# DISCOVERY OF ALIASES NAME FROM THE WEB

**N.Thilagavathy***

**T.Balakumaran****

**P.Ragu****

**R.Ranjith kumar****

**Abstract**

An individual is typically referred by numerous name aliases on the web. Accurate identification of aliases of a given person name is useful in various web related tasks such as information retrieval, sentiment analysis, personal name disambiguation, and relation extraction. We propose a method to extract aliases of a given personal name from the web. Given a personal name, the proposed method first extracts a set of candidate aliases. Second, we rank the extracted candidates according to the likelihood of a candidate being a correct alias of the given name. We propose a novel, automatically extracted lexical pattern-based approach to efficiently extract a large set of candidate aliases from snippets retrieved from a web search engine. We define numerous ranking scores to evaluate candidate aliases using three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web. To construct a robust alias detection system, we integrate the different ranking scores into a single ranking function using ranking support vector machines. We evaluate the proposed method on three data sets: an English personal names data set, an English place names data set, and a Japanese personal names data set. The proposed method outperforms numerous baselines and previously proposed name alias extraction methods, achieving a statistically significant mean reciprocal rank (MRR) of 0.67. Experiments carried out using location names and Japanese personal names suggest the possibility of extending the proposed method to extract aliases for different types of named entities, and for different languages. Moreover, the aliases extracted using the proposed method are successfully utilized in an information retrieval task and improve recall by 20 percent in a relation detection task.

* Asst Professor, Sri ManakulaVinayagar Engineering College, Madagadipet, Puducherry.

** Final year, B.Tech (IT Dept), Sri ManakulaVinayagar Engineering College, Madagadipet, Puducherry.

## 1. Introduction

Searching for information about people in the web is one of the most common activities of internet users. Around 30 percent of search engine queries include person names. However, retrieving information about people from web search engines can become difficult when a person has nicknames or name aliases. For example, the famous Japanese major league baseball player Hideki Matsui is often called as Godzilla on the web. A newspaper article on the baseball player might use the real name, Hideki Matsui,whereas a blogger would use the alias, Godzilla, in a blog entry. We will not be able to retrieve all the information about the baseball player, if we only use his real name. Identification of entities on the web is difficult for two fundamental reasons: first, different entities can share the same name (i.e., lexical ambiguity); second, a single entity can be designated by multiple names (i.e., referential ambiguity). For example, the lexical ambiguity consider the name Jim Clark. 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. 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, 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 of a particular entity.

For an entity, we define the set A of its aliases to be the set of all words or multiword expressions that are used to refer to e on the web. For example, Godzilla is a one-word alias for Hideki Matsui, whereas alias the Fresh Prince contains three words and refers to Will Smith. Various types of terms are used as aliases on the web. For instance, in the case of an actor, the name of a role or the title of a drama (or a movie) can later become an alias for the person (e.g., Fresh Prince, Knight Rider). Titles or professions such as president, doctor, professor, etc., are also frequently used as aliases. Variants or abbreviations of names such as Bill for William, and acronyms such as JFK for John Fitzgerald Kennedy are also types of name aliases that are observed frequently on the web.

Identifying aliases of a name are important in information retrieval. In information retrieval, to improve recall of a web search on a person name, a search engine can automatically expand a

query using aliases of the name . In our previous example, a user who searches for Hideki Matsui might also be interested in retrieving documents in which Matsui is referred to as Godzilla. Consequently, we can expand a query on Hideki Matsui using his alias name Godzilla.

The semantic web is intended to solve the entity disambiguation problem by providing a mechanism to add semantic metadata for entities. However, an issue that the semantic web currently faces is that insufficient semantically annotated web contents are available. Automatic extraction of meta data can accelerate the process of semantic annotation. For named entities, automatically extracted aliases can serve as a useful source of metadata, thereby providing a means to disambiguate an entity. Identifying aliases of a name are important for extracting relations among entities. For example, Matsuoetal. propose a social network extraction algorithm in which they compute the strength of the relation between two individuals A and B by the web hits for the conjunctive query, "A" and "B". However, both persons A and B might also appear in their alias names in web contents. Consequently, by expanding the conjunctive query using aliases for the names, a social network extraction algorithm can accurately compute the strength of a relationship between two persons.

Along with the recent rapid growth of social media such as blogs, extracting and classifying sentiment on the web has received much attention. Typically, a sentiment analysis system classifies a text as positive or negative according to the sentiment expressed in it. However, when people express their views about a particular entity, they do so by referring to the entity not only using the real name but also using various aliases of the name. By aggregating texts that use various aliases to refer to an entity, a sentiment analysis system can produce an informed judgment related to the sentiment.

We propose a fully automatic method to discover aliases of a given personal name from the web. Our contribution can be summarized as follows:

We propose a lexical pattern-based approach to extract aliases of a given name using snippets returned by a web search engine. The lexical patterns are generated automatically using a set of

real world name alias data. We evaluate the confidence of extracted lexical patterns and retain the patterns that can accurately discover aliases for various personal names. Our pattern extraction algorithm does not assume any language specific preprocessing such as part-of-speech tagging or dependency parsing, etc., which can be both inaccurate and computationally costly in web-scale data processing.

- To select the best aliases among the extracted candidates, we propose numerous ranking scores based upon three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web. Moreover, using real-world name alias data, we train a ranking support vector machine to learn the optimal combination of individual ranking scores to construct a robust alias extraction method.

- We conduct a series of experiments to evaluate the various components of the proposed method. We compare the proposed method against numerous baselines and previously proposed name alias extraction methods on three data sets: an English personal names data set, an English place names data set, and a Japanese personal names data set. Moreover, we evaluate the aliases extracted by the proposed method in an information retrieval task and a relation extraction task.

## 2. PROBLEM DEFINITION

### 2.1 Existing system

A method to extract aliases of a given personal name from the web. First extracts a set of candidate aliases. Second, rank the extracted candidates according to the likelihood of a candidate being a correct alias of the given name. Automatically extracted lexical pattern-based approach to efficiently extract a large set of candidate aliases from snippets retrieved from a web search engine. The numerous ranking scores to evaluate candidate aliases using three

approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web.

**2.2.Methods**

1. Extracting lexical patterns from snippets.
2. Ranking of candidate.
3. Lexical pattern frequency
4. Co-occurrences in anchor texts.
   a. CF
   b. TDIDF
   c. CS
   d. LLR
   e. PMI
   f. HD
   g. COSINE
   h. OVERLAP
   i. DICE
5. Hub discounting
6. Page-count-based association measures
7. Training
8. Data set

**2.3 Existing system Architecture**

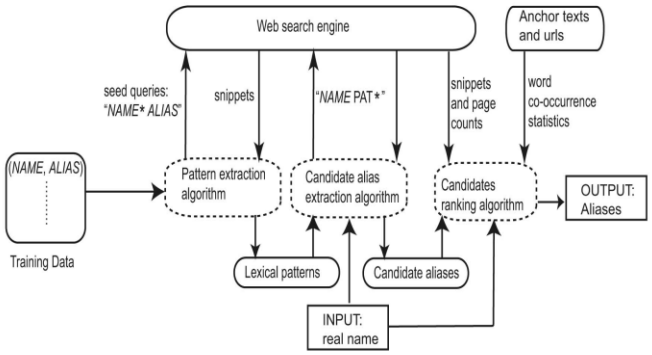


**Fig. 1: Alias name extraction architecture**

## 3. METHOD

The proposed method is outlined in Fig. 1 and comprises two main components: pattern extraction, and alias extraction and ranking. Using a seed list of name-alias pairs, we first extract lexical patterns that are frequently used to convey information related to aliases on the web. The extracted patterns are then used to find candidate aliases for a given name. We define various ranking scores using the hyperlink structure on the web and page counts retrieved from a search engine to identify the correct aliases among the extracted candidates.

### 3.1 Extracting Lexical Patterns from Snippets

Many modern search engines provide a brief text snippet for each search result by selecting the text that appears in the web page in the proximity of the query. Such snippets provide valuable information related to the local context of the query. For names and aliases, snippets convey useful semantic clues that can be used to extract lexical patterns that are frequently used to express aliases of a name. For example, consider the snippet returned by Google2 for the query "Will Smith _ The Fresh Prince." Here, we use the wildcard operator _ to perform a NEAR query and it matches with one or more words in a snippet. In Fig. 2 the snippet contains aka (i.e., also known as), which indicates the fact that fresh prince is an alias for Will Smith. In addition to aka, numerous clues exist such as nicknamed, alias, real name is nee, which are used on the web to represent aliases of a name. Consequently, we propose the shallow pattern extraction method illustrated in Fig. 3 to capture the various ways in which information about aliases of names is expressed on the web. Lexico-syntactic patterns have been used in numerous related tasks such as

extracting hypernyms [14] and meronyms [15]. Given a set S of (NAME, ALIAS) pairs, the function ExtractPatterns returns a list of lexical patterns that frequently connect names and their aliases in web snippets. For each (NAME, ALIAS) pair in S, the GetSnippets function downloads snippets from a web search engine for the query "NAME _ ALIAS." Then, from each snippet, the Create-Pattern function extracts the sequence of words that appear between the name and the alias. Results of our preliminary experiments demonstrated that consideration of words that fall outside the name and the alias in snippets did not improve performance. Finally, the real name and the alias in the snippet are, respectively, replaced by two variables [NAME] and [ALIAS] to create patterns. Our definition of lexical patterns includes patterns that contain words as well as symbols such as punctuation markers. For example, from the snippet shown in Fig. 2, we extract the pattern [NAME], aka [ALIAS]. We repeat the process described above for the reversed query, "ALIAS _ NAME" to extract patterns in which the alias precedes the name. In our experiments, we limit the number of matched words with "_" to a maximum of five words. Because snippets returned by web search engines are very short in length compared to the corresponding source documents, increasing the matching window beyond five words did not produce any additional lexical patterns. Once a set of lexical patterns is extracted, we use the patterns to extract candidate aliases for a given name as portrayed in Fig. 4. Given a name, NAME and a set, P of lexical patterns, the function ExtractCandidates returns a list of candidate aliases for the name. We associate the given name with each pattern, p in the set of patterns, P and produce queries of the form: "NAME p_." Then, the GetSnippets function downloads a set of snippets for the query. Finally, the GetNgrams function extracts continuous sequences of words (n-grams) from the beginning of the part that matches the wildcard operator _ . Experimentally, we selected up to five grams as candidate aliases. Moreover, we removed candidates that contain only stop words such as a, an, and the. For example, assuming that we retrieved the snippet in Fig. 3 for the query "Will Smith aka_," theprocedure described above extracts the fresh and the fresh prince as candidate aliases. For efficiency reasons, we limit the number of snippets downloaded by the function GetSnippets to a maximum of 100 in both Algorithm 3.1 and 3.2. In Google it is possible to retrieve 100 snippets by issuing only a single query by setting the search parameter num to 100, thereby reducing the number queries required in practice.

### 3.2 Ranking of Candidates

Considering the noise in web snippets, candidates extracted by the shallow lexical patterns might include some invalid aliases. From among these candidates, we must identify those, which are most likely to be correct aliases of a given name. We model this problem of alias recognition as one of ranking candidates with respect to a given name such that the candidates, who are most likely to be correct aliases are assigned a higher rank. First, we define various ranking scores to measure the association between a name and a candidate alias using three different approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web. Next, we describe the those three approaches in detail**.**

### 3.3 Lexical Pattern Frequency

In Section 3.1 we presented an algorithm to extract numerous lexical patterns that are used to describe aliases of a personal name. As we will see later in Section 4, the proposed pattern extraction algorithm can extract a large number of lexical patterns. If the personal name under consideration and a candidate alias occur in many lexical patterns, then it can be considered as a good alias for the personal name. Consequently, we rank a set of candidate aliases in the descending order of the number of different lexical patterns in which they appear with a name. The lexical pattern frequency of an alias is analogous to the document frequency (DF) popularly used in information retrieval.

### 3.4 Co-Occurrences in Anchor Texts

Anchor texts have been studied extensively in information retrieval and have been used in various tasks such as synonym extraction, query translation in cross-language information retrieval, and ranking and classification of web pages [16]. Anchor texts are particularly attractive because they not only contain concise texts, but also provide links that can be considered as expressing a citation. We revisit anchor texts to measure the association between a name and its aliases on the web. Anchor texts pointing to a url provide useful semantic clues related to the resource represented by the url. For example, if the majority of inbound anchor texts of a url contain a

personal name, it is likely that the remainder of the inbound anchor texts contain information about aliases of the name. Here, we use the term inbound anchor texts to refer the set of anchor texts pointing to the same url. We define a name p and a candidate alias x as co- occurring, if p and x appear in two different inbound anchor texts of a url. Moreover, we define co-occurrence frequency (CF) as the number of different urls in which they co-occur. It is noteworthy that we do not consider co-occurrences of an alias and a name in the same anchor text. For example, consider the picture of Will Smith shown in Fig. 5. Fig. 5 shows a picture of Will Smith being linked to by four different anchor texts. According to our definition of co-occurrence, Will Smith, and fresh prince are considered as co-occurring

To measure the strength of association between a name and a candidate alias, using Table 1, we define nine popular co-occurrence statistics: CF, tfidf measure (tfidf), chisquared measure (CS), Log-likelihood ratio (LLR), hypergeometric distributions (HG), cosine measure (cosine), overlap measure (overlap), and Dice coefficient (Dice). Next, we describe the computation of those association measures in detail.

### 3.4.1 CF

This is the simplest of all association measures and was defined already in the previous section. The value k in Table 1 denotes the CF of a candidate alias x and a name p. Intuitively, if there are many urls, which are pointed to by anchor texts that contain a candidate alias x and a name p, then it is an indication that x is indeed a correct alias of the name p.

### 3.4.2 tfidf

The CF is biased toward highly frequent words. A word that has a high frequency in anchor texts can also report a high co-occurrence with the name.

### 3.4.3 CS

The $x^2$ measure has been used as a test for dependence between two words in various natural language processing tasks including collocation detection, identification of translation pairs in aligned corpora, and measuring corpus similarity.

### 3.4.4 LLR

The LLR [19] is defined as the ratio between the likelihoods of two alternative hypotheses: that the name p and the candidate alias x are independent or the name p and that the candidate alias x are dependent. Likelihood ratios have been used often in collocation discovery.

### 3.4.5 Pointwise Mutual Information (PMI)

PMI [20] is a measure, that is, motivated by information theory; it is intended to reflect the dependence between two probabilistic events.

### 3.4.7 Cosine

The cosine is widely used to compute the association between words. For strength of association between elements in two sets, X and Y can be computed using the cosine measure

### 3.4.8 Overlap

The overlap between two sets X and Y. Assuming that X and Y , respectively, represent occurrences of name p and candidate alias x. We define a ranking score based on the overlap to evaluate the appropriateness of a candidate alias.

### 3.4.9 Dice

the Dice to retrieve collocations from large textual corpora.

### 3.5 Hub Discounting

A frequently observed phenomenon related to the web is that many pages with diverse topics link to so-called hubs such as Google, Yahoo, or MSN. Two anchor texts might link to a hub for entirely different reasons. Therefore, cooccurrences coming from hubs are prone to noise. Consider the situation shown in Fig. 6 where a certain web page is linked to by two sets of anchor texts. One set of anchor texts contains the real name for which we must find aliases, whereas the other set of anchor texts contains various candidate aliases. If the majority of anchor texts linked to a particular web site use the real name to do so, then the confidence of that page as a source of information regarding the person whom we are interested in extracting aliases increases. We use this intuition to compute a simple discounting measure for co-occurrences in hubs as follows.

### 3.6 Page-Count-Based Association Measures

In Section 3.4, we defined various ranking scores using anchor texts. However, not all names and aliases are equally well represented in anchor texts. Consequently, in this section, we define word association measures that consider co-occurrences not only in anchor texts but in the web overall. Page counts retrieved from a web search engine for the conjunctive query, "p and x," for a name p and a candidate alias x can be regarded as an approximation of their cooccurrences in the web. We compute popular word association measures using page counts returned by a search engine.

Three method WebDice, WebPMI, Conditional Probability.

**REFERENCES**

[1] R. Guha and A. Garg, "Disambiguating People in Search," technical report, Stanford Univ., 2004.

[2] J. Artiles, J. Gonzalo, and F. Verdejo, "A Testbed for People Searching Strategies in the WWW," Proc. SIGIR '05, pp. 569-570, 2005.

[3] G. Mann and D. Yarowsky, "Unsupervised Personal Name Disambiguation," Proc. Conf. Computational Natural Language Learning (CoNLL '03), pp. 33-40, 2003.

[4] R. Bekkerman and A. McCallum, "Disambiguating Web Appearances of People in a Social Network," Proc. Int'l World Wide Web Conf. (WWW '05), pp. 463-470, 2005.

[5] G. Salton and M. McGill, Introduction to Modern Information Retreival. McGraw-Hill Inc., 1986.

[6] M. Mitra, A. Singhal, and C. Buckley, "Improving Automatic Query Expansion," Proc. SIGIR '98, pp. 206-214, 1998.

[7] P. Cimano, S. Handschuh, and S. Staab, "Towards the Self-Annotating Web," Proc. Int'l World Wide Web Conf. (WWW '04), 2004.

[8] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka, "Polyphonet: An Advanced Social Network Extraction System," Proc. WWW '06, 2006.

[9] P. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," Proc. Assoc. for Computational Linguistics (ACL '02), pp. 417-424, 2002.

[10] A. Bagga and B. Baldwin, "Entity-Based Cross-Document Coreferencing Using the Vector Space Model," Proc. Int'l Conf. Computational Linguistics (CO LING '98), pp. 79-85, 1998.