Alias Detection in Link Data Sets

Paul Hsiung1 and Andrew Moore and Daniel Neill and Jeff Schneider
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{hsiu+g, awm, neill, schneide}@cs.cmu.edu

Keywords: Link Analysis, Information Extraction, Information Sharing and Collaboration, Fusion

Abstract

The problem of detecting aliases - multiple text string identifiers corresponding to the same entity - is increasingly important in the domains of biology, intelligence, marketing, and geoinformatics. Aliases arise from entities who are trying to hide their identities, from people with multiple names, or from words which are unintentionally or even intentionally misspelled. While purely orthographic methods (e.g. string similarity) can help solve unintentional spelling cases, many types of aliases (including those adopted with malicious intent) can fool these methods.

However, if an entity has a changed name in some context, several or all of the set of other entities with which it has relationships can remain stable. Thus, the local social network can be exploited by using the relationships as semantic information.

By applying the best combination of both types of information, the combined algorithm outperforms the ones built solely on one type of information.

1. Introduction

Please refer to the full paper (Hsiung et al) for details.

The premise of a link data set is that one entity represents a unique individual, be it person, word, or even research paper. However, each entity can have many names, which are aliases. A link data set consists of a set each of names and links. Each link contains multiple names and represents an observed relation between the names. In this paper, only the occurrence of relations are used and not the meaning behind the relations.

The way that the intelligence community gathers data is compatible with the way in which link data sets are constructed. Intelligence analysts collect articles (often written in foreign languages) and write down relations between names inside the articles. For example, consider the following web-article2:

Wanted al-Qaeda chief Osama bin Laden and his top aide, Ayman al-Zawahri, have moved out of Pakistan and are believed to have crossed the border back into Afghanistan.

A small number of links will summarize the information, such as the one below:

(Osama bin Laden, Ayman al-Zawahri, al-Qaeda)

This paper focuses on many names corresponding to a single entity. For example, Osama bin Laden is also known as Usama bin Laden. These two strings are orthographically similar and, hence, are easy-to-spot aliases. But there are other, more difficult aliases such as The Prince. To detect these aliases, the social network structure of these names must be exploited. The friends of Osama bin Laden are defined as all the names that have some relationship with Osama bin Laden (i.e. occur in the same link). To exploit the social network, friends of Osama bin Laden are compared with that of The Prince, and semantic measures are computed between these two sets of friends.

2. Measures

If two strings are orthographically similar, they are likely to be versions of the same name. The four orthographic measures used in this paper are described below.

- **String Edit Distance (SED)** The minimum number of insertions, deletions, and substitutions required to transform one string into the other
- **Normalized String Edit Distance (NSED)** SED divided by the maximum length of the two strings being compared
- **Discretized String Edit Distance (DSED)** NSED binarized by a threshold
- **Exponential String Edit Distance (ESED)** ESED(s1, s2) = exp(SED(s1, s2))

---

1 The first author is currently attending University of Southern California. This work was done while he was at CMU.

2 http://uk.news.yahoo.com/040225/323/emvp1.html
The four semantic measures that are computed from friend list are described below:

- **Dot Product (DP)** A name’s list of friends is represented as an occurrence vector of other names with which it appears. The dot product of the occurrence vectors from two names is recorded.
- **Normalized Dot Product (NDP)** Each occurrence vector is normalized by dividing its magnitude before taking the dot product.
- **Common Friends (CF)** This is the number of friends that co-occur with both names, or the sum of the conjunction between two occurrence vectors.
- **Kullback Leibler Distance (KL)** Normalized friends lists can be treated as probability vectors between which KL can then measure the similarity.

3. **Alias Classifier**

In this combined model, the goal is to train a classifier that tests whether two names in a link data set are aliases. For the purpose of training the classifier, positive examples come from hand selected aliases. Negative examples are picked by randomly selecting pairs of names among the link data set. For each pair of names, all measures in both the semantic and orthographic models are calculated and incorporated as **attributes** into the training set. An output attribute called **Alias?** labels each example as positive or negative. Each pair of names represents a row in the training set. Logistic regression has slightly better general performance than other methods, so it is used in subsequent experiments in this paper.

For the task of prediction, a query name that is also a name in the link data set is picked - for instance, **Osama bin Laden**. **Osama bin Laden** is paired with all of the other names in the link data set. The attributes of all the pairs are then computed. The classifier predicts on all the pairs and ranks them by the class conditional scores.

4. **Empirical Evaluation**

In order to evaluate this algorithm, large real-world link data sets filled with alias-rich information are needed. However, the types of people who use aliases are the ones that tend to hide from the general public. Suitable candidates thus include terrorists and spammers. The terrorist information can be obtained from newspaper articles, and spam is always readily available.

Spam-based link data sets are alias-rich because spammers will often create aliases through intentional misspelling to confuse the spam filter. For example, instead of using **mortgage** in an email soliciting a loan, the spammers might use the word **m0rtg@ge** instead.

In a spam-based link data set, each spam email is represented as a separate link, with the names in that link being the word-tokens appearing in the email.

To gather positive training examples for learning, all the aliases for a particular entity are manually collected. Alias pairs are then generated by exhaustively matching up all aliases that belong to that entity. Each alias pair corresponds to a positive example in the training set.

Further details concerning the data sets are described in the full paper (Hsiung et al.). The actual datasets are located at: http://www-2.cs.cmu.edu/~awm/mnop_data/

5. **Empirical Results**

The combined classifier is tested alongside one that is built strictly from orthographic attributes and another one from semantic attributes.

K-fold cross-validation is used to evaluate all three classifiers. For each “fold,” an entity with known aliases and all related alias pairs are removed from the training set. Then each classifier is trained on the remaining training set. Since the classifiers test whether two names are aliases, a query is performed with one name of the removed entity against all possible names in the link data set to create the prediction set. From the sorted classifier scores of the prediction set, the ranks of the correct aliases (other names of the removed entity) are identified, and an ROC curve is produced. For the next “fold,” another query on another name of the same removed entity is performed, and another ROC curve is produced; this is repeated until all the names of the entity are exhausted. When that happens, the next entity with known aliases is used.

All ROC curves are represented by first normalizing all the axes and then averaging all the curves. A good measure of performance for each classifier is the area under the averaged ROC curve (AUC).

The following table shows the AUC for all three classifiers on the three link data sets:

<table>
<thead>
<tr>
<th>Data set</th>
<th>Combined</th>
<th>Semantic</th>
<th>Orthographic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terrorists</td>
<td>0.944</td>
<td>0.902</td>
<td>0.735</td>
</tr>
<tr>
<td>HsiungSpam</td>
<td>0.805</td>
<td>0.689</td>
<td>0.721</td>
</tr>
<tr>
<td>ArchiveSpam</td>
<td>0.721</td>
<td>0.714</td>
<td>0.604</td>
</tr>
</tbody>
</table>

Clearly, the combined classifier is able to outperform the semantic and the orthographic classifiers. This implies that the combined classifier is able to take advantage of the unique information supplied by both types of measures.

**References**


Full citation is given in the above paper.