Person Name Disambiguation on the Web by Two-Stage Clustering

Masaki Ikeda
University of Tokyo
ikeda@r.dl.itc.u-tokyo.ac.jp

Shingo Ono
University of Tokyo
ono@r.dl.itc.u-tokyo.ac.jp

Issei Sato
University of Tokyo
sato@r.dl.itc.u-tokyo.ac.jp

Minoru Yoshida
University of Tokyo
mino@r.dl.itc.u-tokyo.ac.jp

Hiroshi Nakagawa
University of Tokyo
nakagawa@dl.itc.u-tokyo.ac.jp

ABSTRACT

The more important web searching becomes, the more we have to focus on the "same name" problem in web searches. In this paper, we report our algorithm for disambiguating person names in web search results. It is a document clustering algorithm based on hierarchical agglomerative clustering using named entities, compound keywords, and URLs as features for document similarity calculation. We propose a two-stage clustering algorithm to improve the low recall values, in which the clustering results of the first stage are used to extract features used in the second stage clustering. We participated in the WePS-2 evaluation with this algorithm. We explain the results and describe other experiments performed with the WePS-1 data sets.

General Terms

Algorithms, Experimentation

Keywords

Name disambiguation, document clustering, two-stage clustering

1. INTRODUCTION

World Wide Web (WWW) search engines have become widely used in recent years to retrieve information about real-world entities such as people. In such cases, users enter the name of the target entity in search engines to obtain a set of Web pages that contain the name. However, the ambiguity of names (i.e., many entities have the same name) typically causes the search results to contain Web pages about several different entities.

For example, if we want to know about one "George W. Bush" who is not the former U.S. president, many pages about the former president are returned as the search results, which may require us search for the target by eye. Depending on the circumstances, we might have to search again to find Web pages about the target person among the numerous results, which is an arduous and time-consuming task.

To solve this problem, we propose a method of clustering Web pages (documents) returned by search engines in response to name queries.

Hereinafter, we use the term "person name" to mean a string indicating the name of a person. Many studies have recently been carried out on disambiguating people's names, as was done in the recent successful WePS workshop [2].

In this task, the typical approach is to define similarities between documents based on some features extracted from the documents. The feature set we used was: named entities (NEs), compound keywords (CKWs), and uniform resource locators (URLs). In terms of performance, NEs have been reported to be the most effective features for these tasks [8], as substantiated by the fact that most top-ranked systems at WePS used them as main features. NEs are good features for distinguishing among persons because they concisely represent real-world concepts related to the persons. We also used CKWs and URLs as additional features. These have similar properties to NEs (i.e., they are not observed frequently, but work as strong evidence for entity identification). The problem we experienced in using such features is that they show high precision values but low recall values. The typical approach for improving the recall is to reduce the threshold for document similarity, which means merging not-so-similar documents, and the end result is typically a drop in precision.

In this paper, we explain a new approach for improving the recall values of NE-based approaches by using a two-stage clustering algorithm. We used this method in the WePS-2 evaluation [3]. Our idea is to refine the features themselves appropriately by using the clustered documents, which are coherent thanks to the high precision, to improve recall with little sacrifice of precision. The idea behind the algorithm is that more documents provide more reliable features. In the first stage, some reliable features (like NEs) are used to connect documents about the same person. After that, the connected documents (i.e., document cluster) are used as a source from which new features are extracted. These extracted new features are used in the second stage to make additional connections between documents. Therefore, our approach is to improve clusters step by step, where each step refines clusters conservatively. Our two-stage clustering algorithm is thus a conservative algorithm in the sense that it uses only high confidence features for clustering.

We also improved the first-step clustering algorithm in comparison with our previous work [15]. First, we investigated the window size when we extracted features from document. We optimized the window size for WePS-1 and used that size in the WePS-2 competition. Secondly, we changed the calculation of similarity by merging several similarities instead of merging clusters generated by individual features [15]. The similarities of several features were calculated with an overlap coefficient. In order to merge several similarities, we got the maximal similarity and regarded the maximal value as the document similarity. Thirdly, the group-average method replaced the single-linked method used in the previous hierarchical agglomerative clustering. It outputs better results than the single-linked method.

The remainder of this paper is organized as follows. Section 2 introduces related research. Section 3 reviews feature extraction. Section 4 explains our two-stage clustering algorithm. Section 5 evaluates our proposed methods using WePS data sets. Section 6 concludes the paper by briefly summarizing the main points and mentioning future work.

2. RELATED WORK

Several important studies have tried to solve the task described in the previous section. Bagga and Baldwin [4] applied the vector space model to the calculation of similarities between names using only co-occurring words. Based on this, Niu et al. [14] presented an algorithm that uses information extraction results in addition to co-occurring words. However, these methods were tested on only small artificial test data sets, which raises doubts as to their suitability in practical use. Mann and Yarowsky [11] used a clustering algorithm to generate person clusters based on extracted biographic data. However, this method was also tested on only artificial test data. Wan et al. [18] proposed a system that rebuilt search results for person names. Their system, called Web-Hawk, was aimed at practical use like ours, but theirs was specialized for English person-name queries consisting of three words: family name, first name and middle name. They mainly assumed queries such as "⟨first name⟩" or "⟨first name⟩ ⟨family name⟩", and took middle names into consideration, which may have improved the accuracy. However, it would not be suitable for other types of names such as those in Japanese (consisting of only family and given names.)

In another approach to this task, Bekkerman and McCallum [5] proposed two methods of finding Web pages that refer to a particular person. They used two distinct mechanisms: one was based on a link structure and the other used agglomerative/conglomerative double clustering. However, they focused on disambiguating an existing social network of people, which is not the case when searching for people in real situations. In addition, as our experience is that pages containing the same name have fewer direct links than expected, it would be difficult to use information about link structures to resolve our task. Although there might be indirect links (i.e., one page can be found from another page via other pages), it would be far too time consuming to find them.

The method proposed by Bollegala et al. [6] is similar to ours in that they used extracted keywords to calculate document similarities. They further extracted keywords from resulting clusters. However, their research was aimed at keyword extraction itself, while our method uses keyword extracted from resulting clusters to improve the performance of clustering itself.

Bunescu et al. [7] reported using Wikipedia knowledge to disambiguate NEs. They used Wikipedia to extract features for supervised learning, which is different from our approach of extracting keywords from Wikipedia articles to use them for unsupervised clustering.

Our approach is based on two-stage clustering. We surveyed previous research on two-stage clustering and found some studies. First, the information bottleneck method is analogous to our method. Proposed by Tishby et al. [17], the information bottleneck method uses information theory for clustering. It has been applied to document clustering [16], where words were clustered and the word clusters were used to form document clusters. This may be a form of two-stage clustering. Secondly, Liu et al. [10] also used two-stage clustering. They identified discriminative features for clustering with voting by clusters and refined the clustering results.

3. FEATURE EXTRACTION

In this section, we explain the feature extraction and similarity calculation.

3.1 Window Size in Feature Extraction

Extracting features from a document is important in clustering for identifying persons with the same name. This problem is caused by the topic coverage of the document. The feature extraction faces the problem that a document contains not only a topic about the target person but also other topics. One attempt to deal with this issue is to limit the scope of feature extraction. However, limited features do not include all of the features available for making true clusters.

In this study, we examined how to extract features from a document. We found that extracting from the whole document improves the clustering results.

3.2 Named Entity Features

We extract NEs related to a person from documents. In this study, we treated person, location, and organization name as NEs.

NEs have been shown to be effective features in person name disambiguation, so we used NEs as features in this study. However, location and organization name have many proper nouns related weakly to a certain person. Therefore, terms having high-document-frequency in training data sets are removed from test data.

3.3 Compound Keyword Features

We also use compound nouns related to the target person. Here, we explain the extraction of CKWs from documents.

Compound nouns are extracted from a document by using morphological analysis based on the method proposed by Nakagawa et al [13]. We apply CKWs to name disambiguation [15]. The procedure is as follows.

1. We extract noun phrases $\{\mathbf{w}\}$ from the results of morphological analysis. A noun phrase $\mathbf{w}$ is compound words, where $\mathbf{w} = \{w_1, w_2, \cdots, w_L\}$.

2. The word importance $\mathrm{LR}(w)$ is calculated as Eq. (1).

$$\mathrm{LR}(w) = \sqrt{(\mathrm{LF}(w) + 1) \cdot (\mathrm{RF}(w) + 1)} \qquad (1)$$

LF(w) and RF(w) are the frequencies of nouns that directly precede or succeed simple noun w. They are smoothed by adding 1.

3. Based on word importance LR(w), the compound noun importance Score($\mathbf{w}$) is calculated as Eq. (2).

$$\mathrm{Score}(\mathbf{w}) = \mathrm{F}(\mathbf{w}) \cdot \left(\prod_{w \in \mathbf{w}} \mathrm{LR}(w) \right)^{\frac{1}{L}} \quad (2)$$

Here, F($\mathbf{w}$) is the number of independent occurrences of compound word $\mathbf{w}$ in a document, where an "independent" occurrence of $\mathbf{w}$ means that $\mathbf{w}$ is not part of any longer compound nouns, and L is the length of the noun phrase.

4. We extract $\{\mathbf{w} | \mathrm{Score}(\mathbf{w}) > \theta_{\mathrm{CKW}}, \mathbf{w} \in \{\mathbf{w}\}\}$ as CKWs, where θ_{CKW} is a threshold set by learning from train data sets.

This procedure can be obtained as a "term extraction system [1]".

3.4 Link Features

We extract links to other documents contained in the document and use them as features. We extract URLs from the tag `<a href="URL">` included in the document and extract actual URLs in the document itself. We treat normalized URLs as URL features. URLs with a high frequency are also removed if they are found in the set of unnecessary-term created from training data sets in the same way as for location/organization name filtering described in section 3.2.

3.5 Overlap Coefficient

In this study, the similarities based on features described in section 3.2–3.4 were calculated using an overlap coefficient [12] defined by Eq. (3).

$$\mathrm{Overlap}(d_x, d_y) = \frac{|\mathbf{f}_x \cap \mathbf{f}_y|}{\max(\min(|\mathbf{f}_x|, |\mathbf{f}_y|), T)} \quad (3)$$

Here, $\mathbf{f}_x$ and $\mathbf{f}_y$ are sets of features included in d_x and d_y, and $|\mathbf{f}_x \cap \mathbf{f}_y|$ is the number of factors that appeared in both d_x and d_y. The denominator of Eq. (3) is limited by the minimum value T, and $T = 4$ in this paper.

3.6 Similarity of Features

- Named Entity Features

  NE similarity $\mathrm{sim}_{\mathrm{NE}}$ is defined by Eq. (4).

$$\begin{aligned}\mathrm{sim}_{\mathrm{NE}}(d_x, d_y) &= \alpha_{\mathrm{P}}\,\mathrm{sim}_{\mathrm{P}}(d_x, d_y) + \alpha_{\mathrm{L}}\,\mathrm{sim}_{\mathrm{L}}(d_x, d_y) \\ &\quad + \alpha_{\mathrm{O}}\,\mathrm{sim}_{\mathrm{O}}(d_x, d_y) \quad (4)\end{aligned}$$

  The similarities of NEs are calculated using the similarities of Person (P), Location (L), and Organization (O). Here, α_{P}, α_{L}, and α_{O} are the corresponding weights (Note: $\alpha_{\mathrm{P}} + \alpha_{\mathrm{L}} + \alpha_{\mathrm{O}} = 1$). Each similarity $\mathrm{sim}_{\mathrm{P}}, \mathrm{sim}_{\mathrm{L}}$, and $\mathrm{sim}_{\mathrm{O}}$ is calculated with the overlap coefficient. We set these values to be $\alpha_{\mathrm{P}} \gg \alpha_{\mathrm{O}} > \alpha_{\mathrm{L}}$. We used the tuned parameters $\alpha_{\mathrm{P}} = 0.78$, $\alpha_{\mathrm{O}} = 0.16$, and $\alpha_{\mathrm{L}} = 0.06$ in the WePS-2 evaluation by learning from training data sets.

[1] `http://www.r.d.itc.u-tokyo.ac.jp/~nakagawa/resource/termext/atr-e.html`

- Compound Keyword Features

  CKW similarity $\mathrm{sim}_{\mathrm{CKW}}$ is defined by Eq. (5).

$$\mathrm{sim}_{\mathrm{CKW}}(d_x, d_y) = \mathrm{Overlap}(d_x, d_y) \quad (5)$$

  We calculate the overlap coefficient with compound nouns extracted by CKW extraction.

- Link Features

  Link similarity $\mathrm{sim}_{\mathrm{URL}}$ is defined by Eq. (6).

$$\mathrm{sim}_{\mathrm{URL}}(d_x, d_y) = \begin{cases} 1 & \text{if } d_x \text{ links to } d_y \text{ or vise versa.} \\ \mathrm{Overlap}(d_x, d_y) & \text{otherwise} \end{cases} \quad (6)$$

  If document d_x has the URL of document d_y or vice versa, URL similarity $\mathrm{sim}_{\mathrm{URL}} = 1$. Otherwise, we calculate $\mathrm{sim}_{\mathrm{URL}}$ with the overlap coefficient using URLs extracted from HTML documents as features.

3.7 Similarity Merging with Several Features

Clusters are formed by using clustering algorithms with the features mentioned above. We assume that each cluster is believed to be part of the true cluster.

To create a new cluster with multiple features, we treated the union of these clusters as the target cluster in our previous work [15]. In the present paper, we report a new path to multiple features. We take the highest similarity value among $\mathrm{sim}_{\mathrm{NE}}$, $\mathrm{sim}_{\mathrm{CKW}}$ and $\mathrm{sim}_{\mathrm{URL}}$ as shown in the following equation.

$$\begin{aligned}\mathrm{sim}_{\max}(d_x, d_y) &= \max(\mathrm{sim}_{\mathrm{NE}}(d_x, d_y), \\ &\quad \mathrm{sim}_{\mathrm{CKW}}(d_x, d_y), \mathrm{sim}_{\mathrm{URL}}(d_x, d_y)) \quad (7)\end{aligned}$$

For proper calculation of the similarity with Eq. (7), all the similarities must vary in the same range. Their range is actually confined within $[0, 1]$.

4. TWO-STAGE CLUSTERING

As a way to extract features for the clustering, we propose using the results of clustering, which results in our *two-stage clustering algorithm.* The first-stage clustering is agglomerative and hard clustering. The second-stage clustering enables soft clustering, which can treat duplication where a document contains references to more than one person with the same name. The output of the first-stage clustering has low recall. We raised the recall by applying second-stage clustering.

4.1 First-Stage Clustering

In the first stage, hierarchical agglomerative clustering (HAC) is used as a clustering method. This combines pairs of elements in order of their similarities. It uses a threshold for terminating the clustering. It is not necessary to determine the number of clusters beforehand.

Our previous approach was single-linked method. In that method, the similarity between cluster C_i and cluster C_j is defined by Eq. (8).

$$\mathrm{sim}_{\mathrm{single-linked}}(C_i, C_j) = \max_{d_x \in C_i, d_y \in C_j} \mathrm{sim}_{\max}(d_x, d_y) \quad (8)$$

Here, document d_x belongs to C_i and document d_y belongs to C_j, and $\mathrm{sim}_{\max}(d_x, d_y)$ is the document similarity between d_x and d_y. The way to calculate document similarity was described in section 3.6. The single-linked method can easily form large clusters, but it is sensitive to the clustering: If only one factor in a cluster is similar to an element in the target cluster, the cluster merges with this factor. This method forms chained clusters and makes wrong clusters by disturbances [9]. Therefore, in the present study, we used the group-average method for clustering. In the group-average method, the similarity between C_i and C_j is defined by Eq. (9).

$$\mathrm{sim}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{d_x \in C_i} \sum_{d_y \in C_j} \mathrm{sim}_{\max}(d_x, d_y) \quad (9)$$

HAC combines pairs of clusters that have the highest similarity $\mathrm{sim}(C_i, C_j)$ in sequence until all of the similarities are no longer over a similarity threshold θ_{HAC}. This clustering handles documents that is belong to only one cluster.

4.2 Second-Stage Clustering

We expect that extracting CKWs from resulting clusters will yield better quality than extracting them from documents, because a document cluster contains more text than an individual document, resulting in more reliable statistics on term frequencies. Therefore, we again extracted CKWs from clusters after carrying out CKW-based clustering.

The two-stage clustering algorithm is outlined in Algorithm 1. D is all documents and each document is $d_i (i = 1, \cdots, N)$. $\mathcal{C}$ is the result of first-stage clustering applied on D. Each cluster is $C_k (k = 1, \cdots, M)$. Clusters in $\mathcal{C}$ have no overlap (i.e., $\forall k, \forall m, (k \neq m), C_k \cap C_m \neq \emptyset$ is satisfied). We calculate the result of two-stage clustering $\mathcal{C}'$ based on the results of first-stage clustering $\mathcal{C}$.

- In line 3, $|C_k|$ is the number of documents contained in cluster C_k. From $\mathcal{C}$, we extract C_k which has the maximum $|C_k|$ in $\mathcal{C}$.

- In line 5, we treat cluster C_k as a single document and extract CKWs from it. We extract the top m keywords as $\{t_j\}$. This ranking follows the method [13] described in section 3.3.

- Lines 7–8 check whether the document d_i is included in another cluster containing t_j. If d_i has at least one t_j, we add d_i to C_k.

- In line 9, we remove cluster C_l from $\mathcal{C}$ if the cluster to which d_i belongs is $C_l = \{d_i\}$. We leave C_l if $C_l \neq \{d_i\}$ and $d_i \in C_l \wedge d_i \in C_k$. This process allows the duplication of a cluster element that is not allowed in the first-stage clustering.

- In lines 15–17, we remove C_k from $\mathcal{C}$ and add C_k to $\mathcal{C}'$. Moreover, we remove documents that contain C_k from D. We repeat this process until $\mathcal{C} = \emptyset$ is satisfied.

If $|C_k| < \theta_s$, where θ_s is a threshold, this cluster is ignored. In the study, we set the number of extracted keyword as $m = |C_k|$.

To perform hard clustering in the second-stage clustering, we need to replace lines 7–12 of Algorithm 1 with Algorithm 2.

5. EXPERIMENT

Algorithm 1 Soft clustering by two-stage clustering

```
 1: $\mathcal{C}' \leftarrow \emptyset$
 2: while $D \neq \emptyset$ do
 3:   Take Cluster $C_k$ from $\mathcal{C}$ with maximum $|C_k|$
 4:   if $|C_k| > \theta_s \quad (k = 1, \cdots, M)$ then
 5:     $\{t_j\} \leftarrow$ term_extraction($C_k$)
 6:     for $d_i \in D \backslash C_k$ do
 7:       if $d_i$ has $t_j \quad (d_i \in C_l)$ then
 8:         $C_k \leftarrow C_k \cup \{d_i\}$
 9:         if $|C_l| = 1$ then
10:           $C_l \leftarrow C_l \backslash \{d_i\}$
11:         end if
12:       end if
13:     end for
14:   end if
15:   $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{C_k\}$
16:   $\mathcal{C} \leftarrow \mathcal{C} \backslash \{C_k\}$
17:   $D \leftarrow D \backslash C_k$
18: end while
19: return $\mathcal{C}'$
```

Algorithm 2 Hard Clustering by Two-Stage Clustering

```
1: if $d_i$ has $t_j \quad (d_i \in C_l)$ then
2:   $C_k \leftarrow C_k \cup \{d_i\}$
3:   $C_l \leftarrow C_l \backslash \{d_i\}$
4: end if
```

We tested our algorithm on WePS-1 data sets [2] and WePS-2 data sets [3]. Data sets were measured [1] with both P-IP and BEP-BER. $\mathrm{F_P}$ is F-measure (P-IP) and $\mathrm{F_B}$ is F-measure (BEP-BER).

The experimental method is explained below.

1. We converted HTML files to TEXT files (one statement in one line) by applying the Automatic English Sentence Segmenter [4] and URL features. We removed unnecessary tags by lxml [5]. We checked whether the target person name was contained in documents with patterns: "⟨*first name*⟩ ⟨*family name*⟩", "⟨*first name*⟩ [A–Z]. ⟨*family name*⟩" and "⟨*family name*⟩, ⟨*first name*⟩".

2. Morphemes and NEs were extracted from documents. The part-of-speech (POS) tagger we used was Tree Tagger [6]. The NE recognizer was Stanford NER [7]. We extracted compound noun phrases as CKWs using the POS tagging results. We extracted URLs from HTML tags in the original HTML files and added the URL of the document itself.

3. NEs, CKWs and URLs were used as features. The similarities between documents were calculated from these features using an overlap coefficient. The maximum similarity of feature similarities was used as similarities of documents.

[2]http://nlp.uned.es/weps/weps-1-data/
[3]http://nlp.uned.es/weps/weps-2-data/
[4]http://www.answerbus.com/sentence/
[5]http://codespeak.net/lxml/
[6]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
[7]http://nlp.stanford.edu/software/CRF-NER.shtml

Table 1: Result of WePS-2 evaluation

Topic	Method	θ_{HAC}	BEP	BER	$\mathrm{F}_{\mathrm{B},0.5}$	$\mathrm{F}_{\mathrm{B},0.2}$	P	IP	$\mathrm{F}_{\mathrm{P},0.5}$	$\mathrm{F}_{\mathrm{P},0.2}$
ITC-UT_1	SOFT	0.001	0.93	0.73	0.81	0.76	0.95	0.81	0.87	0.83
ITC-UT_2	SOFT	0.005	0.94	0.72	0.81	0.75	0.96	0.80	0.87	0.83
ITC-UT_3	HARD	0.001	0.95	0.70	0.79	0.73	0.96	0.79	0.86	0.81
ITC-UT_4	HARD	0.005	0.96	0.65	0.76	0.69	0.97	0.74	0.83	0.78
ITC-UT_5	HARD	0.010	0.97	0.61	0.73	0.65	0.98	0.71	0.81	0.75

Table 2: NE extraction : window size in WePS-1 data sets.

Topic	BEP	BER	$\mathrm{F}_{\mathrm{B},0.5}$	P	IP	$\mathrm{F}_{\mathrm{P},0.5}$
NE, 50	0.96	0.50	0.64	0.90	0.62	0.72
NE, 100	0.95	0.54	0.67	0.89	0.65	0.74
NE, 200	0.93	0.57	0.69	0.88	0.68	0.76
NE, All	0.86	0.65	**0.73**	0.76	0.76	0.74

Table 3: CKW extraction : window size in WePS-1 data sets.

Topic	BEP	BER	$\mathrm{F}_{\mathrm{B},0.5}$	P	IP	$\mathrm{F}_{\mathrm{P},0.5}$
CKW, 50	0.96	0.41	0.56	0.90	0.53	0.65
CKW, 100	0.95	0.47	0.61	0.89	0.59	0.69
CKW, 200	0.94	0.53	0.66	0.88	0.64	0.73
CKW, All	0.82	0.64	**0.70**	0.73	0.74	0.72

4. The first-stage clustering were applied. We made clusters by HAC. The distances between clusters were calculated by the group-average method.

5. The second-stage clustering was applied to the first-stage clustering results. Important CKWs were extracted from clusters output by the first-stage clustering, and clustering was done using these extracted CKWs as features.

5.1 Results of WePS-2 Evaluation

We submitted five clustering results to the WePS-2 evaluation [3]. They covered two types of clustering method. Here, we explain the results.

The result of the WePS-2 evaluation are shown in Table 1. They were based on HAC, which used similarity $\mathrm{sim}_{\max}(d_x, d_y)$. They used two-stage clustering. The second-stage clustering methods were HARD clustering and SOFT clustering. Here, θ_{HAC} is the HAC threshold used in the first-stage clustering. We set θ_{HAC} in the WePS-2 evaluation by learning from traing data sets.

5.2 Experiment : Window Size for Extracted Features

We tested several window sizes around the query person name from which we extracted NEs and compound noun phrases from documents. We compared the clustering performance versus window size for window sizes of 50, 100, and 200 words and for the whole document.

The results for various window size in NEs extraction are presented in Table 2. The results of changing the window size in CKW extraction are presented in Table 3. NE and CKW mean NE features and CKW features. In the left-most column, 50, 100, and 200 denote the window size and "All" denotes extraction from the entire document. Here, the HAC threshold was 0, which means that documents belong to the same cluster if the similarity between documents is over 0. As seen in these tables, the larger windows size was, the better F_{B} was.

Table 4: Experiments using WePS-1 data sets.

Topic	BEP	BER	$\mathrm{F}_{\mathrm{B},0.5}$	P	IP	$\mathrm{F}_{\mathrm{P},0.5}$
Baseline						
ALL IN ONE	0.18	0.98	0.25	0.29	1.00	0.40
ONE IN ONE	1.00	0.43	0.57	1.00	0.47	0.61
COMBINED	0.17	0.99	0.24	0.64	1.00	0.78
First-stage clustering						
URL	0.98	0.48	0.62	0.83	0.56	0.64
CKW	0.82	0.64	0.70	0.73	0.74	0.72
NE	0.86	0.65	0.73	0.76	0.76	0.74
MAX	0.85	0.73	0.77	0.74	0.82	0.76
Second-stage clustering						
HARD	0.84	0.76	**0.78**	0.74	0.84	0.77
SOFT	0.83	0.76	**0.78**	0.74	0.84	0.77

5.3 Experiment using by WePS-1 Data Sets

We compared our proposed method with other methods: one-stage and two-stage clustering methods.

First, let us explain one-stage clustering method. URLs, CKWs, and NEs are employed as features for clustering. The topics are listed in the left most column. URL, CKW, and NE denote one-stage clustering methods each using a single type of the features. MAX is the one-stage clustering method described in section 3.7. HAC was used as the clustering method for these. Next, we explain the two-stage clustering method. HARD and SOFT were confirmed by re-clustering of the results of MAX clustering. HARD stands for a hard clustering method; in other words, a document belongs to only one cluster. SOFT stands for a soft clustering method; in other words, a document belongs to more than one cluster.

We compared these methods with baselines, which were ALL IN ONE, ONE IN ONE, and COMBINED.

The results are shown in Table 4. MAX, HARD and SOFT showed the best performance for several thresholds. The relationship between threshold θ_{HAC} and F-measure (BEP-BER) $\mathrm{F}_{\mathrm{B},0.5}$ is shown in Fig. 1. We extracted NEs from the whole document and CKWs from 100 words around the query. We confirmed that the results were improved by

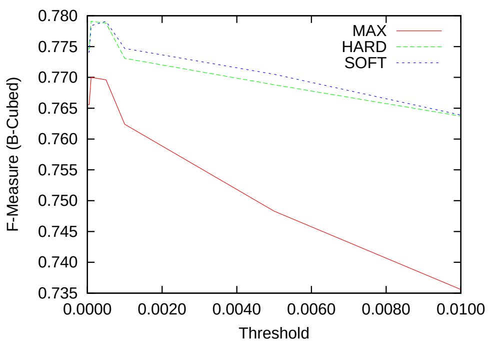

Figure 1: Relationship between threshold and F-measure.

using two-stage clustering. Here, $\theta_{\text{HAC}} = 0$. We found that two-stage clustering improves the recall without harming the precision.

The performance results (BEP-BER scores) show that our system outperformed all the other system in WePS-1 (in B-Cubed F-measure) or took second place (in P-IP F-measure). These results suggests that using two-stage clustering is a good way to achieve high performance rather than simplicity.

6. CONCLUSION

Our algorithm for person name disambiguation consists of the two-stage clustering in which the results of first-stage clustering are used to extract features for the second-stage clustering. Experiments showed that our algorithm performed better than the top systems in the WePS-1 competition. Furthermore, two-stage clustering can handle a change in the first-stage clustering algorithm, so performance can be improved by replacing the first-stage algorithm when new, better ones become available. In the future, we will investigate new types of features to improve the recall score. We will also try to improve the filtering algorithm for extracting CKWs in the second-stage clustering.

7. ACKNOWLEDGMENT

This research was funded in part by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid.

8. REFERENCES

[1] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, pages 1–26, 2008.

[2] J. Artiles, J. Gonzalo, and S. Sekine. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. *The SemEval-2007*, pages 64–69, 2007.

[3] J. Artiles, J. Gonzalo, and S. Sekine. WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task In 2nd Web People Search Evaluation Workshop (WePS 2009). In *18th WWW Conference*, 2009.

[4] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the Vector Space Model. *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 79–85, 1998.

[5] R. Bekkerman and A. McCallum. Disambiguating Web appearances of people in a social network. *Proceedings of the 14th international conference on World Wide Web*, pages 463–470, 2005.

[6] D. Bollegala, T. Honma, Y. Matsuo, and M. Ishizuka. Automatically extracting personal name aliases from the web. *GoTAL '08: Proceedings of the 6th international conference on Advances in Natural Language Processing*, pages 77–88, 2008.

[7] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3–7, 2006.

[8] E. Elmacioglu, Y. Tan, S. Yan, M. Kan, and D. Lee. PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features. *The SemEval-2007*, pages 268–271, 2007.

[9] S. Kamvar, D. Klein, and C. Manning. Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based Approach. *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 283–290, 2002.

[10] X. Liu, Y. Gong, W. Xu, and S. Zhu. Document clustering with cluster refinement and model selection capabilities. In *In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–198. ACM Press, 2002.

[11] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 33–40, 2003.

[12] C. Manning and H. Schütze. *Foundations of statistical natural language processing.* MIT Press, 1999.

[13] H. Nakagawa and T. Mori. Automatic term recognition. *Terminology*, 9(2):201–219, 2003.

[14] C. Niu, W. Li, and R. Srihari. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. *context*, 2:1, 2004.

[15] S. Ono, I. Sato, M. Yoshida, and H. Nakagawa. Person Name Disambiguation in Web Pages Using Social Network, Compound Words and Latent Topics. *LECTURE NOTES IN COMPUTER SCIENCE*, 5012:260–271, 2008.

[16] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–215, 2000.

[17] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method, 2000.

[18] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 163–170, 2005.