records and merging them. There has also been some work that addresses the data-matching issue by using active learning [1,24,28]. Most of these studies are aimed at resolving the entities and have string matching as the initial stage. There are two major strategies, namely grouping and indexing, to reduce the cost of performing pair-wise comparisons [9,13,15]. The former strategy is often adopted in database-based applications, where each entity has various attributes, and the entities are grouped according to the similarity between the attribute values. The latter strategy depends on string similarity, which does not work for semantic alias detection. In this paper, we propose a new subset-based method for effectively reducing the scope of the pair-wise comparisons for each concerned entity, thus making the subsequent processing more efficient.

While many existing studies have focused on detecting the general aliases, such as a person's name, an organizational name, and a place name [5,10,11,14], several other studies have paid close attention to detecting deceptive and unknown identities, including falsified identifiers created by frauds [6,23] and terrorists [16,22]. The aim of [5] was to extract person aliases in the Web; this approach constituted three steps, namely pattern extraction, model training, and candidate extraction. The authors employed a few string patterns, such as "aka" and "better known as", and evaluated the method on person's names and location names. Because they focused on the prominent entity aliases, the common rules (e.g., "aka", "well known as", and "also called") were effective. The authors of [23] employed point-wise mutual information (PMI) to measure the importance of various features and, then, used the cosine measure to compute the similarity between each given entity and its alias candidates. Additionally, researchers [11] attempted to detect entity aliases by learning a set of rules and using the tree augmented naive Bayesian networks (TAN) to calculate the probability for a pair of entity and alias candidates. The study in [14] constructed a social network S from collections of email addresses and identified all of the aliases for a given email address that also appears in S. One of the disadvantages of this method is its specificity on an email alias instead of a general setting. The authors of [16] used a training model to predict terrorist and spammer aliases. The authors of [22] proposed a two-stage latent semantic analysis (LSA) for terrorist detection. It is noteworthy that this study used only the test

case "al Qaeda" for evaluation. Furthermore, the authors of [10] attempted to extract gene aliases from the PubMed corpus. They extracted all of the gene symbols from the text abstracts and collected statistical information (frequency) on them. Finally, the Jaccard distance was used to calculate the similarity between the official gene symbol and each of its aliases.

3. Overview of the proposed framework

In this section, we formulate the problem and describe the proposed solution for semantic alias detection in detail.

3.1. Problem of semantic alias detection

Given a document corpus $D = \{d_1, d_2, \ldots, d_n\}$ and the entity e, we define semantic alias detection as the following task: recognizing entities that denote the same real-world object as e and that could have no (or very low) string similarity to e.

3.2. Overall framework for semantic alias detection

Fig. 1 depicts the overall framework of our proposed solution. Given a document corpus, we first employ a tool for natural language processing, namely, LingPipe [26], to extract all of the entities that are alias candidates and to remove some noisy alias candidates through a stop-word list [27]. Next, we design a subset-based comparison method to initially group all of the entities (i.e., concerned entities and alias candidates) into subsets according to their occurrences in the document corpus. At the same time, a logistic regression classifier is trained based on the given training data. Finally, for each pair of a concerned entity and its alias candidate, the classifier outputs a probability value that represents how likely they are aliases with each other and the probability of the opposite situation, which can be easily obtained by subtracting this value from 1. When the probability is larger than the opposite, it is appropriate to conclude that they denote the same entity, and vice versa. The following section will describe this proposed approach in detail.

4. Methods

4.1. Candidate extraction

Given a document corpus, Lingpipe is used to extract different types of named entities (e.g., organization, person and location). For alias candidate extraction, we mainly address two situations: (1) if the type of the concerned entities is given, then we choose the corresponding type of entities as candidates for the concerned entities; (2) if the entity type is unknown, then all of the extracted entities are considered to be alias candidates despite the entity type. For example, if we aim to detect the aliases for a person, then extracted person names are used as candidates. Although Lingpipe achieves high performance in extracting person names, it often confuses organizations and locations. Therefore, we merge the extracted organizations and locations as a united list of candidates.

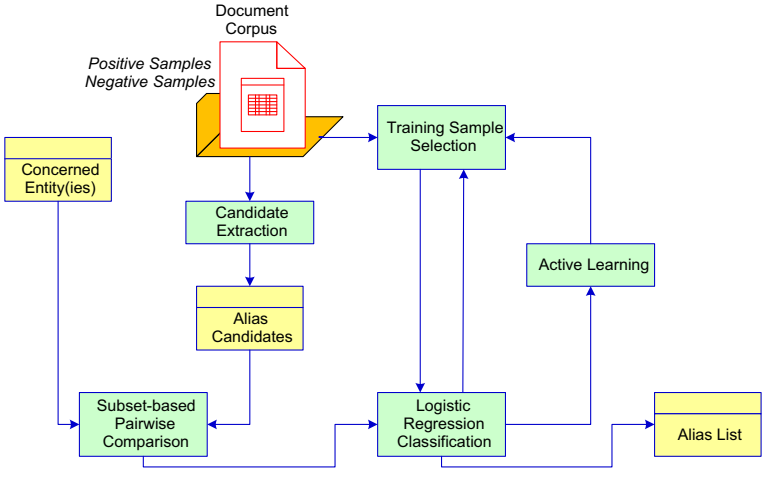

Fig. 1. The overall framework for semantic alias detection.

4.2. The subset-based pair comparison

4.2.1. Motivation

When choosing aliases for a given entity, we must compare the entity with each extracted candidate alias, i.e., the number of comparison will be n assuming that we have n extracted entities. According to the analyses of indexing techniques in the survey [9], the cost of comparing entities increases quadratically as the given entity set and the extracted entity set become larger. In particular, because the true aliases fill only a small part among the thousands of extracted entities, the vast majority of comparisons are redundant and unnecessary. Thus, one of the essential components of our framework is to reduce the time for the comparisons.

Aliases do not always co-occur with the concerned entity. To illustrate different scenarios, we use an uppercase letter to denote an entity, a dash "-" between two entities to denote their co-occurrence in the same document, and an ellipsis "..." between two entities to represent a linkage between them. Here are four scenarios that we are interested in: (1) A - A': entity A and its alias A' co-occur in the same document; (2) A-B-A': entity A and entity B, entity B and entity A's alias A' co-occur in different documents, respectively; (3) A-A''-A': entity A and its alias A' co-occur with A''s alias A'', respectively, in different documents; (4) A-B...B'-A': entity A and its alias A' co-occur with another entity B and its alias B', respectively, in different documents. B and its alias B' could be linked according to scenario (1), (2) or (3). Motivated by the observation of the entity distribution in the document corpus, our subset-based method is to group the entity A and its aliases A' (described in the above scenarios) into the same subset.

Assume that a set $\{a_1, a_2, a_3, \ldots\}$ represents different aliases of the same entity. We observe that: (1) among these entity aliases, some frequently occur in multiple documents, while others occur occasionally. For example, "Wisconsin State", "Badger State", "Dairy State", and "American Dairyland" represent the same entity. The former three are commonly used in various documents, while the "American Dairyland" is rather rare; (2) we sort these aliases in descending order according to the number of documents in which they appear. We observe that the union of documents in which the most popular entities appear almost covers all of the documents in the document corpus. Take the aliases mentioned above as an example, $a_1 \rightarrow \{d_1, d_2, d_3, d_4, d_5\}$, $a_2 \rightarrow \{d_3, d_4, d_6, d_7, d_8\}$, $a_3 \rightarrow \{d_2, d_3, d_5, d_6\}$, and $a_4 \rightarrow \{d_7, d_8\}$, where a_1, a_2, a_3 and a_4 denote the different aliases of the same entity, and the set following the arrow represents the documents where an alias appears. Let D_1 represent the union of documents that contain a_1 and a_2, which represent the most frequent aliases as shown above, and D_2 represents the union of documents where other aliases (i.e., a_3 and a_4) occur; D_1 is either equal to D_2 or a superset of D_2. These observations show that some non-popular aliases cannot be easily detected by simply using the occurrence frequency or the co-occurrence information, but it always co-occurs with some popular aliases in a few documents. With this observation, we consider only the most frequent entities first and divide their occurrence documents into subsets. Subsequently, the entities that occur in the documents within the same subset will be accounted for as entity candidates with respect to each other. This strategy aims at reducing the comparison time with as little performance sacrifice as possible.

4.2.2. Method description

Our subset-based method has three goals: (1) maximizing the number of different aliases for the same entity in the same subset; (2) minimizing the overlapping and size differences among entity subsets; and (3) maximizing the recall. This method facilitates the entity-candidate comparison in a small-scale entity subset and excludes the majority of negative alias candidates from each concerned entity.

The overall process is as follows (Fig. 2): (1) Frequent entities selection: we rank all of the entities in descending order based on their frequencies in the document corpus. Afterward, we extract the most frequent N percentage entities. (2) Document subset construction: for any entity in the selected entities, we obtain their occurring document set, and then, we perform the operations of deletion and merging. For example, s_i and s_j are occurrence document sets for two entities. Then, s_i will be removed if s_j is the superset of s_i, or vice versa. If the intersection ratio between s_i and s_j is larger than a pre-defined threshold λ, then they will be merged. Afterward, we obtain a list of document subsets. (3) Entity subset construction: The entities whose documents are grouped into subsets are then assigned to the same entity subset. According to the above description, there are two parameters, λ and N, among which λ denotes the intersection ratio that is used to decide when two document subsets should be merged, while N denotes the percentage of the most frequently occurring entities in the whole document corpus.

4.2.3. Parameter settings

Among the critical controlling parameters, N is learned through experimental validation on the training dataset (see Section 5). Furthermore, our experiments demonstrate that the frequent percentage across different data corpuses differs, i.e., a fixed threshold value is not suitable for different data corpuses. Therefore, we employ a flexible threshold $\lambda = minimum(\#\text{document})/\ average(\#\text{document})$, i.e., the minimum number of occurrences in the document divided by the average number of occurrences in the document of all of the entities.

4.3. Active learning for semantic alias detection

After applying the subset-based method, we obtain a list of subsets that contain the concerned entities and the extracted alias candidates. The following task will be required to perform a pair-wise comparison on the entities in the same subset.

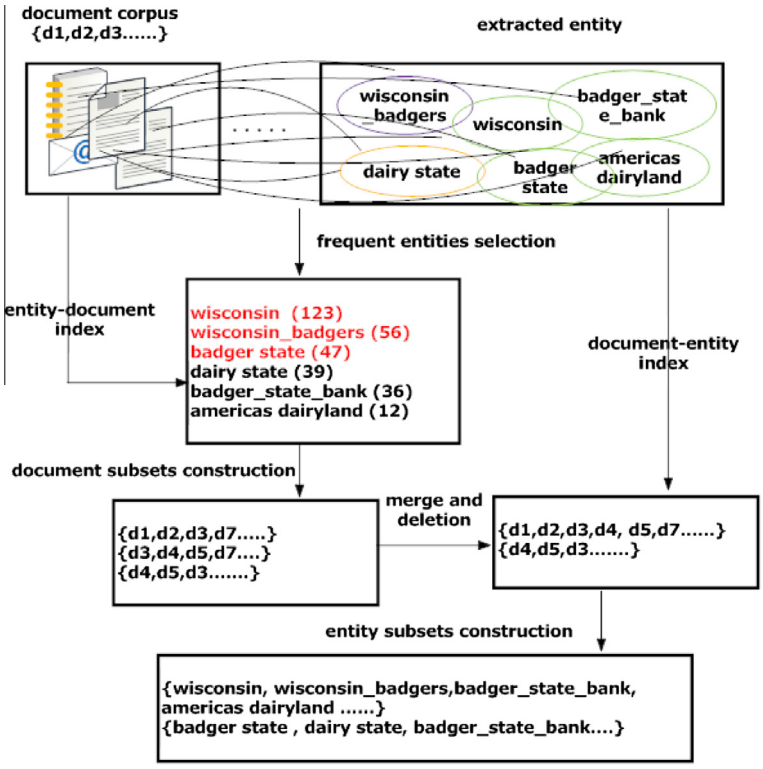

Fig. 2. The workflow of a subset-based method.

We use the classification method of logistic regression in combination with active learning to predict the probability of whether a given entity and any alias candidate in the same subset denote the same object. Finally, all of the results are aggregated for each concerned entity. For example, for entity a and its alias candidate b, the classifier outputs $P(\text{Yes}|(a,b)) = 19\%$ and $P(\text{No}|(a,b)) = 81\%$, where $P(\text{Yes}|(a,b))$ denotes the probability that the entities a and b represent the same real world entity, and $P(\text{No}|(a,b))$ denotes the probability that they do not. The next two subsections will introduce the classification features and the classifier training.

4.3.1. Feature analysis

We choose the training features based on the following motivations:

- The co-occurrence statistics in the document corpus are always useful for entity alias detection. These statistics can capture most of the aliases of the given entity that occur frequently, although this process always results in some additional noisy candidates.
- It is observed that the aliases for the same entity might have the same/similar social network. For example, a person who uses different nicknames often contacts the same friends, affiliates with the same organizations, and visits the same places. This social connection information could capture a few aliases that might not co-occur with the given entity.
- There exist some aliases that seldom co-occur with the given entity or share the same entity mentioned above. However, they sometimes co-occur with the different aliases of the same entity. Consequently, the co-occurrence information and social connection will not work for this situation. To address this issue, we use the concept of alias relevance. The alias relevance captures an alias that co-occurs with its aliases instead of the given entity or an alias that co-occurs with the aliases of other entities. We build a graph to capture the relevance.

Altogether, we use the following three features: co-occurrence relevance, social relevance and alias relevance, to construct a classifier with logistic regression.

Based on the above analyses, we introduce the selected features as follows:

- Co-occurrence relevance

Point-wise mutual information (*PMI*) is employed to measure the association between two entities. The value of $PMI(e,e')$ is zero if e and e' are independent, positive if they are correlated, and negative if they are not related.

$$PMI(e,e') = \log \frac{p(e,e')}{p(e)p(e')} \tag{1}$$

Here, $p(e)$ denotes the occurrence probability of e; in other words, $p(e) = |D_i|/|D|$, where $|D_i|$ is the number of documents in which the entity e appears, and $|D|$ is the number of documents in the corpus. Here, $p(e,e')$ denotes the co-occurrence probability of the entities e and e', which is calculated by dividing the number of documents in which they co-occur by the size of the document union that they appear in.

- Social relevance

Because the Internet can be described as a network, the relevant entities in the real world can be considered to be connected in a social network. For example, an organization and its aliases could co-occur with the persons who are affiliated with this organization, and one person and his/her aliases could be linked with his/her common friends or places. Therefore, based on the idea of a social connection, we calculate the social relevance between e and e'. The larger the number of shared co-occurrences of the entities is, the higher the social relevance of e and e'.

$$SR(e,e') = \frac{F(e) \cap F(e')}{F(e) \cup F(e')} \tag{2}$$

where $F(e_i)$ denotes the number of entities that co-occur with e_i in the same documents.

- Alias relevance

To capture the alias relevance, we construct an entity graph and search the path in the graph between the given entity e and the alias candidate e', using a length of 3 or 4. The intermediate nodes on these paths are required to be in the same subset with e or e'. Finally, we sum these paths and obtain a relevance value, as follows:

$$AR(e,e') = \left(\sum_{p=1}^{s} \sum_{k=0}^{t} (SR(e_k, e_{k+1}) + SR(e'_k, e'_{k+1}))/t^k \right) / s \tag{3}$$

where s is the total number of paths between e and e', and t is the length of path $p(t = 3,4)$; e_k and e_{k+1} are the connected nodes of the $(k+1)$th edge on the current path p. Here, a higher social relevance leads to a stronger alias association between the two entities.

4.3.2. The probabilistic classifier

A logistic regression probability model is first estimated from training data that is composed of a list of vectors and their corresponding categories. We manually label some positive and negative entity pairs as training data. For each pair of entities in the training data, we first obtain their feature values according to Eqs. (1)–(3), and we input these feature values into a dense vector. Here, the dimension size of the vector equals the number of features (i.e., 3). The categories are discrete, and we have only two categories (i.e., Yes and No) in this study, where "Yes" denotes that the input entities are aliased with each other, and "No" denotes that they are not. Second, we take these double-valued feature vectors with their integer-valued category as the input to estimate the logistic regression model.

In the evaluation stage, for each pair of an entity and its alias candidate, the inputs are coded as dense vector instances, and the outputs are a category number with a probability. The conditional probability about how likely these two entities alias with each other is defined as follows:

$$P(Yes|e_1,e_2) = 1 \Big/ \left(1 + \sum_{i<k-1} \exp(\beta[i] * f_{e_1 e_2}) \right) \tag{4}$$

where e_1 and e_2 denote two entities, f denotes the input feature vector of e_1 and e_2, k is the dimensionality of feature vector ($k = 3$), and β denotes the weight vector from the estimated logistic regression model. In addition, $\beta[i]$ $(i < k-1)$ is the ith coefficient in the weight vector β. The weight vector could be obtained through model training.

Some traditional classifiers based on passive learning require much manual annotation and randomly select training samples [18,19]. However, the manual annotation to produce the training data is expensive, and the randomly chosen training samples might not be sufficient to cover all of the possible cases. In contrast to the traditional logistic regression training, we propose a logistic regression model that is based on active learning, which contributes in two respects: (1) this approach pursues a reasonable number of high-quality training samples; and (2) users are involved in the classifier training [2,26,29]. Algorithm 1 presents the process of the classifier training, which is based on active learning.

Algorithm 1. Classifier Training

Input:	Training set $T = \Phi$, initial training sample set I, labeled training sample set L, and unlabeled training sample set U, $m = 5$, *probability* = 0.
Output:	Classifier C
0:	Set $T = I$
	Loop until "$L = \Phi$" or "all samples in U with (*probability* -0.5) > 0"
1:	training C using T
2:	Apply C on all samples of L and U, output *probability* (Eq. (5)) for each sample;
3:	Select m samples from L with the highest *uncertainty*, add to T;
4:	User select m samples from U with the lowest value of (*probability* -0.5), and add to T.
5:	Delete the selected samples in 3 and 4, respectively, from L and U.

We first train an initial classifier C, which is based on the labeled samples from I. Afterward, we divide the training data into labeled samples L and unlabeled samples U. To select the most informative training data for the high-quality classifier, we then repeat the procedure (1–5) described in Algorithm 1: apply the current classifier on each sample in L and U. Next, we choose the m samples that have the highest uncertainty [20] from the L and m samples with the lowest precision from U(m is set to 10 in this study). In this process of active learning, the most informative samples are chosen instead of random selection in passive learning methods. Finally, a probability value is assigned to determine whether the concerned entity and each of its alias candidates denote the same real-world entity.

As described above, we employ the uncertainty sampling method [20,24] in the learning process. The output probability of the classifier is a double value that ranges from 0 to 1, and 0.5 presents that the classifier is most uncertain about the class label. Thus, m samples with an estimated value of P that is close to 0.5 are selected in each iteration. Then, we employ the sampling method in [20], which chooses $m/2$ samples that are below and above 0.5. This approach guarantees that training on opposite sides of a decision boundary is useful. Moreover, the users are allowed to choose an additional m samples before and after 0.5 that have an incorrect classification. In other words, we choose m samples with the minimum value of $|P - 0.5|$ and m samples with the smallest precision. These two sets of samples are used as the input of the next iteration. The average number of iterations in our work is 3. Additionally, the initial training samples are essential, and usually their size is at least one half of the training dataset.

5. Experimental study

5.1. Datasets

Three datasets were used to evaluate the proposed approach, and the performance was experimentally compared with four baseline methods.

The first dataset (Dataset 1)[1] is composed of two sub-datasets; the first contains 50 English place names, and the second contains 50 English person names [5]. The place names are 50 US states, but the person names are extracted from various fields. All of the aliases for these place/person names are given as ground truth. To adapt for the evaluation, we extended this dataset by collecting Web pages through issuing those names/aliases as queries to Google. In particular, for each pair of entities (i.e., a state name or a person name) and its alias, we collected the top 100 Web pages for the given entity, 100 returned Web pages for the alias, and 100 Web pages that contain both the entity and the alias. We removed the duplicated pages and used the remaining pages as a document corpus.

The second dataset (Dataset2)[2] [16] also contains two sub-datasets, which are terrorism data extracted from public Web pages and spam emails from websites [30]. They both contain three types of files (i.e., the entity-alias pairs, all of the extracted words, and the document-word index). The terrorist subset provides 20 given terrorist names, 5578 document-word indices, and 4054 extracted entities. The spam subset provides 10 spammer names, 5563 document-word indices, and 5337 extracted entities.

The third dataset (Dataset3) is a manually constructed dataset. We collected the names of all of the presidents in US history and their nicknames.[3] In a similar way as with Dataset1, we collected Web pages through issuing those president names and nicknames as queries to the search engines. For each president's name, we collected the top 100 web pages for the real name and the 100 top web pages for each of the nicknames. Next, we merged these webpages for each president's name. Afterward, we performed reduplication and took the remaining web pages as a document corpus.

We then divided each dataset into a training dataset (4/5) and a test dataset (1/5). The training datasets were used for sample selection and classifier learning. The test datasets were used for evaluation on the task of extracting semantic entity

[1] http://www.iba.t.u-tokyo.ac.jp/~danushka/data/aliasdata.zip.
[2] http://www.cs.cmu.edu/~awm/mnop_data/.
[3] http://en.wikipedia.org/wiki/President_nicknames.

aliases. In our study, we conducted the experiments on the extraction of the state alias, person nickname, terrorist alias, and spam alias.

5.2. Baseline methods

We implemented four algorithms for semantic alias extraction as the baselines for comparison, namely, a PMI-based approach [23], a graph-based approach [14], the approach based on logistic regression [11,16], and a two-step LSA approach [25]. For convenience, these baselines are called *B_PMI*, *B_Graph*, *B_LR* and *T_LSA*, respectively.

- The baseline *B_PMI* constructs a frequency vector for each concerned entity *e*, and then, it builds a mutual information vector for *e*. The cosine value between the two mutual information vectors is computed for any two entities, where a higher value means a high probability.
- The baseline *B_Graph* models the network of email addresses as an undirected graph and combines the shortest path with the email account of each source to determine the node ranking for each given node (i.e., an email address).
- The baseline *B_LR* combines the orthographic mode and a semantic model of the features to train a supervised model to detect terrorism and spam aliases.
- The baseline *T_LSA* employs LSA twice on the given document corpus *D* and outputs a list of alias candidates for each concerned entity. Note that the authors adjusted three parameters, including the matrix dimension, the percentage of replacement and the probability of *TF-IDF* filtering. For fairness in the comparison, we chose the parameters that lead to the best results of their method.

5.3. Evaluation measures

For the subset-based method evaluation, we used the popular reduction ratio (*rr*) and pair completeness (*pc*) [9], respectively, for space reduction and quality evaluation.

$$rr = 1 - \frac{N_b}{|A| * |B|} \tag{5}$$

where N_b is the number of entity-candidate pairs that are produced by the subset-based algorithm, and |A| and |B| denote the number of the given entities and alias candidates, respectively. The reduction ratio measures the relative reduction in the comparison space.

Pair completeness is used to measure the quality as

$$pc = \frac{N_m}{M} \tag{6}$$

where N_m is the number of correctly classified entity-candidate pairs in the comparison space, and *M* denotes the total number of true matches.

For the classification quality, we used two measures for evaluation: (1) *F*1 as the harmonic mean of the precision and the recall, where *precision* is the ratio of the correctly recognized aliases to the recognized aliases, and *recall* is the ratio of the correctly recognized aliases to the overall true aliases.

$$F1 = \frac{precision \times recall}{precision + recall} \tag{7}$$

(2) *AUC* presents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. The AUC tests whether positives are ranked higher than negatives. The *AUC* is equivalent to the Mann–Whitney–Wilcoxon sum of ranks statistic [21] and is estimated as follows:

$$AUC = \frac{s - (pos \times (pos + 1)/2)}{pos \times neg} \tag{8}$$

Here, *s* is the sum of ranks of true positive aliases, *pos* denotes the number of positive samples, and *neg* denotes the number of negative samples.

5.4. Empirical results

5.4.1. Evaluation of the subset-based method

5.4.1.1. Parameter setting. To choose a suitable value for *N* in the subset-based method, we randomly chose 50 entities from the training dataset for parameter learning. As shown in Fig. 3, the *x*-axis denotes the parameter *N*, and the *y*-axis represents the average performance of alias discovery in terms of *F*1 and *AUC*. Fig. 3 shows that the final performance changes significantly in relation to the value of *N*. According to the experiments on the training dataset, the top 50% entities that appear can totally cover all of the documents of the given document corpus with an acceptable performance. In other words, half of the

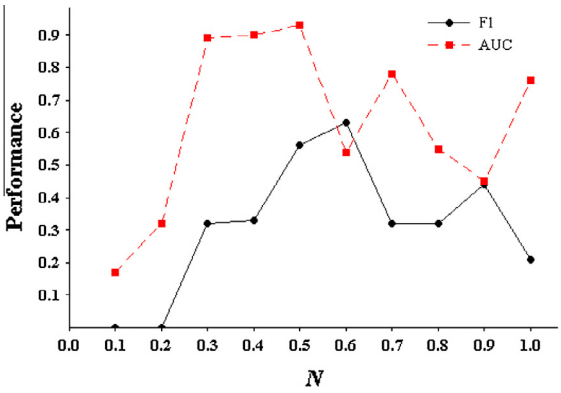

Fig. 3. Parameter setting of the subset-based method.

Table 1
Statistics of the subset-based method.

Datasets	#Subsets	#Min	#Max	#Average
Dataset1	426	3	213	19
Dataset2	34	2	260	29
Dataset3	356	4	278	24

most frequent entities appear in nearly all of the documents. Consequently, we chose N as 50% for the entire test cases, to balance the performance of $F1$ and AUC.

5.4.1.2. Generated subset statistics. Table 1 presents the statistics information on the subset-based method. For each dataset, the number of subsets and their sizes (i.e., min, average, and max) are listed.

It is very likely that a few extracted entities are primarily selected as the alias candidates for different concerned entities. In other words, some entities are possibly grouped into more than one subset. To avoid too much overlapping, we performed an experimental verification by computing the intersection ratio between the generated entity subsets. The results show that for more than 94.9% subsets, the intersection ratios are below 0.1; in fact, the intersection ratio of almost 90% of the subset pairs is zero. Only a few of the generated entity subsets have an intersection ratio that is above 0.5. Therefore, the subset-based pair-wise comparison can ensure a reasonable recall in the initial stage.

5.4.1.3. Performance demonstration. Here, we demonstrate the effect of using the subset-based method in terms of the efficiency and effectiveness. In Table 2, we presented the optimization counts in terms of a pair-wise comparison by using the subset-based method.

The experimental results show that the subset-based method can reduce the times required for the pair-wise comparisons between the given entities and their alias candidates. Considering that the subset-based approach cannot assure that all of the aliases of each given entity are grouped into the same subset, we could be concerned about the subset-based approach leading to a reduction in the recall. Experimental results show that the subset-based method improves the precision of alias detection by filtering out some noisy candidates from each subset.

To evaluate both the precision and the recall, most studies provided the concerned entities in advance. Then, it remains only to compare the candidates with each of the concerned entities, linearly. However, in real-world applications, it is very likely that no entity is given in advance and that all of the entity-alias pairs must be discovered from the given document corpus. In this case, the proposed subset-based method can dramatically reduce the cost of the pair-wise comparison and improve the precision.

5.4.2. Evaluation of the active learning method

The purpose of the active learning method is to select training samples to be as informative and few as possible. In other words, it is expected that we use the fewest and most valuable training data to obtain the highest classification quality. As shown in Table 3, we compared the passive learning method of training sample selection, in terms of both $F1$ and AUC, with the method based on uncertainty sampling of the active learning method [20]. In active learning, the algorithm has a capability of self-learning and picks up information samples for the learning task. This approach reduces the manual intervention of annotation that is used in passive learning, where the model is trained simply on the user-annotated training data. We compared the performance between our proposed active learning and the traditional passive learning in Table 3.

Table 2
Effectiveness of the subset-based method.

	$\|A\| * \|B\|$	N_b	Reduction Ratio	Pairs completeness
Dataset1	112,020	20,940	0.81	0.75
Dataset2	246,030	56,856	0.77	0.94
Dataset3	955,631	187,433	0.80	0.81

Table 3
Comparison between active learning and passive learning.

	Passive learning classifier		Active learning classifier	
	*F*1	*AUC*	*F*1	*AUC*
Dataset1	0.48	0.72	0.56	0.75
Dataset2	0.61	0.91	0.69	0.91
Dataset3	0.49	0.71	0.60	0.72

Table 4
Evaluation between active learning and passive learning.

	B_PMI		*B_Graph*		*B_LR*		*T_LSA*		*Proposed*	
	*F*1	*AUC*	*F*1	*AUC*	*F*1	*AUC*	*F*1	*AUC*	*F*1	*AUC*
Dataset1	0.21	0.52	0.50	0.55	0.45	0.70	0.39	0.59	0.56	0.75
Dataset2	0.43	0.68	0.69	0.66	0.51	0.87	0.47	0.62	0.69	0.91
Dataset3	0.40	0.59	0.54	0.53	0.46	0.72	0.38	0.58	0.60	0.72

5.4.3. Comparison with baseline methods

In Table 4, we present the performance of different methods in terms of *F*1 and *AUC*, where a higher value indicates a higher performance. From these results, it shows that our active learning-based method achieves much better performance than the other four baseline methods.

5.4.4. Discussion

According to the experimental results, we observed that:

- The baseline*B_ PMI* can detect nearly all of the aliases, but most of them are ranked beyond the top 20. This approach often leads to a large number of general terms identified as true aliases by mistake, which causes the accuracy to be lower. Therefore, this method is frequently used to find some entity-alias pairs that co-occur in the same document.
- In comparison, the baseline *B_Graph* has a higher precision and a much lower recall because the linking limitation prevents a large number of aliases from being related to its given entity. The graph-based method is a good solution for exploring various entity relationships. The baseline *B_Graph* simply considers the shortest path and the source size and only detects the aliases that have at least one edge linkage to the given entity. We find that, to discover more aliases, longer paths should be accounted for. Note that the path length limits the recall, and the source size limits the precision. In [31], researchers explored entity semantic association through capturing information about connecting two entities in an RDF graph, which enhanced the precision. This approach can, thus, be referred to as a graph-based method in the future.
- The baseline *B_LR* considers both string similarity and the social connection probability; of these, the social connection is useful, but the string similarity features lead to more noise. For example, ‘‘Wisconsin Votes’’ and ‘‘Wisconsin High School’’ are often extracted as the aliases of ‘‘Wisconsin State’’ because of their string similarity and social connection.
- The baseline *T_LSA* uses a two-step *LSA* method whose output is subjected to a second run of *LSA*. However, using the context information of the concerned entities is not sufficient to obtain ideal results. This approach promotes the rank of the true positive aliases but might add more false positive aliases. In addition, *T_LSA* cannot effectively address words that have multiple meanings. The baseline *T-LSA* is commonly used for semantic word identification, but it also extracts too many general words, which dramatically decreases the recall. These general words do not denote the same entity as the concerned entity but are closely related to it. For example, the generated alias list for ‘‘Osama bin Ladin’’ contains ‘‘September 11 Attacks’’ and ‘‘The Base’’.

The proposed method in this study combines the advantages of the state-of-the-art methods by employing co-occurrence information, social connections in the entity network, and the alias relevance information for the semantic alias detection.

Table 5
Examples of semantic alias detection.

Entity	Subset-based active learning classifier	Active learning classifier
Wisconsin state	**Wisconsinites**[1], **badger_state**[2], senate_race_in_mid[3], national_health[3], automobile_travel, national_political[3], wisconsin_adults[3], Ronald[3], rdd[3], the state_budget_situation[3], views_of_wisconsin[3], stem_cell[3], homosexuality_and_the_law[3], gubernatorial[3], senator_russ_feingold_in[3], iraq_conflict_after_five[3], ...**dairy_state**[18]	**Badger_state**[1], **badger state**[1], **wisconsinites**[3], our_place_in_the_universe, ronald[4], senator_russ_feingold_in[4], senator_from_wisconsin[4], social_security[4], national_health[4], wisconsin_child_support[4], September_snapshot[4], national_political[4], senate_race_in_mid[4], wisconsin_adults[4],healthcare_system[4], ...**dairy state**[178]
Osama bin Ladin	**The_teacher**[1], **the_doctor**[1], **ustaz**[1], **muhammad_ibrahim**[1], **nur**[1], **mujahid_shaykh**[6], **shaykh_usama_bin_ladin**[6], **the_emir**[6], **the_prince**[6], **usama_bin_muhammad_bin_ladin**[6], the_director[6], **hajj**[12], **abu_fatima**[13], abd_al-hadi_al-iraqi[14], **ayman_al-zawahiri**[15], **usama_bin_laden**[16], **osama_bin_laden**[17]	**Abu_abdallah**[1], **muhammad_ibrahim**[2], **the_doctor**[2], **the_teacher**[2], **ustaz**[2], **nur**[2], **abu_muhammad**[2], **abu_al-mu'iz**[2], **the_emir**[9], **the_prince**[9] **shaykh_usama_bin_ladin**[9], **usama_bin_muhammad_bin_ladin**[9], the_director[9], mujahid_shaykh[9], **hajj**[15], **abu_fatima**[16], abdal_al-hadi_al-iraqi[17], abd_al-hadi_al-iraqi[18], **ayman_al-zawahiri**[19], **osama_bin_laden**[20]

5.4.5. Results demonstration

Table 5 shows the top detected aliases for two concerned entities (i.e., "Wisconsin State" and "Osama bin Ladin") based on the approaches with/without the subset-based methods. Each generated alias is numbered with a superscript, and the results in bold represent the true positive aliases. The first example, "Wisconsin State", comes from Dataset1, which is relatively large and has more noise. The second example comes from Dataset2, which is small and clean. These results in Table 5 illustrate that the proposed subset-based method could improve the performance of the logistic regression classifier significantly on larger data. This approach can effectively filter out the negative aliases and reduce the scope of the comparison. For example, the subset-based method reduced the obtained aliases that result for the entity "wisconsin state" from 178 to 18. Because there is a large amount of noisy data on the Web, the subset-based method is rather useful for obtaining a limited number of relevant entities with respect to the concerned entities.

6. Conclusions

This paper proposes an active-learning-based framework for detecting semantic entity aliases. More specifically, it designs a subset-based method to improve the entity detection accuracy, and then, it trains an active learning classifier to detect the semantic aliases for each concerned entity. This method is suitable for various types of entity aliases, especially for the aliases that are intentionally hidden. Experimental results demonstrate its optimal performance and adaptability compared with four previously proposed methods.

Alias detection is gaining attention in many different applications, including detecting entity nicknames and distinguishing alternative IDs for the same person. Therefore, we plan, in the near future, to extend our research on identifying the same entity/person from multiple domains, for example, detecting the same terrorist when the terrorist uses different identities on LinkedIn, Twitter, and Facebook. In contrast to the problem of semantic alias detection on the Web, more resources (e.g., account profiles, records accesses, friends' connections, and postings of content) are required to address this new challenge.

Acknowledgments

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2011CB302206, the "111 Project" of Ministry of Education and State Administration of Foreign Experts Affairs under Grant No. B14025, the International S&T Cooperation Program of Gansu Province under Grant 2013GS09921, the Chinese National Key Technology R& D Program under Grant 2013BAH19F00, and an HP Labs Innovation Research Program award.

References

[1] A. Arasu, M. Götz, R. Kaushik, On active learning of record matching packages, in: Proceedings of The ACM International Conference on Management of Data (SIGMOD), 2010, pp.783–794.
[2] J. Baldridge, M. Osborne, Active learning for HPSG parse selection, in: Proceedings of the 7th Conference on Natural Language Learning (CONLL'03) at HLT-NAACL 2003, vol. 4, Stroudsburg, PA, USA, 2003, pp. 17–24.
[3] R. Baxter, P. Christen, T. Churches, A comparison of fast blocking methods for record linkage, in: Proceedings of ACM SIGKDD Workshop on Data Cleaning, 2003, pp. 25–27.
[4] I. Bhattacharya, L. Getoor, A latent dirichlet model for unsupervised entity resolution, in: Proceedings of The Sixth SIAM Conference on Data Mining, 2006, pp. 47-58.
[5] D. Bollegalla, T. Honma, Y. Matsuo, M. Ishizuka, Identification of personal name aliases on the web, in: Proceedings of WWW'08, Beijing, China, 2008, pp. 1107–1108.
[6] T. Boongoen, Q. Shen, C. Price, Disclosing false identity through hybrid link analysis, Artificial Intelligence and Law 18 (1) (2010) 77–102.
[7] D.G. Brizan, A.U. Tansell, A survey of entity resolution and record linkage methodologies, Communications of the IIMA 6 (3) (2006) 41–50.
[8] P. Christen, Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Springer, 2012. ISBN 978-3-642-31163-5.

[9] P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Transaction on Knowledge and Data Engineering 24 (9) (2012) 1537–1555.
[10] R.S. Coimbra, D.E. Vanderwall, G.C. Oliveira, Disclosing ambiguous gene aliases by automatic literature profiling, BMC Genomics 11 (2010) S3.
[11] J. Davis, I. Dutra, D. Page, C.V. Santos, Establishing identity equivalence in multi-relational domains, in: Proceedings of the International Conference on Intelligence Analysis, McLean, VA, USA, 2005.
[12] A.K. Elmagarmid, P.G. Ipeirotis, V.S. Verykios, Duplicate record detection: a survey, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 1–16.
[13] L. Gravano, P.G. Ipeirotis, H.V. Jagadish, N. Koudas, S. Muthukrishnan, D. Srivastava, Approximate string joins in a database (almost) for free, in: Proceedings of the 27th International Conference on Very Large Data Bases (VLDB), San Francisco, CA, USA, 2001, pp. 491–500.
[14] R. Holzer, B. Malin, L. Sweeney, Email alias detection using social network analysis, in: Proceedings of the 3rd International Workshop on Link Discovery, Chicago, Illinois, USA, 2005, pp. 52–57.
[15] M.A. Hernlcndez, S.J. Stolfo, Real-world data is dirty: data cleansing and the merge/purge problem, Data Mining and Knowledge Discovery 2 (1) (1998) 9–37.
[16] P. Hsiung, A. Moore, D. Neill, J. Schneider, Alias detection in link data sets, 2005 <http://www.autonlab.org/autonweb/14711>.
[17] L. Jiang, J. Wang, P. Luo, N. An, M. Wang, Towards alias detection without string similarity: an active learning based approach, in: Proceedings of the 35th Annual International ACM SIGIR Conference, 2012.
[18] B.N. Li, M.C. Dong, S. Chao, On decision making support in blood bank information systems, Expert Systems With Applications 34 (2) (2008) 1522–1532.
[19] B.N. Li, M.C. Dong, M.I. Vai, Modelling cardiovascular physiological signals using adaptive Hermite and wavelet basis functions, IET Signal Processing 4 (5) (2010) 588–597.
[20] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR'94), 1994, pp. 3–12.
[21] S.J. Mason, N.E. Graham, Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: statistical significance and interpretation, Quarterly Journal of the Royal Meteorological Society (128) (2002) 2145–2166.
[22] T. Oates, V. Bhat, V. Shanbhag, Using latent semantic analysis to find different names for the same entity in free text, in: Proceedings of the 4th International Workshop on Web Information and Data Management (WIDM'02), 2002, pp. 31–35.
[23] P. Pantel, Alias detection in malicious environments, in: Proceedings of AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection, pp. 14–20, 2006.
[24] S. Sarawagi, A. Bhamidipaty, Interactive deduplication using active learning, in: Proceedings of The Eighth ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2002, pp. 2693–327.
[25] R. Schrag, EAGLE Y2.5 Performance Evaluation Laboratory (PE Lab) Documentation (version 1.5), Information Extraction and Transport, Inc., 2004.
[26] http://alias-i.com/lingpipe/.
[27] http://en.wikipedia.org/wiki/Stopwords.
[28] S. Tejada, C.A. Knoblock, S. Minton, Learning domain-independent string transformation weights for high accuracy object identification, in: Proceedings of The Eighth ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2002, pp. 350–359.
[29] A. Vlachos, Active Learning with Support Vector Machines, Master of Science, School of Informatics, University of Edinburgh, UK, 2004.
[30] http://www.spamarchive.org.
[31] V. Viswanathan, K. Ilango, Ranking semantic relationships between two entities using personalization in context specification, Information Sciences 207 (2012) 35–49.