3G1-5

Extracting Key Phrases to Disambiguate Personal Names on the Web

Danushka Bollegala[*1] Yutaka Matsuo[*2] Mitsuru Ishizuka[*1]

[*1]University of Tokyo [*2]AIST

We propose an unsupervised algorithm that extracts key phrases to disambiguate namesakes on the Web. We represent each namesake by a *term-entity* model and cluster web pages using a contextual similarity metric. We evaluate the algorithm on a dataset of ambiguous names. Our method achieves over 80% accuracy and significantly reduces the ambiguity in a web search task.

1. Introduction

The Internet has grown into a collection of billions of web pages. One of the most important interfaces to this vast information are web search engines. We send simple text queries to search engines and retrieve web pages. However, due to the ambiguities in the queries, a search engine may return a lot of irrelevant pages. In the case of personal name queries, we may receive web pages for other people with the same name (*namesakes*). For example, if we search *Google* [*1] for *Jim Clark*, even among the top 100 results we find at least eight different *Jim Clarks*. The two popular namesakes; *Jim Clark* the Formula one world champion (46 pages), and *Jim Clark* the founder of Netscape (26 pages), cover the majority of the pages. What if we are interested only in the Formula one world champion and want to filter out the pages for other *Jim Clarks*? A quick solution is to modify our query by including a phrase such as *Formula one* or *racing driver* with the name, *Jim Clark*.

This paper presents an automatic method to extract such phrases from the Web. We follow a three-stage approach. In the first stage we represent each document containing the ambiguous name by a *term-entity* model. We define a contextual similarity metric based on snippets returned by a search engine, to calculate the similarity between term-entity models. In the second stage, we cluster the documents using the similarity metric. In the final stage, we select key phrases from the clusters to uniquely identify each namesake.

2. Related Work

Person name disambiguation can be seen as a special case of word sense disambiguation (WSD) [12, 10] problem which has been studied extensively in Natural Language Understanding. WSD typically concentrates on disambiguating between 2-4 possible meanings of the word, all of which are a priori known. However, in person name disambiguation in Web, the number of different namesakes can be much larger and unknown. Moreover, WSD utilizes sense dictionaries such as WordNet, whereas no dictionary can provide information regarding different namesakes for a particular name.

Research on multi-document person name resolution [1, 9, 4] focuses on the related problem of determining if two instances with the same name and from different documents refer to the same individual. Bagga and Baldwin [1] first perform within-document coreference resolution to form coreference chains for each entity in each document. They then use the text surrounding each reference chain to create summaries about each entity in each document. These summaries are then converted to a bag of words feature vector and are clustered using standard vector space model often employed in IR. The use of simplistic bag of words clustering is an inherently limiting aspect of their methodology. On the other hand, Mann and Yarowsky [9] proposes a richer document representation involving automatically extracted features. However, their clustering technique can be basically used only for separating two people with the same name. Fleischman and Hovy [4] constructs a maximum entropy classifier to learn distances between documents that are then clustered. Their method requires a large training set.

Li et al. [8] propose two approaches to disambiguate entities in a set of documents: a supervisedly trained pairwise classifier and an unsupervised generative model. However, they do not evaluate the effectiveness of their method in Web search.

Bekkerman and McCallum [2] present two unsupervised methods for finding web pages referring to a particular person: one based on link structure and another using Agglomerative/Conglomerative Double Clustering (A/CDC). Their scenario focuses on simultaneously disambiguating an existing social network of people, who are closely related. Therefore, their method cannot be applied to disambiguate an individual who's social network (for example, friends, colleagues) is not known. Guha and Grag [6] present a re-ranking algorithm to disambiguate people. The algorithm requires a user to select one of the returned pages as a starting point. Then, through comparing the person descriptions, the algorithm re-ranks the entire search results in such a way that pages referring to the same person described in the user-selected page are ranked higher. A user needs to browse the documents in order to find which

連絡先: danushka@mi.ci.i.u-tokyo.ac.jp

[*1] www.google.com

表 1: Data set for experiments

Collection	No of namesakes
person-X	4
Michael Jackson	3
Jim Clark	8
William Cohen	10

matches the user's intended referent, which puts an extra burden on the user.

None of the above mentioned works attempt to extract key phrases to disambiguate person name queries, a contrasting feature in our work.

3. Data Set

We select three ambiguous names (*Micheal Jackson*, *William Cohen* and *Jim Clark*) that appear in previous work in name resolution. For each name we query *Google* with the name and download top 100 pages. We manually classify each page according to the namesakes discussed in the page. We ignore pages which we could not decide the namesake from the content. We also remove pages with images that do not contain any text. No pages were found where more than one namesakes of a name appear. For automated pseudo-name evaluation purposes, we select four names (*Bill Clinton*, *Bill Gates*, *Tom Cruise* and *Tiger Woods*) for conflation, who we presumed had one vastly predominant sense. We download 100 pages from Google for each person. We replace the name of the person by "person-X" in the collection, thereby introducing ambiguity. The structure of our dataset is shown in Table 1.

4. Method

4.1 Problem Statement

Given a collection of documents relevant to an ambiguous name, we assume that each document in the collection contains exactly one namesake of the ambiguous name. This is a fair assumption considering the fact that although namesakes share a common name, they specializes in different fields and have different Web appearances. Moreover, the one-to-one association between documents and people formed by this assumption, lets us model the person name disambiguation problem as a one of hard-clustering of documents.

4.2 Term-Entity Model

The first step toward disambiguating a personal name is to identify the discriminating features of one person from another. In this paper we propose *Term-Entity models* to represent a person in a document.

Definition. *A term-entity model $T(A)$, representing a person A in a document D, is a boolean expression of n literals $a_1, a_2, \ldots, a_n$. Here, a boolean literal a_i is a multi-word term or a named entity extracted from the document D.*

For simplicity, we only consider boolean expressions that combine the literals through AND operator. For automatic multi-word term extraction, we use the *C-value* metric proposed by Frantzi et al. [5]. To extract entities for the term-entity model, the documents were annotated by a named entity tagger [*2]. We select personal names, organization names and location names to be included in the term-entity model.

4.3 Contextual Similarity

Sahami et al. [11] proposed the use of snippets returned by a Web search engine to calculate the semantic similarity between words. A snippet is a brief text extracted from a document around the query term. Many search engines provide snippets alongside with the link to the original document. Since snippets capture the immediate surrounding of the query term in the document, we can consider a snippet as the context of a query term. Using snippets is also efficient because we do not need to download the source document. To calculate the contextual similarity between two terms (or entities), we first collect snippets for each term (or entity) and pool the snippets into a combined "bag of words". Each collection of snippets is represented by a word vector, weighted by the normalized frequency (i.e., frequency of a word in the collection is divided by the total number of words in the collection). Then, the contextual similarity between two phrases is defined as the inner product of their snippet-word vectors. We define the similarity $\mathrm{sim}(T(A), T(B))$, between two term-entity models $T(A) = \{a_1, \ldots, a_n\}$ and $T(B) = \{b_1, \ldots, b_m\}$ of documents A and B as follows,

$$\mathrm{sim}(T(A), T(B)) = \frac{1}{n} \sum_{i=1}^{n} \max_{1 \leq j \leq m} \mathrm{ContSim}(a_i, b_j). \qquad (1)$$

Therein, $\mathrm{ContSim}(a_i, b_j)$ is the contextual similarity between terms/entities a_i and b_j. Without a loss of generality we assume $n \leq m$ in formula 1.

4.4 Clustering

We use Group-average agglomerative clustering (GAAC) [3], a hybrid of single-link and complete-link clustering, to group the documents that belong to a particular namesake. Initially, we assign a separate cluster for each of the documents in the collection. Then, GAAC in each iteration executes the merger that gives rise to the cluster Γ with the largest average correlation $C(\Gamma)$ where,

$$C(\Gamma) = \begin{cases} 1 & |\Gamma| = 1, \\ \frac{1}{2}\frac{1}{|\Gamma|(|\Gamma|-1)} \sum_{u \in \Gamma} \sum_{v \in \Gamma} \mathrm{sim}(u, v) & \text{otherwise.} \end{cases} \qquad (2)$$

Therein: $|\Gamma|$ denotes the number of documents in the merged cluster Γ; u and v are two documents in Γ and $\mathrm{sim}(u, v)$ is given by equation 1. Determining the total number of clusters is an important issue that directly affects the accuracy of disambiguation. We will discuss an automatic method to determine the number of clusters in section [4.7].

[*2] The named entity tagger was developed by the Cognitive Computation Group at UIUC. http://L2R.cs.uiuc.edu/ cogcomp/eoh/ne.html

4.5 Key phrases Selection

GAAC process yields a set of clusters representing each of the different namesakes of the ambiguous name. To select key phrases that uniquely identify each namesake, we first pool all the terms and entities in term-entity models in each cluster. In each cluster, its pool of terms and entities can be considered as defining the namesake represented by that cluster. From each cluster we select the most discriminative terms/entities as the key phrases that uniquely identify the namesake represented by that cluster from the other namesakes. We achieve this in two steps. In the first step, we reduce the number of terms/entities in each cluster by removing terms/entities that also appear in other clusters. In the second step, we find the terms/entities in each cluster which are most relevant to the name. We compute the contextual similarity between the ambiguous name and each term/entity and select the terms/entities that have the maximum similarity.

4.6 Evaluation Metric

We evaluate experimental results based on the confusion matrix, where $A[i.j]$ represents the number of documents of ”person i” predicted as ”person j” in matrix A. $A[i,i]$ represents the number of correctly predicted documents for ”person i”. We define the disambiguation accuracy as the sum of diagonal elements divided by the sum of all elements in the matrix.

4.7 Cluster Quality

Each cluster formed by the GAAC process is supposed to be representing a different namesake. Ideally, the number of clusters formed should be equal to the number of different namesakes for the ambiguous name. However, in reality it is impossible to exactly know the number of namesakes that appear on the Web for a particular name. Moreover, the distribution of pages among namesakes is not even. For example, in the ”Jim Clark” dataset 78% of documents belong to the two famous namesakes (*CEO Nestscape* and *Formula one world champion*). The rest of the documents are distributed among the other six namesakes. If these outliers get attached to the otherwise pure clusters, both disambiguation accuracy and key phrase selection deteriorate. Therefore, we monitor the *quality* of clustering and terminate further agglomeration when the cluster quality drops below a pre-set threshold. Numerous metrics have been proposed for evaluating quality of clustering [7]. We use normalized cuts [13] as a measure of cluster-quality.

Let, V denote the set of documents for a name. Consider, $A \subseteq V$ to be a cluster of documents taken from V. For two documents x,y in V, $\mathrm{sim}(x,y)$ represents the contextual similarity between the documents (Formula 1). Then, the normalized cut $N_{cut}(A)$ of cluster A is defined as,

$$N_{cut}(A) = \frac{\sum_{x \in A\ y \in (V-A)} \mathrm{sim}(x,y)}{\sum_{x \in A\ y \in V} \mathrm{sim}(x,y)}. \qquad (3)$$

For a set, $\{A_1,\ldots,A_n\}$ of non-overlapping n clusters A_i, we define the *quality* of clustering, $\mathrm{Quality}(\{A_1,\ldots,A_n\})$, as follows,

$$\mathrm{Quality}(\{A_1,\ldots,A_n\}) = \frac{1}{n}\sum_{i=1}^{n} N_{cut}(A_i). \qquad (4)$$

表 2: Disambiguation accuracy for each collection.

Collection	Majority Sense	Proposed Method	Found Correct
person-X	0.3676	0.7794	4/4
Michael Jackson	0.6470	0.9706	2/3
Jim Clark	0.4407	0.7627	3/8
William Cohen	0.7614	0.8068	3/10

To explore the faithfulness of cluster quality in approximating accuracy, we compare accuracy (calculated using human-annotated data) and cluster quality (automatically calculated using Formula 4) for person-X data set. We observe a high correlation (Pearson coefficient of 0.865) between these two measures, which enables us to guide the clustering process through cluster quality.

When cluster quality drops below a pre-defined threshold, we terminate further clustering. We assign the remaining documents to the already formed clusters based on the correlation (Formula 2) between the document and the cluster. To determine the threshold of cluster quality, we use person-X collection as training data. We select threshold at 0.935 where accuracy maximizes. Threshold was fixed at 0.935 for rest of the experiments.

5. Results

5.1 Disambiguation Accuracy

Table 2 summarizes the experimental results. The baseline, majority sense , assigns all the documents in a collection to the person that have most documents in the collection. Proposed method outperforms the baseline in all data sets. Moreover, the accuracy values for the proposed method in Table 2 are statistically significant (t-test: P(T≤t)=0.0087, $\alpha = 0.05$) compared to the baseline. To identify each cluster with a namesake, we chose the person thate, we chose the person that has most number of documents in that cluster. ”Found” column shows the number of correctly identified namesakes as a fraction of total namesakes. Although the proposed method correctly identifies the popular namesakes, it fails to identify the namesakes who have just one or two documents in the collection.

5.2 Web Search Task

Key phrases extracted by the proposed method are listed in Figure 1 (Due to space limitations, we show only the top ranking key phrases for two collections). To evaluate key phrases in disambiguating namesakes, we set up a web search experiment as follows. We search for the ambiguous name and the key phrase (for example, *”Jim Clark”* AND *”driver”*) and classify the top 100 results according to their relevance to each namesake. Results of our experiment on *Jim Clark* dataset for the top ranking key phrases are shown in Table 3.

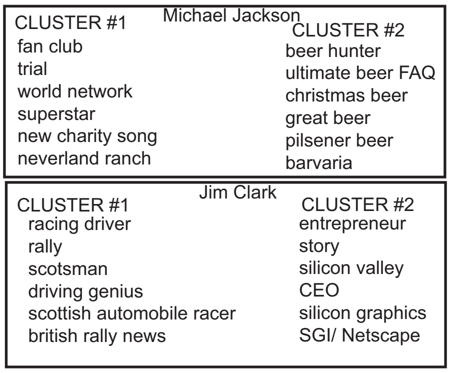

図 1: Top ranking key phrases in clusters for *Michael Jackson* and *Jim Clark* datasets.

表 3: Effectiveness of key phrases in disambiguating namesakes.

Phrase	person-1	person-2	others	Hits
NONE	41	26	33	1,080,000
racing driver	81	1	18	22,500
rally	42	0	58	82,200
scotsman	67	0	33	16,500
entrepreneur	1	74	25	28,000
story	17	53	30	186,000
silicon valley	0	81	19	46,800

In Table 3 we classified Google search results into three categories. "person-1" is the formula one racing world champion, "person -2" is the founder of Netscape and "other" category contains rest of the pages that we could not classify to previous two groups [*3]. We first searched Google without adding any key phrases to the name. Including terms *racing diver*, *rally* and *scotsman*, which were the top ranking terms for *Jim Clark* the formula one champion, yields no results for the other popular namesake. Likewise, the key words *entrepreneur* and *silicon valley* yield results fort he founder of Netscape. However, the key word *story* appears for both namesakes. A close investigation revealed that, the key word *story* is extracted from the title of the book "The New New Thing: A Silicon Valley Story", a book on the founder of Netscape.

6. Conclusion

We proposed and evaluated a key phrase extraction algorithm to disambiguate people with the same name on the Web. Our experiments with pseudo and naturally ambiguous names show a statistically significant improvement over the baseline method. The web search tasks reveals that including the key phrases in the query considerably reduces disambiguity. In future, we plan to extend the proposed method to disambiguate other types of entities such as location names, product names and organization names.

[*3] some of these pages were on other namesakes and some were not sufficiently detailed to properly classify

参考文献

[1] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of COLING*, pages 79–85, 1998.

[2] Ron Bekkerman and Andrew McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470, 2005.

[3] Douglass R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings SIGIR '92*, pages 318–329, 1992.

[4] M.B. Fleischman and E. Hovy. Multi-document person name resolution. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, 2004.

[5] K.T. Frantzi and S. Ananiadou. The c-value/nc-value domain independent method for multi-word term extraction. *Journal of Natural Language Processing*, 6(3):145–179, 1999.

[6] R. Guha and A. Garg. Disambiguating people in search. In *Stanford University*, 2004.

[7] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad, and spectral. In *Proceedings of the 41st Annual Symposium on the Foundation of Computer Science*, pages 367–380, 2000.

[8] Xin Li, Paul Morie, and Dan Roth. Semantic integration in text, from ambiguous names to identifiable entities. *AI Magazine, American Association for Artificial Intelligence*, Spring:45–58, 2005.

[9] Gideon S. Mann and David Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of CoNLL-2003*, pages 33–40, 2003.

[10] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 279–286, 2004.

[11] Mehran Sahami and Tim Heilman. A web-based kernel function for matching short text snippets. In *International Workshop located at the 22nd International Conference on Machine Learning (ICML 2005)*, 2005.

[12] Hinrich Schutze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[13] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.