

# Mining for Personal Name Aliases on the Web

Danushka Bollegala\*, Taiki Honma, Yutaka Matsuo and Mitsuru Ishizuka

The University of Tokyo, Hongo 7-3-1, Tokyo, 113-8656, Japan

{danushka, honma}@mi.ci.i.u-tokyo.ac.jp,

matsuo@biz-model.t.u-tokyo.ac.jp, ishizuka@i.u-tokyo.ac.jp

## ABSTRACT

We propose a novel approach to find aliases of a given name from the web. We exploit a set of known names and their aliases as training data and extract lexical patterns that convey information related to aliases of names from text snippets returned by a web search engine. The patterns are then used to find candidate aliases of a given name. We use anchor texts and hyperlinks to design a word co-occurrence model and define numerous ranking scores to evaluate the association between a name and its candidate aliases. The proposed method outperforms numerous baselines and previous work on alias extraction on a dataset of personal names, achieving a statistically significant mean reciprocal rank of 0.6718. Moreover, the aliases extracted using the proposed method improve recall by 20% in a relation-detection task.

## Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval

## General Terms

Algorithms

## Keywords

Name alias extraction, Semantic Web, Web Mining

## 1. INTRODUCTION

Precisely identifying entities in web documents is necessary for various tasks in the Semantic Web such as relation extraction, metadata extraction, search and integration of data. Nevertheless, identification of entities on the web is difficult for two fundamental reasons: first, different entities can share the same name (*lexical ambiguity*); secondly, a single entity can be designated by multiple names (*referential ambiguity*). As an example of lexical ambiguity the name *Jim Clark* is illustrative. Aside from the two most popular namesakes, the formula-one racing champion and the founder of Netscape, at least 10 different people are listed among the top 100 results returned by Google for the name.

\*Research Fellow of the Japan Society for the Promotion of Science (JSPS)

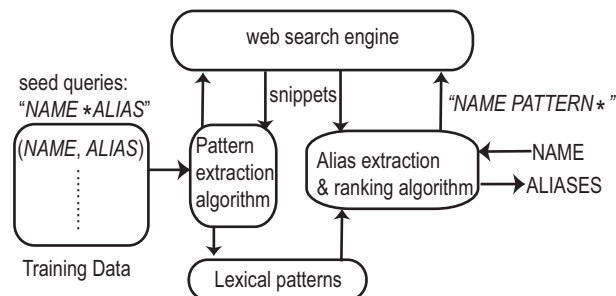


Figure 1: Outline of the alias extraction method

On the other hand, referential ambiguity occurs because people use different names to refer to the same entity on the web. For example, the American movie star *Will Smith* is often called the *fresh prince* in web contents. Although lexical ambiguity, particularly ambiguity related to personal names, has been explored extensively in the previous studies of name disambiguation [1], the problem of referential ambiguity of entities on the web has received much less attention. In this paper, we specifically examine on the problem of automatically extracting the various references on the web to a particular entity. In contrast to the real name of an entity, we use the term *alias* to describe words or multi-word expressions that are used to refer to the entity (e.g., *fresh prince* is an alias for *Will Smith*).

## 2. METHOD

The proposed method is outlined in Fig.1 and comprises two main components: pattern extraction, and alias extraction and ranking. To identify the various ways in which information related to aliases are represented on the web, we use a seed list of (name, alias) pairs as queries to a web search engine and download text-snippets that contain both a name and its alias. Here, we use the wildcard operator \* to perform a *NEAR* query. The operator \* matches with one or more words in a snippet. Figure 2 shows a snippet retrieved for the query "Will Smith \* The Fresh Prince". In Fig.2 the

...Rock the House, the duo's debut album of 1987,  
demonstrated that **Will Smith**, aka **the Fresh Prince**,  
was an entertaining and amusing storyteller...

Figure 2: A snippet returned for the query "Will Smith \* The Fresh Prince" by Google

Table 1: Lexical patterns with the highest  $F$ -scores

pattern	$F$ -score
* aka [NAME]	0.335
[NAME] aka *	0.322
[NAME] better known as *	0.310
[NAME] alias *	0.286
[NAME] also known as *	0.281
* nee [NAME]	0.225
[NAME] nickname *	0.224
* whose real name is [NAME]	0.205

snippet contains *aka* (i.e. *also known as*), which indicates the fact that *fresh prince* is an alias for *Will Smith*. In addition to *a.k.a.*, numerous clues exist such as *nicknamed*, *alias*, *real name is*, *nee*, which are used on the web to represent aliases of a name. We create lexical patterns from snippets by replacing the name and alias respectively by two variables [NAME] and [ALIAS], and extracting the phrases that appear in between. We repeat this procedure with the reversed query, “*alias \* name*” to extract patterns in which alias precedes the name (e.g., [ALIAS] *is an alias for* [NAME]). The created patterns can then be used to query a search engine to find candidate aliases of a given name. Specifically, we substitute the given name for the variable [NAME] and \* for the variable [ALIAS] and download snippets. The words that match the wildcard operator in snippets are selected as candidate aliases for the given name. Using 50 pairs of names and aliases as seeds, we extracted over 8000 lexical patterns. Patterns are ranked according to their  $F$ -scores on training data. Top ranking patterns are shown in Table 1.

Considering the noise in web-snippets, candidates extracted using a set of shallow lexical patterns might include some invalid aliases. Therefore, it is imperative that we identify the candidates which are most likely to be correct aliases of a given name. We model this problem of alias recognition as one of ranking candidate aliases with respect to a given name such that the candidates which are most likely to be correct aliases are assigned a higher rank. We define various ranking scores to measure the association between a name and a candidate alias using two approaches: hyperlink structure on the web and page-counts retrieved from a search engine. Inbound anchor texts of a url provide useful semantic clues related to the resource represented by the url. We define two words  $a$  and  $b$  as *co-occurring*, if they appear in at least two different inbound anchor texts  $A$  and  $B$ , respectively, in a url  $u$ . Using this definition, we compute 18 popular co-occurrence measures. Hubs in the hyperlink graph are automatically detected and co-occurrences in hubs are appropriately adjusted. Moreover, four page-count-based association measures are computed [2]. All ranking scores are integrated using raking support vector machines to leverage a robust ranking function. Because of the limited availability of space, we omit the details of the ranking algorithm.

### 3. EVALUATION

We evaluate the proposed alias extraction algorithm on three datasets<sup>1</sup>: English personal names (50 names), Japanese personal names (100 names) and English place names (50 U.S. states). Personal names datasets include people from various fields of cinema, sports, politics and science. As shown in Table 2, the proposed method reports high mean

<sup>1</sup>www.miv.t.u-toyko.ac.jp/danushka/aliasdata.zip

Table 2: Overall performance

Dataset	MRR	Average Precision
English Personal Names	0.6150	0.6865
English Place Names	0.8159	0.7819
Japanese Personal Names	0.6718	0.6646

Table 3: Aliases extracted by the proposed method

Real Name	Extracted Aliases
David Hasselhoff	<b>hoff, michael knight, michael</b>
Courteney Cox	<b>dirt lucy, lucy, monica</b>
Al Pacino	<b>michael corleone</b>
Teri Hatcher	<b>susan mayer, susan, mayer</b>
Texas	<b>lone star state, lone star, lone</b>
Vermont	<b>green mountain state, green,</b>
Wyoming	<b>equality state, cowboy state</b>
Hideki Matsui	<b>Godzilla, nishikori, matsui</b>

reciprocal rank (MRR) and average precision on all datasets. Moreover, the proposed method outperforms a previously proposed alias extraction algorithm by Hokama and Kitagawa [3] (MRR=0.6314 on Japanese names dataset) which uses manually created patterns specific to Japanese names. Top ranking aliases as extracted by the proposed method for some names are shown in Table 3. Overall, in Table 3 the proposed method extracts most aliases assigned in the manually created gold standard (shown in bold).

Table 4: Effect of aliases on relation detection

Real name only			Real name and top alias		
Precision	Recall	F	Precision	Recall	F
.4812	.7185	.4792	.4833	.9083	.5918

We evaluate the extracted aliases on a relation detection task. First, we manually classify 50 people in the English personal names dataset into four categories: *music*, *politics*, *movies*, and *sports*. Then we measure the association between two people using the WebPMI [2] and perform group average agglomerative clustering to form four clusters. Clustering accuracies with and without using aliases are shown in Table 4. The use of aliases significantly improves recall (ca. 20%) and consequently the  $F$  score. By considering not only real names but also aliases, it is possible to discover relations that are unidentifiable solely using real names.

### 4. CONCLUSION

We proposed a lexical-pattern-based approach to extract aliases for a given name. The extracted candidates were ranked using various ranking scores computed using the hyperlink structure on the web and page-counts retrieved from a search engine. The proposed method reported high MRR scores on three different datasets and significantly improved recall in a relation detection task.

### 5. REFERENCES

- [1] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proc. of WWW'05*, pages 463–470, 2005.
- [2] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proc. of WWW'07*, pages 757–766, 2007.
- [3] T. Hokama and H. Kitagawa. Extracting mnemonic names of people from the web. In *Proc. of 9th Intl. Conf. on Asian Digital Libraries (ICADL'06)*, pages 121–130, 2006.