A Co-occurrence Graph-based Approach for Personal Name Alias Extraction from Anchor Texts

Danushka Bollegala [*]
The University of Tokyo
7-3-1, Hongo, Tokyo,
113-8656, Japan
danushka@mi.ci.i.u-tokyo.ac.jp

Yutaka Matsuo
National Institute of Advanced Industrial Science and Technology
1-18-13, Sotokanda, Tokyo,
101-0021, Japan
y.matsuo@aist.go.jp

Mitsuru Ishizuka
The University of Tokyo
7-3-1, Hongo, Tokyo,
113-8656, Japan
ishizuka@i.u-tokyo.ac.jp

Abstract

A person may have multiple name aliases on the Web. Identifying aliases of a name is important for various tasks such as information retrieval, sentiment analysis and name disambiguation. We introduce the notion of a word co-occurrence graph to represent the mutual relations between words that appear in anchor texts. Words in anchor texts are represented as nodes in the co-occurrence graph and an edge is formed between nodes which link to the same url. For a given personal name, its neighboring nodes in the graph are considered as candidates of its aliases. We formalize alias identification as a problem of ranking nodes in this graph with respect to a given name. We integrate various ranking scores through support vector machines to leverage a robust ranking function and use it to extract aliases for a given name. Experimental results on a dataset of Japanese celebrities show that the proposed method outperforms all baselines, displaying a MRR score of 0.562.

1 Introduction

Searching for information about people in the Web is one of the most common activities of Internet users. Around 30% of search engine queries include person names (Guha and Garg, 2004). However, an individual might have multiple nicknames or *aliases* on the Web. For example, the famous Japanese major league baseball player *Hideki Matsui* is often called as *Godzilla* in web contents. Identifying aliases of a name is important in various tasks such as information retrieval (Salton and McGill, 1986), sentiment analysis (Turney, 2002) and name disambiguation (Bekkerman and McCallum, 2005).

In information retrieval, to improve recall of a web search on a person name, a search engine can automatically expand the 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*. People use different aliases when expressing their opinions about an entity. By aggregating texts written on an individual that use various aliases, a sentiment analysis system can make an informed judgment on the sentiment. Name disambiguation focuses on identifying different individuals with the same name. For example, for the name *Jim Clark*, aside from the two most popular namesakes - the formula-one racing champion and the founder of Netscape - at least ten different people are listed among the top 100 results returned by Google for the name. Although namesakes have identical names, their nicknames usually differ. Therefore, a name disambiguation algorithm can benefit from the knowledge related to name aliases.

We propose an alias extraction method that exploits anchor texts and the links indicated by the anchor texts. Link structure has been studied extensively in information retrieval and has been found to be useful in various tasks such as ranking of web pages, identification of hub-authority

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

sites, text categorization and social network extraction (Chakrabarti, 2003). Anchor texts pointing to an url provide useful semantic clues regarding the resource represented by the url.

If the majority of inbound anchor texts of an url contain a person name, then it is likely that the remainder of the anchor texts contain information about aliases of the name. For example, an image of *Hideki Matsui* on a web page might be linked using the real name, *Hideki Matsui*, as well as aliases *Godzilla* and *Matsu Hide*. However, extracting aliases from anchor texts is a challenging problem due to the noise in both link structure and anchor texts. For example, web pages of extremely diverse topics link to yahoo.com using various anchor texts. Moreover, given the scale of the Web, broken links and incorrectly linked anchor texts are abundant. Naive heuristics are insufficient to extract aliases from anchor texts.

Our main contributions are summarized as follows:

- We introduce **word co-occurrence graphs** to represents words that appear in anchor texts and formalize the problem of alias extraction as a one of ranking nodes in the graph with respect to a given name.

- We define various ranking scores to evaluate the appropriateness of a word as an alias of a name. Moreover, the ranking scores are integrated using support vector machines to leverage a robust alias detection method.

2 Related Work

Hokama and Kitagawa (2006) propose an alias extraction method that is specific to Japanese language. For a given name p, they search for the query * *koto p* [1] and extract the words that match the asterisk. However, *koto* is highly ambiguous and extracts lots of incorrect aliases. Moreover, the method cannot extract aliases when a name and its aliases appear in separate documents.

Anchor texts and link structure have been employed in synonym extraction (Chen et al., 2003) and translations extraction (Lu et al., 2004). Chen et al. (2003) propose the use of hyperlink structure within a particular domain to generate a domain-specific thesaurus. First, a set of high quality websites from a given domain is selected. Second, several link analysis techniques are used to remove noisy links and the navigational structure of the website is converted into a content structure. Third, pointwise mutual information is applied to identify phrases within content structures to create a domain specific thesaurus. They evaluate the thesaurus in a query expansion task. Anchor texts written in different languages that point the same object have been used in cross-language information retrieval (CLIR) to translate user queries. Lu et al. (2004) extend this idea by associating anchor texts written using a pivotal third language to find translations of queries.

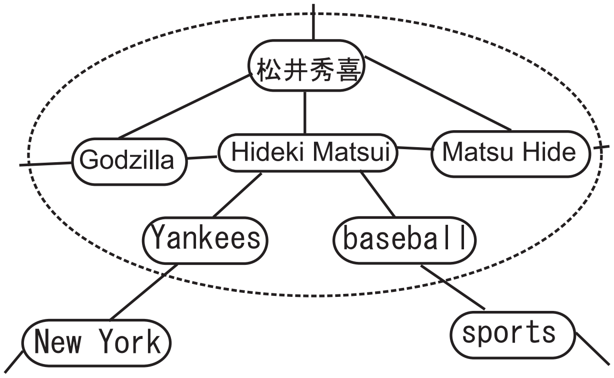

Figure 1: Co-occurrence graph for *Hideki Matsui*

3 Method

3.1 Outline

We introduce *word co-occurrence graph*, an undirected graph, to represent words that appear in anchor texts. For each word that appears in the vocabulary of words in anchor texts, we create a node in the graph. Two words are considered as co-occurring if two anchor texts containing these words link to the same url. An edge is formed between two nodes if the words represented by those nodes co-occur. Figure 1 illustrates a portion of the co-occurrence graph in the proximity of *Hideki Matsui* as extracted by this method from anchor texts.

Representing words that appear in anchor texts as a graph enables us to capture the complex inter-relations between the words. Words in inbound anchor texts of an url contain important semantic clues

[1] *koto* is written in hiragana and and means *also known as*

regarding the resource represented by the url. Such words form a clique in the co-occurrence graph, indicating their close connectivity. Moreover, co-occurrence graphs represent indirect relationships between words. For example, in Figure 1 *Hideki Matsui* is connected to *New York* via *Yankees*.

We model the problem of extracting aliases as a one of ranking nodes in the co-occurrence graph with respect to a real name. Usually, an individual has just one or two aliases. A name alias extraction algorithm must identify the correct aliases among a vast number of related terms for an individual.

3.2 Word Co-occurrence Graph

Let V be the vocabulary of words w_i that appear in anchor texts. The boolean function $A(a_i, w_i)$ returns true if the anchor text a_i contains the word w_i. Moreover, let the boolean function $L(a_i, u_i)$ to be true if the anchor text a_i points to url u_i. Then two words w_i, w_j are defined to be *co-occurring* in a url u, if $A(a_i, w_i) \wedge A(a_j, w_j) \wedge L(a_i, u) \wedge L(a_j, u)$ is true for at least one pair of anchor texts $(a_i,\, a_j)$. In other words, two words are said to co-occur in an url if at least one inbound pair of anchor texts contains the two words. Moreover, we define the number of co-occurrences of w_i and w_j to be the number of different urls they co-occur.

We define *word co-occurrence graph*, $G(V, E)$ (V is the set of nodes and E is the set of edges) as an undirected graph where each word w_i in vocabulary V is represented by a node in the graph. Because one-to-one mapping pertains between a word and a node, for simplicity we use w_i to represent both the word and the corresponding node in the graph. An edge $e_{ij} \in E$ is created between two nodes w_i, w_j if they co-occur. Given a personal name p, represented by a node p in the co-occurrence graph, our objective is to identify the nodes that represent aliases of p. We rank the nodes in the graph with respect to p such that more likely a node is an alias of p, the higher the rank it is assigned. According to our definition, a node that lies n hops away from p has an n-order co-occurrence with p. Considering the fact that a single web page might link to many pages with diverse topics, higher order co-occurrences with p (i.e. nodes that appear further from p) are unreliable as aliases of p. Consequently, we limit $C(p)$, the set of candidate aliases of p, to nodes which are directly connected to p in the graph. In Figure 1 candidates of *Hideki Matsui* fall inside the dotted ellipse.

Table 1: Contingency table for a candidate alias x

	x	$C(p) - \{x\}$	
p	k	$n - k$	n
$V - \{p\}$	$K - k$	$N - n - K + k$	$N - n$
V	K	$N - K$	N

3.3 Ranking of Candidates

To evaluate the strength of co-occurrence between a candidate alias and the real name, for each candidate alias x in $C(p)$ we create a contingency table as shown in Table 1. In Table 1, the first row represents candidates of p and the first column represents nodes in the graph. Therein, k is the number of urls in which p and x co-occur, K is the number of urls in which at least one inbound anchor text contains the candidate x, n is the number of urls in which at least one inbound anchor text contains p and N is the total number of urls in the crawl. Next, we define various ranking scores based on Table 1.

Simplest of all ranking scores is the *link frequency* (lf). We define link frequency of an candidate x as the number of different urls in which x and p co-occur. This is exactly the value of k in Table 1.

Link frequency is biased towards highly frequent words. A word that has a high frequency in anchor texts can also report a high co-occurrence with p. *tfidf* measure which is popularly used in information retrieval can be used to normalize this bias. *tfidf* is computed from Table 1 as follows,

$$tfidf(n_j) = k \log \frac{N}{K+1}.$$

From Table 1 we compute co-occurrence measures; log likelihood ratio **LLR** (Dunning, 1993), chi-squared measure **CS**, point-wise mutual information **PMI** (Church and Hanks, 1991) and hyper geometric distribution **HG** (Hisamitsu and Niwa, 2001). Each of these measures is used to rank candidate aliases of a given name. Because of the limited availability of space, we omit the definitions of these measures.

Furthermore, we define popular set overlap measures; *cosine measure*, *overlap coefficient* and *Dice coefficient* from Table 1 as follows,

$$\text{cosine}(p, x) = \frac{k}{\sqrt{n} + \sqrt{K}},$$

$$\text{overlap}(p, x) = \frac{k}{\min(n, K)},$$

$$\text{Dice}(p, x) = \frac{2k}{n + K}.$$

3.4 Hub weighting

A frequently observed phenomenon on the Web is that many web pages with diverse topics link to so called *hubs* such as Google, Yahoo or Amazon. Because two anchor texts might link to a hub for entirely different reasons, co-occurrences coming from hubs are prone to noise. To overcome the adverse effects of a hub h when computing the ranking scores described in section 3.3, we multiply the number of co-occurrences of words linked to h by a factor $\alpha(h, p)$ where,

$$\alpha(h, p) = \frac{t}{d - 1}. \qquad (1)$$

Here, t is the number of inbound anchor texts of h that contain the real name p, d is the total number of inbound anchor texts of h. If many anchor texts that link to h contain p (i.e., larger t value) then the reliability of h as a source of information about p increases. On the other hand, if h has many inbound links (i.e., larger d value) then it is likely to be a noisy hub and gets discounted when multiplied by $\alpha(<< 1)$. Intuitively, Formula 1 boosts hubs that are likely to be containing information regarding p, while penalizing those that contain various other topics.

3.5 Training

In section 3.3 we introduced 9 ranking scores to evaluate the appropriateness of a candidate alias for a given name. Each of the scores is computed with and without weighting for hubs, resulting in $2 \times 9 = 18$ ranking scores. The ranking scores capture different statistical properties of candidates; it is not readily apparent which ranking scores best convey aliases of a name. We use real world name-alias data to learn the proper combination of the ranking scores.

We represent each candidate alias as a vector of the ranking scores. Because we use the 18 ranking scores described above, each candidate is represented by an 18-dimensional vector. Given a set of personal names and their aliases, we model the training process as a preference learning task. For each name, we impose a binary preference constraint between the correct alias and each candidate.

For example, let us assume for a name w_p we selected the four candidates a_1, a_2, a_3, a_4. Without loss of generality, let us further assume that a_1 and a_2 are the correct aliases of p. Therefore, we form four partial preferences: $a_1 \succ a_3$, $a_1 \succ a_4$, $a_2 \succ a_3$ and $a_2 \succ a_4$. Here, $x \succ y$ denotes the fact that x is preferred to y. We use ranking SVMs (Joachims, 2002) to learn a ranking function from preference constraints. Ranking SVMs attempt to minimize the number of discordant pairs during training, thereby improving average precision. The trained SVM model is used to rank a set of candidates extracted for a name. Then the highest ranking candidate is selected as the alias of the name.

4 Experiments

We crawled Japanese web sites and extracted anchor texts and urls linked by the anchor texts. A web site might use links for purely navigational purposes, which convey no semantic clue. To remove navigational links in our dataset, we prepare a list of words that are commonly used in navigational menus, such as *top, last, next, previous, links*, etc and remove anchor texts that contain those words. In addition we remove any links that point to pages within the same site. All urls with only one inbound anchor text are removed from the dataset. After the above mentioned processing, the dataset contains $24, 456, 871$ anchor texts pointing to $8, 023, 364$ urls. The average number of inbound anchor texts per url is 3.05 and its standard deviation is 54.02. We tokenize anchor texts using the Japanese morphological analyzer MeCab (Kudo et al., 2004) and select nouns as nodes in the co-occurrence graph.

For training and evaluation purposes we manually assigned aliases for 441 Japanese celebrities. The name-alias dataset covers people from various fields

Table 2: Mean Reciprocal Rank

Method	MRR	Method	MRR
SVM (RBF)	0.5625	lf	0.0839
SVM (Linear)	0.5186	cosine	0.0761
SVM (Quad)	0.4898	tfidf	0.0757
SVM (Cubic)	0.4087	Dice	0.0751
tfidf(h)	0.3957	overlap(h)	0.0750
LLR(h)	0.3879	PMI(h)	0.0624
cosine(h)	0.3701	LLR	0.0604
lf(h)	0.3677	HG	0.0399
HG(h)	0.3297	CS	0.0079
Dice(h)	0.2905	PMI	0.0072
CS(h)	0.1186	overlap	0.0056

of cinema, sports, politics and mass-media. The majority of people in the dataset have only one alias assigned. For each real name in the dataset we extract a set of candidates using the proposed method. We then sort the real names in the dataset according to the number of candidates extracted for them. We select the top 50 real names with the greatest number of candidate aliases for evaluation purposes because recognizing the correct alias from numerous candidates is a more challenging task that enables us to perform a strict evaluation. On average a name in our evaluation dataset has 6500 candidates, of which only one is correct. The rest of the 391 ($441 - 50$) names are used for training.

We compare the proposed method (SVM) against various baseline ranking scores using mean reciprocal rank (MRR) (Baeza-Yates and Ribeiro-Neto, 1999). The MRR is defined as follows;

$$\mathrm{MRR} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{R_i}. \tag{2}$$

Therein, R_i is the rank assigned to a correct alias and n is the total number of aliases. The MRR is widely used in information retrieval to evaluate the ranking of search results. Formula 2 gives high MRR to ranking scores which assign higher ranks to correct aliases.

Our experimental results are summarized in Table 2. The hub weighted versions of ranking scores are denoted by (h). We trained rank SVMs with linear *SVM (Linear)*, quadratic *SVM (Quad)*, cubic *SVM (Cubic)* and radial basis functions (RBF) *SVM (RBF)* kernels. As shown in Table 2, the proposed SVM-based method has the highest MRR values among all methods compared. The best results are obtained with the RBF kernel (SVM RBF). In fact for 21 out of 50 names in our dataset, SVM (RBF) correctly ranks their aliases at the first rank. Considering the fact that each name has more than 6000 candidate aliases, this is a marked improvement over the baselines. It is noteworthy in Table 2 that the hub-weighted versions of ranking scores outperform the corresponding non-weighted version. This justifies the hub weighting method proposed in section 3.4. The hub-weighted tfidf score (tfidf(h)) has the best MRR among the baseline ranking scores. For polynomial kernels, we observe a drop of precision concomitant with the complexity of the kernel, which occurs as a result of over-fitting.

Table 3 shows the top-three ranked aliases extracted for *Hideki Matsui* by various methods. English translation of words are given within brackets. The correct alias, *Godzilla*, is ranked first by SVM (RBF). Moreover, the correct alias is followed by the last name *Matsui* and his team, *New York Yankees*. In fact, tfidf(h), LLR(h) and lf(h) all have the exact ranking for the top three candidates. *Hide*, which is an abbreviated form of *Hideki*, is ranked second by these measures. However, none contains the alias *Godzilla* among the top three candidates. The non-hub weighted measures tend to include general terms such as *Tokyo*, *Yomiuri* (a popular Japanese newspaper), *Nikkei* (a Japanese business newspaper), and *Tokyo stock exchange*. A close analysis revealed that such general terms frequently co-occur with a name in hubs. Without adjusting the co-occurrences coming from hubs, such terms invariably receive high ranking scores, as shown in Table 3.

Incorrect tokenization of Japanese names is a main source of error. Many aliases are out-of-dictionary (*unknown*) words, which are known to produce incorrect tokenizations in Japanese morphological analyzers. Moreover, a name and its aliases can be written in various scripts: Hiragana, Katanaka, Kanji, Roman and even combinations of multiple scripts. Some foreign names such as *David* even have orthographic variants in Japanese: *da-bid-do* or *de-bid-do*. Failing to recognize the different ways in which a name can be written engenders wrong preference constraints during training.

Table 3: Top ranking candidate aliases for Hideki Matsui

Method	First	Second	Third
SVM (RBF)	ゴジラ (Godzilla)	松井 (Matsui)	ヤンキース (Yankees)
tfidf(h)	松井 (Matsui)	秀 (Hide)	ヤンキース (Yankees)
LLR(h)	松井 (Matsui)	秀 (Hide)	ヤンキース (Yankees)
cosine(h)	松井 (Matsui)	ヤンキース (Yankees)	秀 (Hide)
lf(h)	松井 (Matsui)	秀 (Hide)	ヤンキース (Yankees)
HG(h)	松井 (Matsui)	ヤンキース (Yankees)	秀 (Hide)
Dice(h)	松井 (Matsui)	ヤンキース (Yankees)	秀 (Hide)
CS(h)	松井 (Matsui)	メジャーリーグ (Major league)	プレーヤー (player)
lf	東京 (Tokyo)	読売 (Yomiuri)	日経 (Nikkei)
cosine	読売 (Yomiuri)	東証 (Tokyo stock exchange)	松井 (Matsui)
tfidf	読売 (Yomiuri)	東京 (Tokyo)	東証 (Tokyo stock exchange)
Dice	読売 (Yomiuri)	東証 (Tokyo stock exchange)	松井 (Matsui)
overlap(h)	プレー (play)	ゴジラ (Godzilla)	スタインブレナー (Steinbrenner)
PMI(h)	プレー (play)	ゴジラ (Godzilla)	スタインブレナー (Steinbrenner)
LLR	読売 (Yomiuri)	東証 (Tokyo stock exchange)	時事通信社 (jiji.com)
HG	読売 (Yomiuri)	東証 (Tokyo stock exchange)	松井 (Matsui)
CS	時事通信社 (jiji.com)	東証 (Tokyo stock exchange)	読売 (Yomiuri)
PMI	コムタチン (Komdatzien)	写真 (picture)	コンテンツ (contents)
overlap	コムタチン (Komdatzien)	写真 (picture)	コンテンツ (contents)

5 Conclusion

We proposed a method to extract aliases of a given name using anchor texts and link structure. We created a co-occurrence graph to represent words in anchor texts and modeled the problem of alias extraction as a one of ranking nodes in this graph with respect to a given name. In future, we intend to apply the proposed method to extract aliases for other entity types such as products, organizations and locations.

References

R.A. Baeza-Yates and B.A. Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley.

R. Bekkerman and A. McCallum. 2005. Disambiguating web appearances of people in a social network. In *Proc. of the World Wide Web Conference (WWW' 05)*, pages 463–470.

S. Chakrabarti. 2003. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann.

Z. Chen, S. Liu, L. Wenyin, Ge. Pu, and W. Ma. 2003. Building a web thesaurus from web link structure. In *Proc. of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 48–55.

K. Church and P. Hanks. 1991. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16:22–29.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74.

R. Guha and A. Garg. 2004. Disambiguating people in search. In *Stanford University*.

T. Hisamitsu and Y. Niwa. 2001. Topic-word selection based on combinatorial probability. In *Proc. of NLPRS'01*, pages 289–296.

T. Hokama and H. Kitagawa. 2006. Extracting mnemonic names of people from the web. In *Proc. of 9th International Conference on Asian Digital Libraries (ICADL'06)*, pages 121–130.

T. Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. of the ACM conference on Knowledge Discovery and Data Minning (KDD)*.

T. Kudo, K. Yamamoto, and Y. Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proc. of EMNLP'04*.

W. Lu, L. Chien, and H. Lee. 2004. Anchor text mining for translation of web queries: A transitive translation approach. *ACM Transactions on Information Systems*, 22(2):242–269.

G. Salton and M.J. McGill. 1986. *Introduction to Modern Information Retreival*. McGraw-Hill Inc., New York, NY.

P. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proc. of the ACL*, pages 417–424.