Computing and Informatics, Vol. 37, 2018, 1485–1515, doi: 10.4149/cai_2018_6_1485

WEB PERSON NAME DISAMBIGUATION USING SOCIAL LINKS AND ENRICHED PROFILE INFORMATION

Hojjat Emami, Hossein Shirazi

Social Network and Intelligent Systems Laboratory
Department of Information and Communication Technology (ICT)
Malek-Ashtar University of Technology
Tehran, Iran
e-mail: {h_emami, shirazi}@mut.ac.ir

Ahmad Abdollahzadeh Barforoush

Intelligent Systems Laboratory
Computer Engineering and IT Department
Amirkabir University of Technology
Tehran, Iran
e-mail: ahmad@aut.ac.ir

Abstract. In this article, we investigate the problem of cross-document person name disambiguation, which aimed at resolving ambiguities between person names and clustering web documents according to their association to different persons sharing the same name. The majority of previous work often formulated cross-document name disambiguation as a clustering problem. These methods employed various syntactic and semantic features either from the local corpus or distant knowledge bases to compute similarities between entities and group similar entities. However, these approaches show limitations regarding robustness and performance. We propose an unsupervised, graph-based name disambiguation approach to improve the performance and robustness of the state-of-the-art. Our approach exploits both local information extracted from the given corpus, and global information obtained from distant knowledge bases. We show the effectiveness of our approach by testing it on standard WePS datasets. The experimental results are encouraging and show that our proposed method outperforms several baseline methods and also its counterparts. The experiments show that our approach not only improves the performances, but also increases the robustness of name disambiguation.

Keywords: Web mining, cross-document name disambiguation, social links, profile enrichment, clustering

Mathematics Subject Classification 2010: 97R40, 97R50,68T50, 68U35, 90B40

1 INTRODUCTION

The volume of information available on the web is increasing considerably. This vast volume of information brings new challenges such as lost in hyperspace or information overload [1, 2]. These challenges make it difficult to retrieve useful information about various entities from web pages. Searching for entities, especially people, and their related information is one of the most common activities of Internet users. As personal names are highly ambiguous, personal information extraction and information retrieval systems deal with a fundamental problem, namely name ambiguity problem. The problem of name ambiguity causes the results of a personal name search to be a mix of web pages about different people sharing the same name. This issue emphasizes the necessity of developing high-quality name disambiguation systems to resolve ambiguity between people names and cluster search results according to different people having the same name. Developing such a name disambiguation system can be useful in a wide range of areas including semantic web, information extraction, question answering, machine translation, data fusion, speech recognition, and social network analysis, among others.

In recent years, several research efforts have been conducted towards the name disambiguation in web context. The problem of name disambiguation is usually formulated as a name clustering problem [3, 4, 5, 6]. Clustering-based name disambiguation approaches are well-known due to their superior feasibility and efficiency in dealing with a large amount of data. Clustering-based methods are useful when we do not have a large labeled corpus and there are varying ambiguities in the corpus. A majority of previous work [3, 4, 5, 6, 7, 8, 9, 10, 11] use a combination of various features to compute similarities between entities, and then utilize clustering algorithm to disambiguate entity names. A great deal of research exploits syntactic and semantic local features derived from the given corpus [7, 8, 11]. However, the local information may not be sufficient to resolve ambiguities and the robustness of system will be severely degraded due to

1. the substantial noise and low quality of information extraction (IE) and natural language processing (NLP) systems used for extracting local document-level features, and
2. insufficient information contained in web pages.

To alleviate these problems, authors in [3, 4, 6], besides the local information, have exploited the global information derived from extra corpora or distant knowledge bases. Nevertheless, these solutions do not completely utilize all the semantic information contained in web pages such as entity attributes and social links. In this paper, we attempt to overcome the deficiencies of previous work by proposing a person name disambiguation approach that not only uses local personal attributes and social links stored in given web pages, but also global semantic information about persons embedded in external knowledge bases. Our approach makes a full use of the merits of both attribute-based and social network based methods. Personal attributes and social links are two different sources of information that can complement each other. This leads to more precise person name disambiguation, and confirms the need of a framework for integrating social links and the enriched attributes of a person are needed.

To summarize, our contributions lie in the approaches we propose to solve subtasks of name disambiguation:

- We map all of the information about persons in text documents to an undirected weighted graph. In graph creation process, we propose new methods for each task of *social link extraction*, *profile extraction*, and *profile enrichment*. Specifically, we propose a new method for social links extraction relying on the closeness centrality theory [12]; we propose a profile enrichment method relying on the closeness centrality theory and deep semantic analysis of the text to deal with the problem of data sparseness and to make name disambiguation system more robust.
- We employ BIC-Means algorithm to cluster nodes of graph, and propose a dynamic weight learning method based on a new TF-IDF schema [45] to learn importance coefficient of attributes in computing similarity among entities.
- We perform extensive evaluation of our proposed approach over real, standard WePS datasets [13, 14]. We demonstrate that our method outperforms baseline methods and state-of-the-art counterparts. This justifies that our approach is a promising solution for the problem of person name disambiguation.

Having this short introduction, the rest of this paper is organized as follows. Section 2 is devoted to literature review and presents an overview of the related work. Section 3 introduces the working principle of our name disambiguation approach. Then, in Section 4, the proposed approach is evaluated on benchmark datasets and the results are compared to the baselines and state-of-the-art methods. Finally, Section 5 makes conclusions and discusses some future works.

2 RELATED WORK

Our work addresses the problem of cross-document person name disambiguation in web context. Let $D = \{d_1, d_2, \ldots, d_N\}$ be a collection of web documents referring to a set of persons having the same name, and let $M = \{m_{11}, m_{12}, \ldots, m_{21}, m_{22}, \ldots, \}$,

$m_{ij} \in d_i$ be a set of name observations within collection D, which need to be disambiguated. Name ambiguity can occur within a web document or across documents. In case of within-document ambiguity, name observations in M are from the same document $d_i \in D$. In case of cross-document, the mentions in M are from the entire corpus D. According to the type of ambiguity, name disambiguation systems are categorized into two classes:

1. within-document name disambiguation, which often referred to as within-document co-reference, and
2. cross-document name disambiguation.

Within a document, mentions with the same string typically refer to the same entity in reality, whereas in different documents identical entity mentions may have different meanings [4, 6]. This is important information, which shows that cross-document name disambiguation cannot be solved by applying within-document co-reference resolution to a super document formed by concatenating all documents in the corpus. In this paper, we focus on cross-document name disambiguation. In the following, we briefly present and discuss the most significant research work in the area of cross-document name disambiguation, their limitations, and compare our approach with them. This short discussion highlights the need for developing new and more efficient name disambiguation approaches.

In recent years, many research efforts have been made towards name disambiguation in relational databases and web context. There are several surveys in name disambiguation area, among which we point out Brizan and Tansel [15], Elmagarmid et al. [16], Kopcke and Rahm [17], and Ferreira and Gonçalves [18]. Entity name disambiguation in web is similar to those approaches developed in database domain. However, there are several differences [11]:

1. web documents are often unstructured while database records are structured, and
2. web documents often only contain partial or incomplete information about the entities.

Therefore, most disambiguation methods, which were developed for databases are not directly applicable on web data.

Name disambiguation is often formulated as a clustering problem [3, 4, 5, 6, 11]. Clustering-based name disambiguation approaches are well-known due to their superior efficiency in dealing with a large amount of data. They are useful when we do not have a large labelled corpus, and there are varying ambiguities in a corpus [4]. Clustering methods often include three main steps:

1. feature extraction,
2. similarity computation, and
3. name grouping.

In most of the existing approaches, employed features are either syntactic or semantic. Syntactic features include tokens [7], specific keywords [6, 11], n-gram features, snippet-based features [4], etc. Semantic features include personal attributes [19, 20, 21], hyperlinks [8, 9], named-entities [6, 11], etc. Each web document, which needs to be disambiguated, is represented as a vector of desired features. Similarities among document vectors are then computed using various similarity measures to identify whether they refer to the same entity. Similarity computation forms the basis of name disambiguation in clustering approaches. The quality of similarity computation significantly depends on the type of analyzed input data, similarity measures and features under evaluation. Many existing approaches, such as [19, 20, 21, 22, 23], compute similarity between entities by matching their corresponding feature vectors containing attributes of those entities. However, such methods ignore some important implicit semantic information, such as links between entities. Some other works have harnessed co-occurring entity mentions to compute similarities [8, 24, 25]. These approaches often create a social relationship graph of entity names (especially person names) co-occuring in a document and then partition the graph into sets of groups using graph partitioning algorithms. The idea behind social network-based methods comes from the fact that linked entities might be having the similar characteristics. The main problem of these methods is that they may fail to distinguish entities when a web document does not contain any information about people connections. A few attempts, such as [8, 26], integrate both the entities' links and attributes to resolve ambiguities. The idea behind such models is that the attributes of an entity can complement social network structure, and vice versa. In other words, if one source of information is missing or noisy, the other can make it up. These hybrid models make full use of the merits of both attribute-based and social network based name disambiguation methods. However, these methods do not employ external data sources beyond the given corpus, and use only the information contained in the given corpus being processed. Our approach extends these methods through utilizing both local persons' social links and attributes, and global semantic attributes from distant knowledge bases.

Exploiting external, global features for disambiguation was also studied in previous works [3, 4, 27, 28]. However, the context information including social links among entities and attributes has not been utilized entirely. To alleviate this shortcoming, similar to ours, Dutta and Weikum [6] exploit both the context where entities appear and the information from external knowledge bases for co-referent entities. However, limitations of their approach are that

1. co-occurrence of entities is only considered within web pages,
2. the information contained in web pages is not completely exploited, particularly information expressed in informal-style fragments, and
3. in knowledge enrichment stage, a simple string matching method is utilized for entity matching.

Our approach is robust enough and regardless of the external knowledge features, it uses all of the information contained in the given web pages expressed either in formal-style or informal-style formats. Our approach relies on deep semantic analysis of the text and closeness centrality theory to exploit social links of entities across web pages.

In summary, our work extends previous work by integrating social links with the attributes of the local semantic profile attributes and global attributes from external knowledge bases. Following this way, our approach exploits all of the information about person entities contained in the textual parts of local web pages and external knowledge bases. This leads to more robust name disambiguation approach.

3 OUR PROPOSED APPROACH

In this paper, we formulate the person name disambiguation as a clustering problem. Let $G = \{C_1, C_2, \dots, C_K\}$ be a set of K clusters, and $C_i \neq \emptyset$, $C_i \cap C_j = \emptyset$, $i \neq j$, $i, j = 1, 2, \dots, K$, $\bigcup_{j=1}^{K} C_j = D$, where $D = \{d_1, d_2, \dots, d_N\}$ is the N web pages referring to the different people sharing the same name. The goal of our name disambiguation is to find such G, where objects $\left\{u_p^i, u_{p+1}^i, \dots, u_q^i\right\}$, $(1 \leq p \leq q \leq N)$ in cluster $C_i \in G$ refer to the same entity in reality. Figure 1 shows an overview of our proposed name disambiguation approach. It consists of four main stages as follows:

Pre-processing. Pre-processing takes as input web pages and transforms them to system-desired format using existing pre-processing tools. Pre-processing consists of five main subtasks: *extracting clean text document*, *named-entity tagging*, *intra-document co-reference resolution*, *sentence splitting*, and *sentence type detection.*

Graph creation. This component takes pre-processed text of web documents and extracts person discourse profile; enriches the discourse profile with external semantic features; and extracts social links between entities from the text. It then maps the profile attributes and links into an undirected weighted graph (Attribute-Relationship Graph). This graph is an abstract and structured representation of the information constituents from the web documents and relevant information from external knowledge bases.

Similarity computation. We use a modified random walk model to compute similarities among graph's nodes. The employed similarity measure considers both people's attributes and social links.

Graph clustering. The clustering phase takes as input the attribute-relationship graph and similarity measures, and then groups graph nodes into sets of clusters, where each cluster contains all the nodes referring to a unique person.

In the following, we describe these components in more detail.

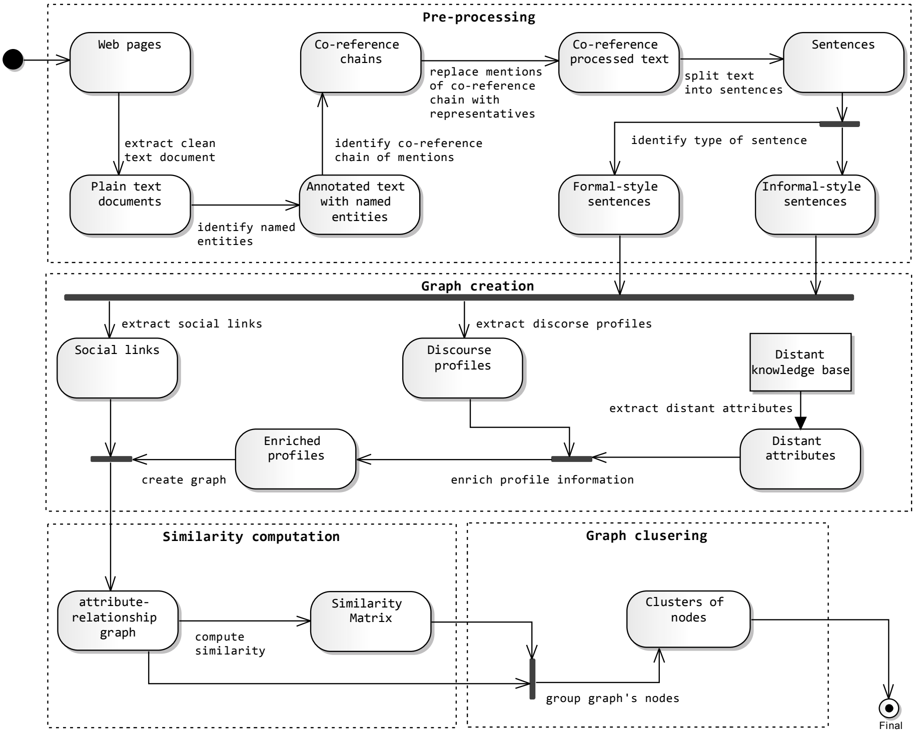


Figure 1. Outline of our name disambiguation approach

3.1 Pre-Processing

In this article, we focus on the textual part of the web pages, because the majority of the information about entities on the web is often expressed in the natural language text. The web pages need to be pre-processed and prepared according to system's desired format. First, for each web page, Jsoup[1] (an HTML parser) is run to cast it into plain text document. Next, for each document the Stanford named-entity tagger [29] is run to tag the text for coarse-grained lexical entity types including person, location, organization, etc. For each identified named-entity, we assign a unique index to distinguish the identity of entity. The annotated text documents are passed to the intra-document co-reference resolution module. We use the state-of-the-art Stanford co-reference resolution system [30] to identify co-reference chains for all the entities mentioned in each document. The mentions in every co-reference chain of interest are then replaced with their corresponding representative mentions. Next, for the co-reference chain of interest within each document, we use the Stanford

[1] Jsoup: Java HTML Parser, `http://jsoup.org/`

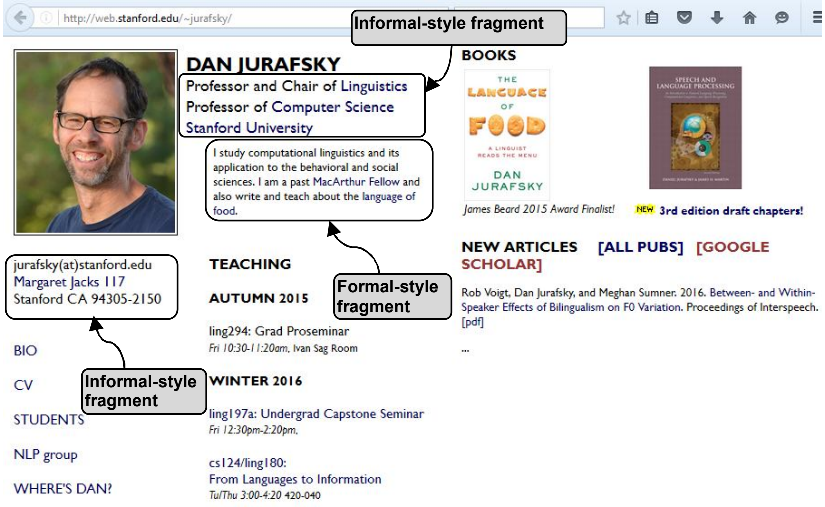
CoreNLP toolkit[2] [31] to extract all the sentences from a document. The content on a web page is often expressed in a mixture of different representations: *formal-style* format and *informal-style* format [4, 32]. A formal-style text follows prescribed writing standards, and is prepared for a fairly broad audience [4, 32]. Formal writing needs to be well-structured, clear and unambiguous. Longer and complete sentences are likely to be more prevalent in formal-style text. Complete sentences usually contain a subject, object and one or multiple verbs. On the contrary, informal-style text has few constraints on writing format, mixes various representations, is prepared quickly and intended for a narrow audience [32]. In sentence type identification, we classify each sentence in a web document as one of the two classes, formal-style or informal-style. In Figure 2, we show an excerpt of formal-style and informal-style fragments. We use support vector machines (SVMs) [33] to classify sentences into formal-style or informal-style classes. The main feature for classification is the percentage of capitalized tokens and length of the sentence. The selection of these features comes from the fact that an informal-style sentence mainly is short and contains capitalized tokens. Each of the formal-style and informal-style expression format requires different information extraction method. Identifying the type of sentence expression helps us to overcome the problem of structure variation and choose proper attribute extraction methods (Section 3.2.2) to extract entity-centric information according to data representation format.

The pre-processing tools may produce errors, which propagate to the later stages. However, improving the pre-processing components is beyond the scope of this paper. The remainder of the processing described in the following uses this pre-processed text.

3.2 Graph Creation

In graph creation, the abstract representation we wish to create is an undirected weighted graph G for each pre-processed web document. Formally, we represent this graph as $G = (VS \cup VA, ES \cup EA)$, where VS is the set of structure nodes, VA is the set of attributes nodes, ES is the set of structure edges, and EA is the set of attribute edges. For each entity e, we create a structure node and append it to structure node set VS. Similarly, for each attribute class $a \in A$, we create an attribute node v_a and append it to attribute node set VA. We create a structure edge $es \in ES$ between a pair of structure nodes u and v, if their corresponding entities co-occur in the corpus. The weight of structure edge es between a pair of structure nodes u and v indicates the strength of the relationship between corresponding entities e_i and e_j. We draw an attribute edge $ea \in EA$ between a structure node $u \in VS$ and an attribute node $v_a \in VA$, if the node u corresponding to person e takes a value on attribute class $a \in A$. The weight of attribute edge ea indicates the importance coefficient of the target attribute.

[2] `http://nlp.stanford.edu/software/corenlp.shtml`



http://web.stanford.edu/~jurafsky/

DAN JURAFSKY

Professor and Chair of Linguistics
Professor of Computer Science
Stanford University

I study computational linguistics and its application to the behavioral and social sciences. I am a past MacArthur Fellow and also write and teach about the language of food.

jurafsky(at)stanford.edu
Margaret Jacks 117
Stanford CA 94305-2150

BIO

CV

STUDENTS

NLP group

WHERE'S DAN?

TEACHING

AUTUMN 2015

ling294: Grad Proseminar
Fri 10:30-11:20am, Ivan Sag Room

WINTER 2016

ling197a: Undergrad Capstone Seminar
Fri 12:30pm-2:20pm,

cs124/ling180:
From Languages to Information
Tu/Thu 3:00-4:20 420-040

BOOKS

James Beard 2015 Award Finalist!

NEW 3rd edition draft chapters!

NEW ARTICLES [ALL PUBS] [GOOGLE SCHOLAR]

Rob Voigt, Dan Jurafsky, and Meghan Sumner. 2016. Between- and Within-Speaker Effects of Bilingualism on F0 Variation. Proceedings of Interspeech. [pdf]

...

Figure 2. Examples of formal-style and informal-style data expression format on a sample web page

The graph creation consists of three steps:

1. social link extraction,
2. profile extraction, and
3. profile enrichment.

The system first extracts the co-occurring entities in the neighbourhood of each given person. The co-occurring entities form what we call the social relationship network. In the profile extraction stage, the system extracts local attributes associated with every entity in question from given web documents, and forms entity's discourse profile. In the profile enrichment stage, the system then enriches the local discourse profiles with rich global features retrieved from external knowledge bases by considering co-occurring entities and their surrounding context. The linked (co-occurring) entities, global attributes, and local attributes associated with each person are mapped into an undirected graph. In the following, we describe the graph creation stages in more details.

3.2.1 Social Link Extraction

We assume that relationship between entities in the real world is reflected by their closeness in text of the documents they are mentioned in. We assume that two entities are linked if they collocate together in a corpus more frequently. To identify the linked entities with an entity e, we extract the co-occurring entities in the neighbourhood of entity e. To do this, we first identify sentences in which the focused entity e or its co-referent mentions occur. The sentences of containing these entities contain also other entities that can be co-occurring with other ones as well. Formally, in a document, entities in the neighbourhood of the entity e appear in the following sentences:

$$\begin{aligned} S(e) &= H^n(e), \\ H^1(e) &= \left\{ \bigcup_j s(e_j) \mid \forall \mathrm{e}_j \in M(e) \right\}, \\ H^n(e) &= H^{n-1}(e) \cup \left\{ \bigcup_k H^1(e_k) \mid \exists s_f : (e \in s_f) \wedge (e_k \in s_f) \right\}, \quad \text{for } n \geq 2 \end{aligned} \tag{1}$$

where $S(e)$ is a set of sentences in the neighbourhood of entity e, $H^n(e)$ is a set of n^{th}-hop sentences containing entities co-occurring with target entity e, $M(e)$ is an intra-document co-reference chain of entities with respect to entity e, and s_f is the f^{th} sentence in which both e and e_k co-occur together. Let $N(e) = \{e_1, e_2, \ldots, e_m\}$ be the list of co-occurring entities with target entity e, which appear in $S(e)$. We create a social relationship graph between the entities in $N(e)$. We define the strength of relationship between a pair of linked entities e_i and e_j as the normalized distance-weighted frequency of entities' co-occurrences in the input corpus as follows:

$$w_{ij} = \frac{\theta_{ij}}{\sum_k \sum_l \theta_{kl}} \tag{2}$$

where θ_{ij} is the distance-weighted frequency of entities' co-occurrences in the corpus, which we defined it as follows:

$$\theta_{ij} = \sum_{d \in D} \begin{cases} \sum_{(e_i,e_j) \in d} 1 - \left(\frac{log_2(\chi_{ij})}{2} \right), & \text{if } (\chi_{ij} < \eta), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where d is a document in the corpus D, e_i and e_j are the i^{th} and j^{th} entities in document d, respectively. χ_{ij} is the position distance between two entities with value 1 if entities e_i and e_j collocate in the same sentence, 2 in neighbouring sentence, and so on. The entities having position distance χ_{ij} above the threshold η are ignored. We empirically set η to 4, which prevents linking far entities. Obviously, the bigger the θ_{ij} is, the bigger the w_{ij} is. w_{ij} is in the range $[0, 1]$. In this paper, we use only the web pages of each ambiguous name in the given corpus as the entity co-occurrence corpus. To compute w_{ij}, one can use large-scale corpora beyond the given corpus as entity co-occurrence corpus. This is considered as one of the future work of this paper. Nonetheless, our experiments show that the given corpus is sufficient for extracting social links.

3.2.2 Profile Extraction

To extract attributes of a certain person and form their discourse profile, we have developed an integrated profile extraction system. Our profile extraction system takes as input the pre-processed text, extracts the person entities with related information and forms their discourse profile. Discourse profile of a person entity contains a set of ⟨attribute, value⟩ pairs. Formally, we define the discourse profile of entity e as follows:

$$P(e) = \{(a, v) \mid a \in A, v \in V(a)\} \tag{4}$$

where A is the vocabulary of attributes that describes characteristics of person entity e, and $V(a)$ represents the set of valid values for attribute $a \in A$. In our implementation, each person entity can take at most sixteen kinds of attributes, which include *affiliation*, *award*, *birth place*, *date of birth*, *other name*, *occupation*, *school*, *major*, *degree*, *mentor*, *nationality*, *relatives*, *phone*, *fax*, *e-mail*, and *website*. Due to limited space, in this paper, we do not give more detailed discussion of our profile extraction system. For more detail refer to our technical report given in (`http://ceit.aut.ac.ir/islab/guest/Emami/PersonProfiling_TechnicalReport.pdf`). Our profiling approach is efficient, and can extract personal attributes from both formal-style and informal-style fragments.

3.2.3 Profile Enrichment

The discourse profile of entities can be applied directly to compute similarities and resolve ambiguity. However, the sparse data contained in discourse profiles may not be sufficient to resolve ambiguities and the system robustness will be degraded due to low quality of profile extraction system. For these reasons, we propose an enrichment method of the persons' profile via global attributes extracted from external knowledge base. Profile enrichment attempts to alleviate the problem of data sparseness and improve the robustness of system. Profile enrichment includes two steps:

1. entity linking, and
2. attribute extraction.

In entity linking step, for an entity mention e, we determine its identity in text to identify the best matching entity in the external knowledge base. In attribute extraction step, we retrieve the global attributes for the target person e from external knowledge base. These attributes are beyond the discourse profile.

Our entity linking system takes as input the target entity mention e and the context $S(e)$ (Equation (1)) around it. It identifies the entity mentions in neighbourhood of entity e and forms a list of co-occurring entity mentions $N(e) = \{e_1, e_2, \ldots, e_m\}$; where each $e_i \in N(e)$ refers to a co-occurring entity mention with target entity mention e appearing in $S(e)$. Entity linking system then extracts the intra-document co-reference chain of the entity e and entities in $N(e)$. To match an entity mention e against an entity from external ontology, our entity linking system creates a phrase query comprising mentions from the co-reference chain of e

and its neighbours $N(e)$. Entity linking system feeds the query to Babelfy [34] to identify the BabelNet synset id of the target entity e, and links it into the corresponding DBPedia URI. Babelfy is a state-of-the-art word sense disambiguation and entity linking system. As a matter of fact, using Babelfy is not mandatory; any disambiguation or entity linking strategy can be used at this stage. However, a knowledge-based unified approach like Babelfy is best suited to our setting. The attribute extraction phase takes as input the DBPedia URI of the target entity e, and retrieves its attribute from DBPedia ontology [36] to enrich discourse profile of entity e. The choice of DBPedia for enrichment is not mandatory, and other knowledge bases such as Freebase and Yago can be used for enrichment purpose. However, DBPedia is best suited to our setting, because it provides high-coverage structured information about entities.

We primarily rely on the Babelfy itself to identify the correct identity of the target entity e. Babelfy may produce some noisy data because in some cases it cannot infer the correct identity of entities. Therefore, to avoid dependency on the output of the Babelfy to infer whether the retrieved external entity t best matches with the target entity e, we rank the candidate entity t by our similarity measure and prune out candidates with low confidence. In similarity computation, we first compare the type tag of the entities e and t. If the entity type tag of entities e and t are not the same, we ignore the external entity t; otherwise we compare the attributes of entity t with local attributes of the target entity e. For this purpose, we compute the normalized similarity between entities t and e based on their attributes:

$$\mathrm{Sim}(e,t) = \frac{1}{|A_{e,t}|} \times \sum_{a \in A_{e,t}} \beta_a \times M_a(e,t) \tag{5}$$

where $M_a(e,t)$ is the similarity of the two entities based on attribute a, β_a is the importance coefficient of attribute a, and $A_{e,t}$ represents all the attributes associated with both entities e and t. $A_{e,t}$ is equal to $A_{e,t} = (A_e \cup A_t)$, where A_e and A_t respectively represent the set of attributes associated with entity e and t. We define a confidence threshold T, such that a candidate entity t having the similarity value $\mathrm{Sim}(e,t)$ below the threshold T is pruned out. If $\mathrm{Sim}(e,t) \geq T$, the system then appends the attributes of entity t to discourse profile of entity e.

In general, each attribute class $a \in A$ may be one of the following types: single-value attribute or multiple-value attribute [37]. Single-value attribute (e.g. *date of birth*) can only take a single value, while multiple-value attribute (e.g. *affiliation*, *occupation*) can take one or more different values. If attribute a is a multiple-value attribute, to compute $M_a(e,t)$ we first compute single-value similarities for all the possible values of a and then aggregate the maximum single-value similarities. Therefore, we define $M_a(e,t)$ as follows:

$$M_a(e,t) = \frac{1}{\min\left(|I_e|, |I_t|\right)} \times \sum_{p \in I_e} \max\left(\delta\left(p, I_t\right)\right) \tag{6}$$

where I_e and I_t represent the item set of attribute a for person e and t, respectively. $\delta(p, I_t)$ is the set of single-value attribute similarities computed between element $p \in I_e$ and all elements in I_t. We define $\delta(p, I_t)$ as follows:

$$\delta(p, I_v) = \{\varphi(p, q) \mid q \in I_t\}. \tag{7}$$

$\varphi(p, q)$ computes the similarity between item p and q using an appropriate standard similarity measure. Personal attributes are heterogeneous; therefore it is not reasonable to use the same similarity measure for different attributes in computing $\varphi(p, q)$. This enforces us to use appropriate similarity measures for any type of the attribute. There are different standard similarity measures, each of which is appropriate for a particular attribute class. In order to determine which similarity measure is appropriate for an attribute class, we adopt the following methodology. Borrowing the idea from [35], first, we adopt five syntactic similarity measures, which include normalized Levenshtein distance (len) [38], Dice's coefficient (dic) [39], Cosine (cos) [39], Jaccard index (jac) [40] and dates' relative similarity (Spd) [48]. The reason to select these measures is that these measures are widely used in literature to calculate similarity of data objects. We then identify the appropriateness of a similarity metric for a particular class of attribute through assessing its importance on name disambiguation. To do this, we use ground truth of training data given in WePS-1 dataset [13] and WePS-2 dataset [14]. We analyse the similarity measures in turn for each person in ground truth of dataset. Each similarity measure was computed for the attributes of each pair of persons that are co-referent, i.e., they are in the same cluster and refer to the same entity in reality. A typical similarity metric is considered to be proper for a particular attribute, if it results in higher normalized similarity value for the people who are co-referent in the ground truth. The simulation results on WePS-1 training dataset are shown in Figure 3, and the results for WePS-2 training dataset are given in Figure 4.

We apply the similarity measures on single-value items of attributes. Each similarity measure has its own strategy to compute similarity value. For example, to compute similarity by Cosine measure, we first transform the single-value items to vectors of occurrences of n-grams (sequences of n characters). In this n-dimensional space, the similarity between two items is the cosine of their respective vectors. In other words, it is computed as $(V_1.V_2) / (|V_1| \times |V_2|)$, where V_1 and V_2 is the vector representation of two comparing items p and q.

For attribute *date of birth*, we first normalize the date values using Stanford SUTime library [41] and then compute the similarity. Borrowing the idea presented in [48], to compare *date of birth* values, we first convert dates into a number of days. We calculate the number of days according to the fix date 01-01-2016. Let d_1 and d_2 be the two day values that are being compared, *Spd*, the dates' relative similarity, is calculated as follows:

$$\mathrm{Spd}(d_1, d_2) = \begin{cases} 1 - \left(\frac{\mathrm{pd}(d_1,d_2)}{pd_{\max}}\right), & \text{if } (\mathrm{pd}(d_1, d_2) < pd_{\max}), \\ 0, & \text{else}, \end{cases} \tag{8}$$

where $pd_{\max}$ $(0 < pd_{\max} < 1)$ is the maximum percentage difference that is tolerated in similarity computation. In our implementations, we empirically set $pd_{\max}$ to 0.2. $\mathrm{pd}(d_1, d_2)$ is the percentage difference, which is defined as follows:

$$\mathrm{pd}(d_1, d_2) = \frac{|d_1 - d_2|}{\max(d_1, d_2)}. \tag{9}$$

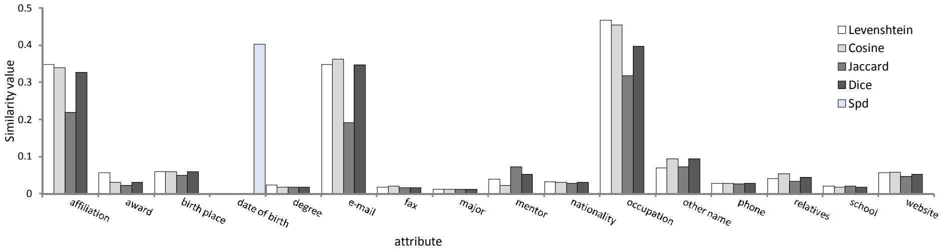


Figure 3. The results reported by different similarity measures for attributes of the co-referent person names in WePS-1 training dataset in terms of $B^3F_{\alpha=0.5}$

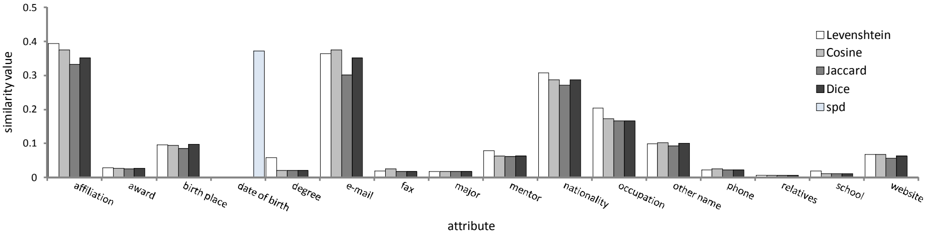


Figure 4. The results reported by different similarity measures for attributes of the co-referent person names in WePS-2 test dataset in terms of $B^3F_{\alpha=0.5}$

Results obtained by our experiments on the given datasets show that the normalized Levenshtein metric is appropriate for the attributes of *affiliation*, *award*, *degree*, *nationality*, *occupation*, and *school*; the Cosine similarity metric for the attributes of *relatives*, *phone*, *fax*, *e-mail*, *website*; and the Jaccard index for the attribute of *mentor*; and Dice coefficient for the attribute of *birth place*. For some attributes one or more similarity measures relatively reported the same results. For the attribute of *other name*, one can use Dice's coefficient or Cosine similarity measure. For the attribute of *major*, it is no matter which similarity measure is used; however, in our implementations, we used the normalized Levenshtein metric for the attribute of *major*. We notice that for attribute *date of birth*, we only use Spd measure.

As shown in Figures 3 and 4, some attributes report higher similarity values rather than others. This means some attributes are more important for name disambiguation. For example, the attribute of *date of birth* is more important than the attribute of *school* for resolving name ambiguity. This issue testifies that we should give different weight for each attribute according to its impact on name disambiguation. To compute β_a, the weight of attribute a, we use its specific similarity measure to compute the average similarity between co-referent persons based on attribute a. The resulting average similarity value is considered as weight of attribute a. To fulfil this aim, we first compute Equation (7) using the specified appropriate similarity measure of the attribute a. We then compute the pairwise similarity between persons using Equation (6) in terms of attribute a. In some cases that we deal with missing data, we consider 1 as the similarity value. Finally, to compute β_a, we calculate the average of pairwise similarity for all co-referent persons:

$$\beta_a = \left(\frac{\sum_{i,j=1}^{R} M_a\left(e_i, e_j\right)}{R} \right) / \left(\sum_{k=1}^{m} \max\left(\beta_{ak}\right) \right) \tag{10}$$

where R is the number of co-referent persons; e_i and e_j are the i^{th} and j^{th} co-referent entities, respectively. Table 1 shows the appropriate similarity measure and the weight for each attribute on WePS-1 training and WePS-2 training dataset.

		WePS-1 Training	WePS-2 Training
Method	Similarity Metric	β_a	β_a
Affiliation	len	0.164	0.181
Award	len	0.027	0.013
Birth place	dic	0.028	0.044
Date of birth	spd	0.191	0.170
Degree	len	0.011	0.027
Email	cos	0.172	0.172
Fax	cos	0.010	0.011
Major	len	0.006	0.008
Mentor	jac	0.034	0.036
Nationality	len	0.015	0.141
Occupation	len	0.221	0.093
Other name	dic	0.045	0.047
Phone	cos	0.013	0.011
Relatives	cos	0.026	0.003
School	len	0.010	0.009
Web site	cos	0.027	0.031

Table 1. The weight of attributes on WePS-1 training and WePS-2 training datasets

We designed another test to indicate that using appropriate similarity measure for each type of attribute class has great impact on the quality of name disambiguation. To do this, we tested the similarity measures in turn on the dataset, and computed the similarity matrix. We then use agglomerative clustering algorithm with single linkage merging strategy to group the person names. We compare the results generated by the system with the ground truth included in the WePS training datasets to compute performances. Figure 5 shows the results corresponding to each similarity metric on the WePS-1 training dataset, and Figure 6 shows the results for the WePS-2 training dataset. In Figures 5 and 6, *Fa* refers to B-cubed F-score ($B^3F_{\alpha=0.5}$) [46, 47], and *Fp* refers to purity-based F-score ($F_{p=0.5}$) [46]. Combination similarity measures uses the normalized Levenshtein metric for the attributes of *affiliation*, *award*, *degree*, *nationality*, *occupation*, *school* and *major*, the Cosine similarity metric for the attributes of *relatives*, *phone*, *fax*, *e-mail*, and *website*, and the Jaccard index for attribute *mentor*, Dice coefficient for attribute *other name* and *birth place*, and Spd measure for attribute *date of birth.* As shown in Figures 5 and 6, the correct combination of similarity measures improves the performance of name disambiguation in terms of $B^3F_{\alpha=0.5}$ and $F_{p=0.5}$.

We also designed a test to assess the potential of Babelfy on cross-document name disambiguation. We let Babelfy to disambiguate person names and obtain their BabelNet synset id. We then group persons based on their BabelNet synset id. For this purpose, we use a dump of WePS-1 training and WePS-2 training datasets. The sampled dump contains person names that are taken from Wikipedia. The reason to this choice is that the knowledge base of Babelfy, BabelNet is primarily constructed by linking Wikipedia to WordNet. Figure 7 a) shows the results obtained by the Babelfy system and the attribute-based method (with combination similarity measure) on the sample dump drawn from the WePS-1 training, and Figure 7 b) shows the results obtained for sample dump drawn from the WePS-2 training dataset. In Figure 7, *AV_Combination* shows the attribute-based method equipped with combination similarity measure. Given the results in Figure 7, we conclude that the majority of information for name disambiguation is given in the web pages being processed. This implies that our name disambiguation system does not heavily rely on the Babelfy disambiguation results. However, incorporating the Babelfy and combining it with other name disambiguation methods such as attribute-based method can improve the overall performance of system.

At the end of the graph creation process, we have an undirected weighted graph summarizing all of the information about entities contained in the given web text documents, and the given external knowledge base. The remainder components of the name disambiguation system work with this rich graph instead of web text documents. This enables us to use optimal graph mining algorithms for name disambiguation task. Figure 8 shows an example of mapping people's attributes and social links extracted from the web page given in Figure 2 into a graph. Figure 8 a) shows the discourse profiles extracted by profile extraction system. In Figure 8 a), the local attributes are shown in black colour and external attributes obtained by profile enrichment are shown in blue colour. The profile information and social links are mapped to a graph shown in Figure 8 b). In Figure 8 b), the structure node corresponding to target person *"Dan Jurafsky"* is shown in filled rectangle, the structure nodes for other persons are shown in rectangles, the structure nodes corresponding

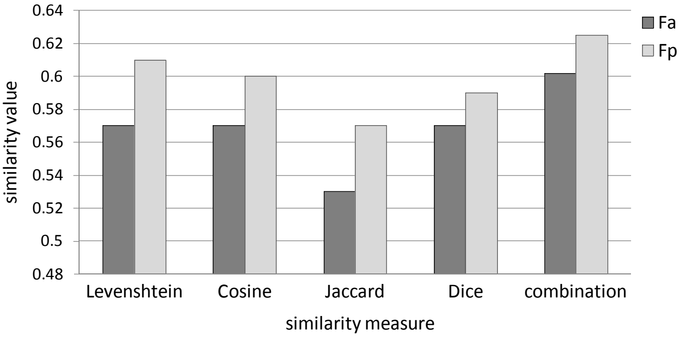


Figure 5. The performance of attribute-based name disambiguation method on WePS-1 training dataset using different similarity measures

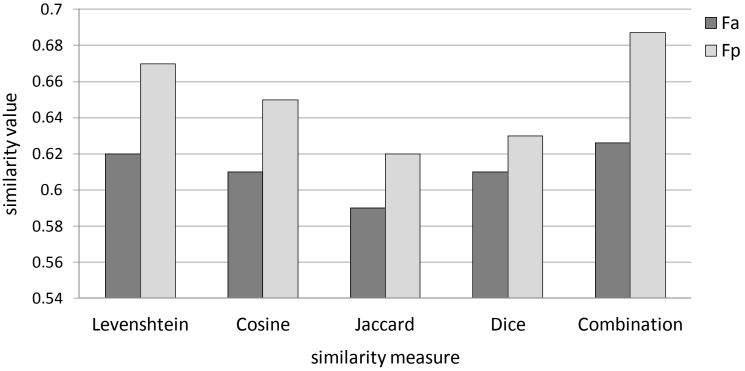
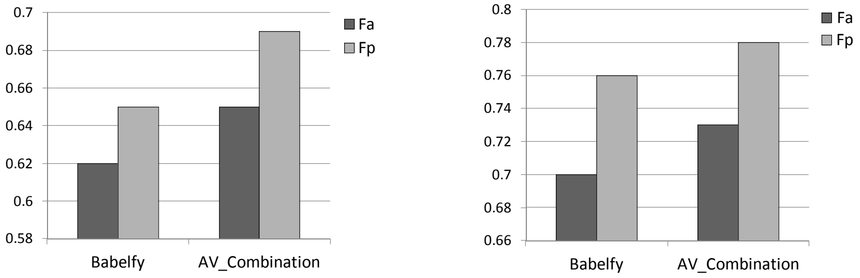


Figure 6. The performance of attribute-based name disambiguation method on WePS-2 training dataset using different similarity measures

to organizations in triangles, attribute nodes in round ellipses, structure edges by solid lines, and attribute edges by dotted lines.

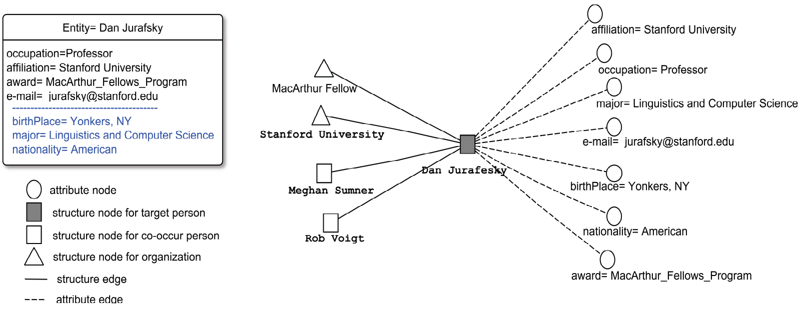
3.3 Similarity Computation

Our similarity metric relies on the node closeness in the graph through both structure and attribute edges. The main idea behind our similarity metric is that "two people are closely related if they share more common attributes and linked through many common entities". Our similarity metric obeys the theory of homophily [42], the principle that people with similar characteristics tend to form a relationship. People in a homophilic relationship share similar characteristics.



a) b)

Figure 7. The performance of Babelfy and AV_Combination method on a dump of Wikipedia person names drawn from a) WePS-1 training dataset and b) WePS-2 training dataset


Entity= Dan Jurafsky
occupation=Professor
affiliation= Stanford University
award= MacArthur_Fellows_Program
e-mail= jurafsky@stanford.edu
birthPlace= Yonkers, NY
major= Linguistics and Computer Science
nationality= American
attribute node
structure node for target person
structure node for co-occur person
structure node for organization
structure edge
attribute edge
MacArthur Fellow
Stanford University
Meghan Sumner
Rob Voigt
Dan Jurafesky
affiliation= Stanford University
occupation= Professor
major= Linguistics and Computer Science
e-mail= jurafsky@stanford.edu
birthPlace= Yonkers, NY
nationality= American
award= MacArthur_Fellows_Program

a) b)

Figure 8. An example of mapping the sample web page given in Figure 2 to attribute-relationship graph; a) discourse profile extracted by our profiling system and enriched with external global attributes; b) attribute-relationship graph. Note that the weights of attribute edges and structure edges are not given here.

Since people's attributes and social links are two different types of information, integrating both of them in a unified similarity metric for name disambiguation is so challenging.

Here, we adapt and modify the recently proposed neighbourhood random walk distance (NRWD) [26] to built our similarity measure. Zhou et al. [26] applied NRWD for graph partitioning and demonstrated that it can improve the performance of clustering. Let τ be a path from node u to node v, whose length is $l(\tau)$ with transition probability $p(\tau)$, L be the longest length that the random walk can proceed, and $\gamma \in (0, 1)$ be the restart probability. The similarity between nodes u

and v using NRWD can be formulated as follows:

$$R(u,v) = \sum_{\substack{\tau : u \to v \\ l(\tau) \leq L}} \gamma(1-\gamma)^{l(\tau)} \times p(\tau) \tag{11}$$

The value of $R(u,v)$ is in the interval $[0,1]$. The transition probability $p(\tau)$ from structure node u to structure node v through structure edge es is computed as follows:

$$p_{u,v}(\tau) = \begin{cases} \frac{ws}{|N(u)| \times ws + |A(u)| \times wa}, & \text{if } (u,v) \in ES, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $N(u)$ represents a set of neighbours of entity u that are connected through structure edges, $A(u)$ represents a set of neighbours of entity u that are connected through attribute edges, ws is the weight of structure edge es, and wa indicates the importance coefficient of attribute edge ea. Similarly, the transition probability from structure node u to attribute node $v_a \in AV$ through an attribute edge ea is calculated as follows:

$$p_{u,v_a}(\tau) = \begin{cases} \frac{wa}{|N(u)| \times ws + |A(u)| \times wa}, & \text{if } (u,v_a) \in EA, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

The transition probability from attribute node v_a to structure node u through an attribute edge ea is computed as follows:

$$p_{v_a,u}(\tau) = \begin{cases} \frac{wa}{|N(v_a)|}, & \text{if } (v_a,u) \in EA, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $N(v_a)$ is the set of structure nodes that share the attribute node v_a. Since there is no edge between any pair of attribute nodes v_{ai} and v_{aj}, the transition probability between attribute nodes is 0:

$$p_{v_{ai},v_{aj}}(\tau) = 0. \tag{15}$$

3.4 Graph Clustering

Our procedure for clustering takes as input the attribute-relationship graph G, and uses the recently proposed BIC-Means [43], an efficient graph clustering algorithm to partition the graph into sets of clusters. BIC-Means is a bisecting version of the incremental K-means clustering algorithm equipped with Bayesian information criterion (BIC) [44] as a termination criterion. It starts with a single cluster C_1^0 containing all the objects, and bisects this cluster into two sub clusters C_1^1 and C_2^1 by applying the incremental K-means algorithm [43]. C_i^t refers to the i^{th} cluster at the t^{th} level. The algorithm continues by splitting each cluster C_i^t into two sub clusters C_{i1}^{t+1} and C_{i2}^{t+1}, if the BIC score of sub clusters C_{i1}^{t+1} and C_{i2}^{t+1} is greater than the BIC score of C_i^t. It terminates the divisive procedure when there is no separable leaf cluster according to the BIC score. Our reason to use BIC-Means as clustering algorithm comes from the fact that

1. it is an efficient algorithm, which has low computational cost to hierarchically clustering large-scale data sets,
2. it does not need the number of clusters as input parameter, considering the notion that we do not know the number of clusters previously, and
3. it is robust against the parameter setting.

We run the BIC-Means clustering with the similarity metric given in Equation (8), which measures the closeness between nodes. In clustering process, we use a blocking technique to avoid computational bottlenecks. By this way, we apply clustering algorithm on documents that are about an ambiguous person.

As mentioned before, personal attributes are heterogeneous; therefore each class of attribute has different impact on clustering quality. This enforces us to assign different weights for each particular attribute class. In order to determine the importance coefficient for the attribute classes, we propose a dynamic weight learning approach. Our approach is an extension to the weight learning schema taken by [26]. Our approach for learning the weights of attributes is as follows. Let $Wa^t = \{wa_1^t, wa_2^t, \ldots, wa_m^t\}$ be the attribute weights in the t^{th} iteration of clustering. We initialize $wa_1^0 = wa_2^0 = \ldots = wa_m^0 = 1$ at first iteration of clustering. At each bisection stage of clustering process, we update the value of wa_i^t with a weight increment Δwa_i^t, which indicates the weight update of attribute a_i between the iteration t and $t+1$. The weight of attribute a_i in iteration $t+1$ is calculated as follows:

$$wa_i^{t+1} = \frac{1}{2}(wa_i^t + \Delta wa_i^t). \tag{16}$$

Δaw_i^t is computed by the following formula:

$$\Delta wa_i^t = \frac{m \times \sum_{j=1}^{k} \mathrm{H}(a_i, C_j)}{\sum_{f=1}^{m} \sum_{j=1}^{k} \mathrm{H}(a_f, C_j)} \tag{17}$$

where m is the number of attribute classes in attribute vocabulary A, and the scoring function $H(a_i, C_j)$ quantifies the saliency of attribute a_i in cluster C_j. The reasoning behind this weighting schema is as follows: an attribute a_i has great impact on clustering, if a large number of nodes within clusters share the same value on a_i, on the other hand, if nodes within clusters have quite different values on a_i, then it has not a good clustering tendency. To compute $H(a_i, C_j)$, we adopt and modify the term frequency (TF) weighting component of the TF-IDF method developed by [45]. We define $H(a_i, C_j)$ as follows:

$$H(a_i, C_j) = \alpha \times \Phi_1(a_i, C_j) + (1-\alpha) \times \Phi_2(a_i, C_j) \tag{18}$$

where $\Phi_(a_i, C_j)$ is the relative intra-document TF, $\Phi_2(a_i, C_j)$ is the normalized length regularized TF, and α is a tunable importance coefficient. Without loss of generality, we set $\alpha = 0.5$. $\Phi_1(a_i, C_j)$ controls the distribution of attributes within a document. It computes the importance of attribute a_i by considering its frequency relative to the average frequency of attributes within cluster C_j. We define $\Phi_1(a_i, C_j)$ as follows:

$$\Phi_1(a_i, C_j) = \frac{\log_2(1 + \mathrm{TF}(a_i, C_j))}{\log_2(1 + \mathrm{MTF}(C_j))} \tag{19}$$

where $\mathrm{TF}(a_i, C_j)$ is the frequency of attribute a_i in cluster C_j, and $\mathrm{MTF}(C_j)$ denotes average attribute frequency within cluster C_j. $\Phi_2(a_i, C_j)$ normalizes the attribute frequency by considering the number of objects available in a cluster:

$$\Phi_2(a_i, C_j) = \mathrm{TF}(a_i, C_j) \times \log_2\left(1 + \frac{\mu_G}{|C_j|}\right) \tag{20}$$

where μ_G is the average length of clusters in cluster collection G. The general principle behind scoring function $\Phi_2(a_i, C_j)$ is that if two clusters have different lengths and the same TF values for a given attribute a_i, then the contribution of $\mathrm{TF}(a_i, C_j)$ should be higher for the shorter cluster. Thus, it is necessary to regulate the TF value in accordance with the length of clusters. We define $\mathrm{TF}(a_i, C_j)$ as follows:

$$\mathrm{TF}(a_i, C_j) = \sum_{u \in C_j} X_i(u, c_j) \tag{21}$$

where c_j is the centroid of cluster C_j, and $X_i(u, c_j)$ determines whether node u and v share a same value on attribute a_i.

$$X_i(u, c_j) = \begin{cases} 1, & \text{if } (a_i \in u \,\&\, a_i \in c_j), \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

4 EXPERIMENTS AND RESULTS

In this section, we first describe benchmark datasets and performance metrics, and then give the results obtained by our approach, baseline methods, and the competitive state-of-the-art algorithms.

4.1 Dataset

We used two standard benchmark datasets to validate and compare our approach with other algorithms developed for disambiguation of people names on the web. These datasets include WePS-1 test dataset[3] [13], and WePS-2 test dataset[4] [14]. Each dataset consisted of collections of web pages obtained from the results for

[3] Available for download at `http://nlp.uned.es/weps/weps-1/weps1-data`

[4] Available for download at `http://nlp.uned.es/weps/weps-2/weps2-data`

a person name query to an Internet search engine. For each name the top ranked N web pages (100 for WePS-1 dataset and 150 for WePS-2 dataset) from the search results were included into the dataset. For each person name the ground truth files are also provided by human annotators. These datasets provide a real corpus, which can test a disambiguation system for personal names with varying ambiguity and in different domains. Both WePS-1 test and WePS-2 test datasets consisted of 30 web page collections, each one corresponding to an ambiguous person name. These 30 person names were chosen from three different sources (10 name sets from Wikipedia, 10 from the US Census, and 10 from ACL conference) in order to provide different ambiguity scenarios.

4.2 Performance Measures

Various measures are presented to assess the quality of name disambiguation methods. We conducted evaluations using two types of scoring measures, *B-cubed* scoring measure and the *purity-based* scoring measure. We use three B-cubed scoring measures including B-cubed precision (B^3P), recall (B^3R), and F-score (B^3F_α). A more detailed discussion of these quality metrics is given in [46, 47]. In addition to B-cubed measures, we used three purity-based scoring measures, including purity (Pr), inverse purity (IPr), and harmonic mean of purity and inverse purity (F_p-score). A more detailed discussion of purity-based quality metrics is given in [46].

4.3 Numerical Results and Discussion

In addition to comparing our algorithm to prominent solutions and the state-of-the-art methods in the literature, we also implemented five solutions as our baseline methods. We compare our name disambiguation system with five baseline methods. The baseline algorithms include bag of words model (*BOW* model) [3], social network based method (*SN* model) [25], attribute-based method (*AV* model) [20], *ALL_IN_ONE* [14], and *ONE_IN_ONE* method [14]. The BOW baseline is based on the traditional bag of words models: agglomerative vector space clustering with TF/IDF weighting schema. The BOW method is widely employed as a benchmark in a series of previous works; e.g. in [3, 8, 24]. The SN baseline [25] represents the approach where only the social relationships are employed for name disambiguation. The AV baseline [20] is an attribute-based name disambiguation algorithm, which relies only on the people attributes, and it ignores social relations between people. The ALL_IN_ONE and ONE_IN_ONE baselines are provided by the WePS share-task [14]. In ALL_IN_ONE baseline all documents related to a person are placed in a single cluster. In contrast, in ONE_IN_ONE baseline each document is included in a separate cluster. We notice that we implemented the baseline methods as described in their original paper. We perform all experiments on a 3 GHz, and 4 GB RAM Personal Computer Intel® Pentium® 4. We coded all the mentioned algorithms using JAVA and MATLAB language.

We notice that there are two parameters needed to be configured. These parameters include the knowledge base matching threshold T and the random walk length L. To configure these parameters, we adopt a k-fold cross validation strategy (with $k = 4$). We at first randomly partition the training data into four equal folds. At each iteration, we used one of the folds as test data and the other three folds as training data. At each iteration, we empirically learn the value of parameters L and T on training data that provides the best performances for name disambiguation. The final results we report are averaged over four independent iterations. Setting T to 0.60, the results obtained for different values of L from 2 (2-hop neighbours) to 10 (10-hop neighbours) on training data are given in Table 2. We observe that the algorithm achieves the best result when the value of L is between 4 and 8. The results indicate that the algorithm is not very sensitive to L, thus the exact tuning of the parameter is not an important matter. We set $L = 6$ for the remainder of our experiments. Table 3 shows the results obtained on training datasets with different values of T from 0 (append all external attributes into local discourse profile) to 1 (ignore external attributes) with step 0.2. We observe that the algorithm achieves its best performance when the value of T is between 0.4 and 0.8. The results indicate that the algorithm is not very sensitive to T, thus the exact tuning of parameter T is not an important matter. We set $T = 0.60$ for the remainder of our experiments.

Dataset	L				
	2	4	6	8	10
WePS-1	69.8	**70.7**	70.5	69.5	69.3
WePS-2	74.2	76.5	**77.8**	77.4	76.1

Table 2. The results obtained by our approach for different values of L on training datasets in terms of $B^3F_{\alpha=0.5}$

Dataset	T					
	0	0.2	0.4	0.6	0.8	1
WePS-1	69.5	69.9	70.1	**70.4**	69.7	68.8
WePS-2	75.7	76.2	**76.8**	76.2	75.2	74.6

Table 3. The results obtained by our approach for different values of knowledge base matching threshold T on training datasets in terms of $B^3F_{\alpha=0.5}$

Table 4 shows the results obtained by our name disambiguation methods on WePS-1 test dataset. Table 5 shows the results for WePS-2 test dataset. In order to indicate the effect of each clustering feature, in Table 4 and 5, we begin with the feature of social links and then add features of local discourse profile attributes and external attributes one by one. The results clearly show the effect of profile enrichment and integrating attributes with social relationships. In Table 4 and 5, we notice that the performance is consistently increasing when incorporating more clustering features. The final feature model (*social links* + *local attributes* + *external attributes*) achieves the best performances. In Table 4, the performances on WePS-1 test dataset increase about +1.9 % in terms of $B^3F_{\alpha=0.5}$ and +2.06 % in terms of $F_{p=0.5}$ from the local feature model *social links*+*local attributes* to final feature model *social links* + *local attributes* + *external attributes*. As given in Table 5, the improvement rate from the local feature model to final feature model is about +1.55 % in terms of $B^3F_{\alpha=0.5}$ and +2.33 % in terms of $F_{p=0.5}$ for WePS-2 test dataset. This implies that the majority of information for name disambiguation is given in the web pages being processed. However, incorporating the external attributes improves performances.

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
Social links	67.3	73.7	68.1	71.5	87.1	80.4
+ Local attributes	74.4	80.5	75.2	78.6	89.3	82.2
+ external attributes	75.2	81.6	77.1	79.1	90.8	84.26

Table 4. Performances of our name disambiguation approaches on WePS-1 test datasets

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
Social links	66.3	74.5	68.7	68.2	87.4	72.0
+ Local attributes	84.7	80.1	82.5	83.6	89.24	86.3
+ external attributes	86.2	82.9	84.05	85.4	90.8	88.63

Table 5. Performances of our name disambiguation approaches on WePS-2 test datasets

Table 6 shows the best performance obtained from the baselines and our method on WePS-1 test dataset. Table 7 shows the results for WePS-2 test dataset. As shown in Table 6 and 7, our method clearly outperforms the baseline methods for both datasets in terms of both $B^3F_{\alpha=0.5}$ and $F_{p=0.5}$. For WePS-1 dataset, on average our method outperforms BOW, SN, AV, ALL_IN_ONE, and ONE_IN_ONE by +7.8 %, 8.5 %, 16.4 %, 21.1 %, and 45.1 %, respectively, in terms of $B^3F_{\alpha=0.5}$, and +8.76 %, 5.76 %, 21.76 %, 12.86 % and 50.26 %, respectively, in terms of $F_{p=0.5}$. The improvement is also evident for WePS-2 dataset, in which our method obtains +11.55 %, 16.75 %, 22.65 %, 31.05 %, and 50.05 % improvement compared to BOW, SN, AV, ALL_IN_ONE, and ONE_IN_ONE, respectively, in terms of $B^3F_{\alpha=0.5}$, and with respect to $F_{p=0.5}$, the improvements are +11.83 %, 18.13 %, 21.33 %, 21.43 % and 54.63 %, respectively. The ONE_IN_ONE baseline obtained the best result in terms of B^3P and Pr measure on both WePS-1 and WePS-2 dataset. The ALL_IN_ONE baseline outperformed other algorithms in terms of B^3R and IPr measure. The higher B^3P and Pr for ONE_IN_ONE baseline arises from the fact that in WePS-1 and WePS-2 datasets documents are distributed among the clusters. Since in average half of the documents in the dataset belong to one specific person, the ALL_IN_ONE baseline gave better results in terms of IPr and B^3R.

Table 8 summarizes the average performance obtained by our proposed method and four state-of-the-art methods for the benchmark datasets. We compared the

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
BOW	62.1	75.5	69.3	70.4	85.0	75.5
SN	65.0	73.5	68.6	69.7	89.2	78.5
AV	59.4	68.4	60.7	62.1	71.4	62.5
ALL_IN_ONE	44.0	**100**	56.0	59.0	**100**	71.4
ONE_IN_ONE	**100**	20.0	32.0	**100**	22.0	34.0
Our method	75.2	81.6	**77.1**	79.1	90.8	**84.26**

Table 6. Comparison of results obtained by baselines and our method on WePS-1 test dataset

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
BOW	66.2	80.5	72.5	78.1	82.0	76.8
SN	64.2	81.1	67.3	64.2	90.7	70.5
AV	62.5	77.7	61.4	62.6	79.4	67.3
ALL_IN_ONE	43.0	**100**	53.0	56.0	**100**	67.2
ONE_IN_ONE	**100**	24.0	34.0	**100**	24.0	34.0
Our method	86.2	82.9	**84.05**	85.4	90.8	**88.63**

Table 7. Comparison of results obtained by baselines and our method on WePS-2 test dataset

results obtained by our method with those reported in Han and Zhao [3] and Chen et al. [4] on WePS-1 and WePS-2 test dataset; Dutta and Weikum [6] and Yerva et al. [11] on WePS-2 test dataset. We notice that the comparison is not precise, because the mentioned algorithms were implemented and tested with different settings on machines with different processing characteristics. In Table 8, Han and Zhao [3] did not report obtained B-cubed scores. As shown in Table 8, our approach performs well on datasets, exceeding or matching the best performance obtained by the state-of-the-art methods in terms of $B^3F_{\alpha=0.5}$.

		Han and Zhao [3]	Yerva et al. [11]	Chen et al. [4]	Dutta and Weikum [6]	Our method
WePS-1	$B^3F_{\alpha=0.5}$	–	–	76	–	77.1
	$F_{p=0.5}$	84	–	90	–	84.26
WePS-2	$B^3F_{\alpha=0.5}$	–	74.7	82	83.48	84.05
	$F_{p=0.5}$	86	78.8	89	–	88.63

Table 8. Comparison of results obtained by state-of-the-art methods and our method on WePS-1 test and WePS-2 test dataset

In this research, we concerned to answer the question "what is the maximum performance that a name disambiguation system can obtain if it uses information found in the web documents (local information) and attributes from external knowledge base (global information)?" The results indicate that integrating global information with local information improves the performance.

The results look promising but far from ideal. This justifies that name disambiguation in web is a demanding problem and more effort is needed in this respect. We analysed the failed cases where the algorithm could not resolve ambiguity. Our manual investigation over failed cases reveals that almost half of the failures were because of the inefficiency of pre-processing subtasks including named-entity tagging, intra-document co-reference resolution, and the erroneous attributes in discourse profiles extracted by profile extraction system, and not because of the inefficiency of our name disambiguation approach. For example, the employed co-reference resolution system identifies incorrect co-reference chains of mentions, which propagate to the later stages. Similarly, named-entity tagger could not correctly identify entities in web documents, which leads to a significant degradation in performance of the name disambiguation system. Pre-processing tools could not perform effectively on web documents because

1. most of the pre-processing tools have trained on news corpora,
2. web documents are quite diverse, noisy and contain partial information about entities.

These problems decrease the performance of name disambiguation system. Therefore, it is needed to develop pre-processing components suitable for web documents. Nonetheless, improving pre-processing tools is beyond the scope of this paper. We have attempted to improve the results by exploiting the context-independent features such as social links.

To summarize, our approach has several advantages. Using people profile attributes for name disambiguation degrades the undesired effect of noisy data and increases the efficiency of name disambiguation. It also decreases time complexity because instead of raw web text, we only compare structured attributes to compute similarities. Our approach could be considered as a language-neutral and domain-independent approach, because instead of raw text, it works on abstract information including entity attributes and social links. It could be extended to a more complex setting and applied to many applications, for example, social network extraction and information integration. In our implementations, we only consider the weighted co-occurrence of entities as social relationships. We simulate implicit social relationships and give meaningful semantics to meaningless co-occurrence relationships by exploiting personal attributes. In general, both insufficient entity-centric attributes and noisy social relations affect the robustness of name disambiguation. The results show that our method is capable to integrate both local and global attributes and social relationships, and provide more information for name disambiguation.

5 CONCLUSIONS

In this paper, we proposed a cross-document person name disambiguation approach in web context. Our approach attempts to use both the local information about persons available in the given web pages and the global information from external knowledge bases. The local information includes profile attributes and social links, and the global information includes the attributes extracted from external knowledge bases. We evaluated the effectiveness of our proposed approach on WePS-1 and WePS-2 datasets. It achieved 77.1 % B-cubed F-score on the WePS-1 test dataset, and 84.05 % on WePS-2 test dataset. The results indicate that our approach is robust enough and outperformed the state-of-the-art name disambiguation approaches. Although our results seem satisfactory, some points to improve our research have remained. One of the most interesting directions is to consider other clustering features and media such as images, and integrating them with a text for name disambiguation, which can improve the results. As the final results show, our system depends on the effectiveness of three main subtasks including profile extraction, profile enrichment and social link extraction. Therefore another interesting future work is to develop more robust systems for these subtasks, which subsequently can improve the overall quality of the name disambiguation system. As the information about entities on the web changes over time, an interesting future work is to develop a dynamic name disambiguation system using dynamic clustering algorithms. Finally, we plan to develop a generic name disambiguation system in which entities are not limited to persons.

REFERENCES

[1] Barla, M.—Tvarožek, M.—Bieliková, M.: Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems. Computing and Informatics, Vol. 28, 2009, No. 4, pp. 399–427.

[2] Barla M.: Towards Social-Based User Modeling and Personalization. Information Sciences and Technologies Bulletin of the ACM Slovakia, ACM Slovakia, Vol. 3, 2011, No. 1, pp. 52–60.

[3] Han X.—Zhao, J.: Structural Semantic Relatedness: A Knowledge-Based Method to Named Entity Disambiguation. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 50–59.

[4] Chen, Y.—Mei Lee, S. Y.—Huang, C. R.: A Robust Web Personal Name Information Extraction System. Expert Systems with Applications, Vol. 39, 2012, No. 3, pp. 2690–2699, doi: 10.1016/j.eswa.2011.08.125.

[5] Khabsa, M.—Treeratpituk, P.—Giles, C. L.: Online Person Name Disambiguation with Constraints Categories and Subject Descriptors. Proceedings of the 15th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL '15), ACM, 2015, pp. 37–46, doi: 10.1145/2756406.2756915.

[6] Dutta S.—Weikum, G.: Cross-Document Co-Reference Resolution Using Sample-Based Clustering with Knowledge Enrichment. Transactions of the Association for Computational Linguistics, Vol. 3, 2015, pp. 15–28, doi: 10.18653/v1/d15-1101.

[7] Bagga, A.—Baldwin, B.: Entity-Based Cross-Document Coreferencing Using the Vector Space Model. Proceedings of the 36th Annual Meeting of the Association

for Computational Linguistics and 17th International Conference on Computational Linguistics – Volume 1 (ACL '98/COLING '98), 1998, pp. 79–85.

[8] Kalashnikov, D. V.—Chen, Z.—Mehrotra, S.—Nuray-Turan, R.: Web People Search via Connection Analysis. IEEE Transactions on Knowledge and Data Engineering, Vol. 20, 2008, No. 11, pp. 1550–1565, doi: 10.1109/tkde.2008.78.

[9] Gong, J.—Oard, D. W.: Determine the Entity Number in Hierarchical Clustering for Web Personal Name Disambiguation. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009, doi: 10.1145/1571941.1572124.

[10] Chen, Z.—Kalashnikov, D. V.—Mehrotra, S.: Exploiting Context Analysis for Combining Multiple Entity Resolution Systems. Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09), 2009, pp. 207–218, doi: 10.1145/1559845.1559869.

[11] Yerva, S. R.—Miklós, Z.—Aberer, K.: Quality-Aware Similarity Assessment for Entity Matching in Web Data. Information Systems, Vol. 37, 2012, No. 4, pp. 336–351, doi: 10.1016/j.is.2011.09.007.

[12] Brandes, U.—Erlebach, T. (Eds.): Network Analysis: Methodological Foundations. Springer-Verlag, Berlin, Heidelberg, Theoretical Computer Science and General Issues, Vol. 3418, 2005, doi: 10.1007/b106453.

[13] Artiles, J.—Gonzalo, J.—Sekine, S.: The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval '07), Prague, Czech Republic, 2007, pp. 64–69, doi: 10.3115/1621474.1621486.

[14] Artiles, J.—Gonzalo, J.—Sekine, S.: Weps 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[15] Brizan, D. G.—Tansel, A. U.: A Survey of Entity Resolution and Record Linkage Methodologies. Communications of the IIMA, Vol. 6, 2006, No. 3, pp. 41–50.

[16] Elmagarmid, A. K.—Ipeirotis, P. G.—Verykios, V. S.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering, Vol. 19, 2007, No. 1, pp. 1–16, doi: 10.1109/tkde.2007.250581.

[17] Köpcke, H.—Rahm, E.: Frameworks for Entity Matching: A Comparison. Data and Knowledge Engineering, Vol. 69, 2010, No. 2, pp. 197–210, doi: 10.1016/j.datak.2009.10.003.

[18] Ferreira, A. A.—Gonçalves, M. A.—Laender, A. H. F.: A Brief Survey of Automatic Methods for Author Name Disambiguation. ACM SIGMOD Record, Vol. 41, 2012, No. 2, pp. 15–26, doi: 10.1145/2350036.2350040.

[19] Mann, G. S.—Yarowsky, D.: Unsupervised Personal Name Disambiguation. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 – Volume 4 (CONLL '03), 2003, pp. 33–40, doi: 10.3115/1119176.1119181.

[20] Nagy, I.: Person Attribute Extraction from the Textual Parts of Web Pages. Acta Cybernetica, Vol. 20, 2012, No. 3, pp. 419–440, doi: 10.14232/actacyb.20.3.2012.4.

[21] Srinivasan, H.—Chen, J.—Srihari, R.: Cross Document Person Name Disambiguation Using Entity Profiles. Proceedings of the Text Analysis Conference (TAC) Workshop, 2009.

[22] Lan, M.—Zhang, Y. Z.—Lu, Y.—Su, J.—Tan, C. L.: Which Who Are They? People Attribute Extraction and Disambiguation in Web Search Results. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[23] Song, F.—Cohen, R.—Lin, S.: Web People Search Based on Locality and Relative Similarity Measures. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[24] Bekkerman, R.—McCallum, A.: Disambiguating Web Appearances of People in a Social Network. Proceedings of the 14th International Conference on World Wide Web (WWW '05), 2005, pp. 463–470, doi: 10.1145/1060745.1060813.

[25] Malin, B.: Unsupervised Name Disambiguation via Social Network Similarity. SIAM SDM Workshop on Link Analysis, Counterterrorism and Security, 2005, pp. 93–102.

[26] Zhou, Y.—Cheng, H.—Yu, J. X.: Graph Clustering Based on Structural/Attribute Similarities. Proceedings of the VLDB Endowment, Vol. 2, 2009, No. 1, pp. 718–729, doi: 10.14778/1687627.1687709.

[27] Han, X.—Zhao, J.: Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. Proceedings of the 18th ACM Conference on Information and knowledge management (CIKM '09), 2009, pp. 215–224, doi: 10.1145/1645953.1645983.

[28] Li, Y.—Wang, C.—Han, F.—Han, J.—Roth, D.—Yan, X.: Mining Evidences for Named Entity Disambiguation. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 1070–1078, doi: 10.1145/2487575.2487681.

[29] Finkel, J. R.—Grenager, T.—Manning, C.: Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005), 2005, pp. 363–370, doi: 10.3115/1219840.1219885.

[30] Lee, H.—Chang, A.—Peirsman, Y.—Chambers, N.—Surdeanu, M.—Jurafsky, D.: Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. Computational Linguistics, Vol. 39, 2013, No. 4, pp. 885–916, doi: 10.1162/coli_a_00152.

[31] Manning, C. D.—Surdeanu, M.—Bauer, J.—Finkel, J.—Bethard, S. J.—McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60, doi: 10.3115/v1/p14-5010.

[32] Minkov, E.—Wang, R. C.—Cohen, W. W.: Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP '05), 2005, pp. 443–450, doi: 10.3115/1220575.1220631.

[33] Cortes, C.—Vapnik, V.: Support-Vector Networks. Machine Learning, Vol. 20, 1995, No. 3, pp. 273–297, doi: 10.1007/bf00994018.

[34] Moro, A.—Raganato, A.—Navigli, R.: Entity Linking Meets Word Sense Disambiguation: A Unified Approach. Transactions of the Association for Computational Linguistics, Vol. 2, 2014, pp. 231–244, doi: 10.1145/1459352.1459355.

[35] Mazhari, S.—Fakhrahmad, S. M.—Sadeghbeygi, H.: A User-Profile-Based Friendship Recommendation Solution in Social Networks. Journal of Information Science, Vol. 41, 2015, No. 3, pp. 284–295, doi: 10.1177/0165551515569651.

[36] Auer, S.—Bizer, C.—Kobilarov, G.—Lehmann, J.—Cyganiak, R.—Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K. et al. (Eds.): The Semantic Web (ISWC 2007, ASWC 2007). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4825, 2007, pp. 722–735, doi: 10.1007/978-3-540-76298-0_52.

[37] Akcora, C. G.—Carminati, B.—Ferrari, E.: User Similarities on Social Networks. Social Network Analysis and Mining, Vol. 3, 2013, No. 3, pp. 475–495, doi: 10.1007/s13278-012-0090-8.

[38] Yujian, L.—Bo, L.: A Normalized Levenshtein Distance Metric. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, 2007, No. 6, pp. 1091–1095, doi: 10.1109/tpami.2007.1078.

[39] Liu, C.: Discriminant Analysis and Similarity Measure. Pattern Recognition, Vol. 47, 2014, No. 1, pp. 359–367, doi: 10.1016/j.patcog.2013.06.023.

[40] Koutrika, G.—Bercovitz, B.—Garcia-Molina, H.: FlexRecs: Expressing and Combining Flexible Recommendations. Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09), 2009, pp. 745–757, doi: 10.1145/1559845.1559923.

[41] Chang, A. X.—Manning, C. D.: SUTime: A Library for Recognizing and Normalizing Time Expressions. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC '12), 2012, pp. 3735–3740.

[42] Easley, D.—Kleinberg, J.: Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge University Press, 2010, doi: 10.1017/cbo9780511761942.

[43] Hourdakis, N.—Argyriou, M.—Petrakis, E. G. M.—Milios, E. E.: Hierarchical Clustering in Medical Document Collections: The BIC-Means Method. Journal of Digital Information Management, Vol. 8, 2010, No. 2, pp. 71–77.

[44] Schwarz, G. E.: Estimating the Dimension of a Model. The Annals of Statistics, Vol. 6, 1978, No. 2, pp. 461–464, doi: 10.1214/aos/1176344136.

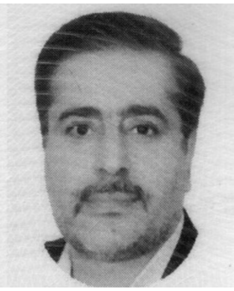
[45] Paik, J. H.: A Novel TF-IDF Weighting Scheme for Effective Ranking. Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13), 2013, pp. 343–352, doi: 10.1145/2484028.2484070.

[46] Amigó, E.—Gonzalo, J.—Artiles, J.—Verdejo, F.: A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. Information Retrieval, Vol. 12, 2009, No. 4, pp. 461–486.

[47] Cai, J.—Strube, M.: Evaluation Metrics For End-to-End Coreference Resolution Systems. Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2010), 2010, pp. 28–36.

[48] Christen, P.: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, Heidelberg, New York, Dordrecht, London, 2012.

Hojjat Emami is Assistant Professor at the Computer Engineering Department, University of Bonab. He received his B.Sc. degree in software engineering and M.Sc. degree in artificial intelligence from University of Tabriz. He received his Ph.D. degree in artificial intelligence under the supervision of Prof. H. Shirazi and Prof. A. A. Barforoush. His interest research areas are: data mining, machine learning, evolutionary computation, multi-agent systems, and social network analysis.



Hossein Shirazi received his B.Sc. from Mashhad University, Iran. He received his M.Sc. and Ph.D. in artificial intelligence from the University of New South Wales, Australia. He is currently Associate Professor at the Malek-Ashtar University of Technology, Iran.

Ahmad Abdollahzadeh Barforoush is Professor in Computer Engineering and IT Department of Amir Kabir University of Technology, Iran. He is the author of books entitled "Introduction to Distributed Artificial Intelligence" and "Software Quality Assurance Methodology". His research areas are: data quality, artificial intelligence, agent-based systems, automated negotiation, expert systems, natural language processing, decision support systems, business intelligence, data mining, data warehouse, and software engineering.