Serdica J. Computing **4** (2010), 217–242

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

CULTURAL KNOWLEDGE FOR NAMED ENTITY DISAMBIGUATION: A GRAPH-BASED SEMANTIC RELATEDNESS APPROACH

Anna Lisa Gentile, Ziqi Zhang, Lei Xia, José Iria

Abstract. One of the ultimate aims of Natural Language Processing is to automate the analysis of the meaning of text. A fundamental step in that direction consists in enabling effective ways to automatically link textual *references* to their *referents*, that is, real world objects. The work presented in this paper addresses the problem of attributing a sense to *proper names* in a given text, i.e., automatically associating words representing *Named Entities* with their *referents*. The method for Named Entity Disambiguation proposed here is based on the concept of semantic relatedness, which in this work is obtained via a graph-based model over Wikipedia. We show that, without building the traditional *bag of words* representation of the text, but instead only considering named entities within the text, the proposed method achieves results competitive with the state-of-the-art on two different datasets.

1. Introduction.

Reading a written text implies comprehension of the information that words are carrying. Comprehension is an intrinsic capability

ACM Computing Classification System (1998): I.2.7.
Key words: Wikipedia, Named Entity Disambiguation, Semantic Relatedness, Graph.

for a human, but not for a machine. Providing machines with such an ability, by "anchoring" meaning to words, is considered a task with great significance for Artificial Intelligence.

The focus of this work is on proper names, that is, on those words within text that represent *entites*: we want to attribute a meaning to such pieces of text since they carry high information value. This task is called Named Entity Disambiguation (NED). Many computational tasks may benefit from the additional metadata provided by NED. For example, since many web search queries concern named entities, NED is arguably a valuable pre-processing step to Information Retrieval techniques.

We propose an automatic method to associate a *unique sense* (the *referent*, that will also be designated in the remainder of this work as *meaning*, *concept* or simply *sense*) to each *entity* (the *reference* within the text), exploiting Wikipedia[1] as a freely available knowledge resource. Our proposed method constitutes a novel approach to the named entity disambiguation problem. We empirically show the effectiveness of the proposed methodology with two experimental sessions.

Our contributions are twofold. Firstly, we show the use of a random walk-based semantic relatedness approach to NED. Graph-based models have previously been applied to Word Sense Disambiguation (WSD) [1, 2, 3, 4] but not explored for the problem of NED. To the best of our knowledge, previous approaches to NED were based on the vector space model, treating *concepts* and context text as a bag of words [5, 6]. On the other hand, graph-based models have been utilized only for a specific type of NED, that of Person Name Disambiguation [7], or for specific domains, such as bibliographic citations [8]. Secondly, we introduce and show the efectiveness of an alternative way to model the context of a target entity, which, rather than consisting of the surrounding words, is only composed of the neighbor named entities present in the text.

The approach presented in this paper has the advantage of a clear separation of two concerns: the computation of semantic relatedness and the disambiguation of the named entities given the computed relatedness. In this way, the two independent steps can studied and improved separately. Compared to the best previously reported results by Cucerzan [6], an accuracy of 91.4% and of 88.3%, our method achieves a competitive accuracy of 91.5 % and 89.8% respectively, while adding the benefit of having two clearly separate steps, which expectedly opens the way to further improving each of the steps.

The work is structured as follows: Section 2 proposes an overview of the NED task, with focus on available solutions exploiting Wikipedia. Section 3

[1] `http://en.wikipedia.org/wiki/Wikipedia`

presents the proposed NED methodology, describing in details the four designed steps. Section 4 presents the experiments carried out to validate the proposed solution and finally conclude the paper with some remarks and an outline of future work.

2. Related Work. In Natural Language Processing, Named Entity Disambiguation is the problem of mapping mentions of entities in a text with the object they are referencing. It is a step further from Named Entity Recognition (NER), which involves the identification and classification of so-called named entities: expressions that refer to people, places, organizations, products, companies, and even dates, times, or monetary amounts, as stated in the Message Understanding Conferences (MUC) [9]. NED associates names with entities that are predefined in an external repository, which we will refer as *name inventory*: the task establishes a unique mapping between a mention of name in a text, the *surface form*, and the real world object in the *name inventory.* It can be assumed to have a dictionary of all possible entity entries.

Definition 1. *Named Entity Disambiguation is the task of mapping the list of entities appearing in a text with the correct unique real-world objects respectively referred. The occurrence of an entity in the text will be denominate with the term* ***reference*** *and the addressed real world object with the term* ***referent.*** *It is assumed the availability of an* ***entity inventory****, containing all possible referents, and that each reference has associated a set of candidate entities, that is, all real-world objects that a single reference could refer.*

Previous work on NED can be characterized as Knowledge-based methods, which comprehend Rule-based, Ontology-based, Wikipedia-based methods, Learning methods and Graph-based methods. In what follows we will give a panorama of such methods.

Peng et al. in [10] proposed a knowledge-based approach of processing documents to disambiguate not all types of Named Entities but only those representing locations. Their method automatically extract training data from large collection of documents based on local context disambiguation, and then sense profiles based on global context are generated automatically for disambiguation use. Local context of a location mention is direct neighbor words of the mention in a document. Global context of a sense is frequently co-located words of the sense in a collection of documents. The hypothesis behind this work is that every sense of location entity has different global context, so building a profile based on this kind of information could be useful to disambiguate location entity, by looking for similar context in profiles. To build these profiles the authors generate some training data from English newspaper articles (TDT4 collection[2]), automatically disambiguating a small portion of location mentions extracted with an Entity Recognition Tool, matching local context and information from a world gazetteer[3]: if the context of an entity appears in the parent or child node of a sense in the gazetteer, than that entity is stored as training data. All acquired training data have then been used to generate profiles, which have been indexed with a search engine tool. Disambiguation is then performed by querying the search engine over the profiles with ambiguous location. The answer to the query is obtained linearly combining three different scores: ranking position, local context, and the popularity of individual location sense. Results over 300 articles from the collection, manually annotated to construct a ground truth, show that weighting different scores could conduct to better results.

Aswani et al. [11] propose an instance unification methodology, that is, determining whether two instances to the same object in the real world. They focus on person names within an ontology containing publications, titles, authors, abstracts, etc., where different instances of these are created from bibliography records. The ontology population has been performed automatically, assuming that all authors of all publications are different and a corresponding instance is created in the ontology for each of them. Then the addressed instance unification task is to determine the number of distinct authors and insert the required "sameIndividualAs"statements in the ontology. The proposed approach combines the use of citation information (abstract, initials, titles and co-authorship information) with web mining and charge the attention at identifying which features lead to the best performance on the author disambiguation task. Results show that the information mined from the web contributes substantially towards the successful handling of highly ambiguous cases which lowered the performance of previous methods.

Several NED methods used a *name inventory* in the form of an ontology. García et al. [12] proposed a method for NED in the news domain that uses the NEWS ontology as *name inventory.* The proposed method, named IdentityRank, is inspired by PageRank algorithm [13]. The basic idea of the work is the *sematic coherence* of entities, that is, instances of a certain type usually occur in news of a certain category. Also the occurrence of an entity in a text gives information about other entities: similarly to what PageRank does with web pages, IdentityRank finds and ranks entities within a text, assuming that an instance has high rank if the sum of the ranks in the news item of the instances that typi-

[2] `http://projects.ldc.upenn.edu/TDT4/`

[3] `http://www.world-gazetteer.com/`

cally co-occur with it is high. The evaluation is done using a self-built corpus containing two ambiguous entities (Alonso, Georgia) with two different metrics: global and relative accuracy. Global accuracy is measured as total number of correct assignment entity/intance of the ontology divided by the total number of assignments. Relative accuracy for the entity used to build the corpus concerns the total number of correct assignment on the decisions of that entity divided by the total number of decisions of that entity. Results shown are mostly above the theoretically built baseline.

Hassell et al. [14] proposed a novel method for NED which utilizes background knowledge in the form of a populated ontology. The method works on unstructured text and uses different relationships in a document as well as from the ontology to provide clues in determining the correct entity. Successful experiments have been carried out on a collection of DBWorld[4] posts using a large scale, real-world ontology extracted from the DBLP bibliography website[5]. The authors argue that rich semantic metadata representations allow a variety of ways to describe a resource. The first step of the approach is specifying which entities from a populated ontology are to be spotted in text and later disambiguated. To do this,the authors indicate which literal property is the one that contains the "name" of entities to be spotted. After spotting entity names in text every potential match with the ontology is given a confidence score. The confidence score for each ambiguous entity is then adjusted based on whether existing information of the entity from the ontology matches accordingly to the relationship types found in the ontology (such as text-proximity, text co-occurance, semantic relationsships). Evaluation has been carried out on a manually constructed dataset, consisting of 20 documents from DBLP. Each entity appearing within documents has been labelled with the corresponding DBLP author's page: this link has been used within the ontology as the URI that uniquely identifies a researcher. Results are calculated in terms of precision and recall. Given the set of unique names from the dataset and the set of entities identified by the proposed algorithm, precision is the proportion of correctly identified entities with regard to B and recall is the proportion of correctly disambiguated entities with regard to A. The method reported a precision of 97.1% and a recall of 79.4%.

Nguyen and Cao [15] present a method that overcomes the problem of the shortage of available training data, by automatically generating an annotated corpus based on a specific ontology. They employ a machine learning model to disambiguate and identify named entities, by using an unsupervised method,

[4] `http://www.cs.wisc.edu/dbworld/`

[5] `http://www.informatik.uni-trier.de/ ley/db/`

based on Harris' Distributional Hypothesis [16], stating that words occurring in similar contexts tend to have similar senses. The proposed method also aims at exploring meaningful features for representing NEs in texts and a Knowledge Base (KB), then assigning each NE referred to in a text to the contextually most similar instance in the KB. Empirical evaluation shows that, while the ontology provides basic features of named entities, Wikipedia is a fertile source for additional features to construct accurate and robust named entity disambiguation systems.

Many studies that exploit Wikipedia as a knowledge source have emerged [17, 18, 19]. In particular, Wikipedia turned to be very useful for the problem of Named Entities due to its greater coverage than other popular resources, such as WordNet [20], resembling more to a dictionary, has little coverage over named entities [18]. Previous works exploited Wikipedia for the task of NER, e.g., to extract gazetteers [21] or as an external knowledge of features to use in a Conditional Random Field NER-tagger [22], to improve entity ranking in the field of Information Retrieval [23]. On the other hand, little has been carried out on the field of NED. The most related works on NED based on Wikipedia are those by Bunescu and Pasca [5] and Cucerzan [6].

Bunescu and Pasca consider the problem of NED as a ranking problem. The authors define a scoring function that takes into account the standard cosine similarity between words in the context of the query and words in the page content of Wikipedia entries, together with correlations between pages learned from the structure of the knowledge source (mostly using Wikipedia Categories assigned to the pages). Their method achieved accuracy between 55.4% and 84.8% [5].

Cucerzan proposes a very similar approach: the vectorial representation of the document is compared with the vectorial representation of the Wikipedia entities. In more details the proposed system represents each entity of Wikipedia as an *extended vector* with two principal components, corresponding to context and category information; then it builds the same kind of vector for each document. The disambiguation process consists of maximizing the *Context Agreement*, that is, the overlap between the document vector for the entity to disambiguate and each possible entity vector. Cucerzan proposed the Vector Space Model as a solutuion for the NED problem and the best result for this approach is an accuracy of 91.4% [6].

Both presented works [5, 6] are based on the Vector Space Model, which means that a pre-computation on the Wikipedia knowledge resource is needed to build the vector representation. What is more, their methods treat content in a

Wikipedia page as a bag-of-words (with the addition of categories information), without taking into account other structural elements in Wikipedia.

Han et al. [24] identify the key problem of Named Entity Disambiguation in measuring the similarity between occurrences of names. Traditional methods measure the similarity using essentially two kind of models: the bag of words (BOW) or Social networks. Both kind of measures have some limitations: BOW-based measures ignore all the semantic relations such as social relatedness between named entities, associative relatedness between concepts, polysemy and synonymy between key terms. Social networks can only capture the social relatedness between named entities. To overcome these deficiencies, Han and Zhao propose to use Wikipedia as the background knowledge for disambiguation, which surpasses other knowledge bases by the coverage of concepts, rich semantic information and up-to-date content and allow to measure the similarity between occurrences of names more accurately. Given the computed similarities, name observations are disambiguated by grouping them according to their represented entities, using the hierarchical agglomerative clustering (HAC) algorithm. The proposed method has been evaluated on the disambiguation of personal name, over the standard WePS data sets[6] using WePS proposed measures: *Purity* (homogeneity of the observations of names in the same cluster), *Inverse purity* (completeness of a cluster) and *F-Measure* (harmonic mean of purity and inverse purity). Empirical results show that the disambiguation performance of our method gets 10.7% improvement over the traditional BOW-based methods and 16.7% improvement over the traditional social network based methods.

Similarly to these methods, in our work we use Wikipedia as dictionary of all possible entity entries, yet proposing a novel method, which uses a graph model combing multiple features extracted from Wikipedia. We calculate *semantic relatedness* over this graph and we exploit obtained relatedness values to resolve the problem of NED.

In the ambit of NED, Learning methods, both supervised and unsupervised, have been mostly used on a subtask of NED, Person Name Disambiguation. Person Name Disambiguation is a particular kind of Entity Disambiguation and focuses the attention on persons instead of entities of whatever category. Disambiguation of person names can turn to be crucial in a Web-searching scenario, has also recognized by popular press; Reuters (March 13, 2003) observed the problem of name ambiguity to be a major stumbling block in personal name web searches. The problem has also been faced in one of SEMEVAL 2007 task[7], with the goal

[6] http://nlp.uned.es/weps/weps-1/weps-1-data

[7] http://nlp.uned.es/weps/

of grouping documents referring to the same individual.

Sugiyama et al. in [25] propose a semi-supervised clustering approach for the task of Personal Name Disambiguation. They integrate similar documents into a labeled document then they use agglomerative clustering: initially, each Web page is an individual cluster, and then two clusters with the largest similarity are iteratively merged to generate a new cluster until this similarity is less than a predefined threshold. The authors claim that if a seed page that describes a person is introduced, the clustering for personal name disambiguation would be much more accurate. Therefore, they use two kinds of seed page: (a) the article on each person in Wikipedia, and (b) the top-ranked Web page in the Web search results. Results show that this method, compared to pure agglomerative clustering, generates a smaller number of clusters, making easier for a user the task of browsing Web pages based on each personal entity.

A cross-document Person Name Disambiguation system is presented in [26]. The goal is to cluster documents: each cluster must contain all documents referring to the same person. The authors introduce entity profiles, information collected for each ambiguous person in the entire document collection. Also, they represent entities in a vector-space model (VSM), using a modified term frequency-inverse document frequency (TF-IDF) weighting scheme. Disambiguation is then performed via single-link hierarchical agglomerative clustering. Experiments are carried out on two corpora: the Bagga Baldwin corpus [27], which contains one ambiguous name and the English Boulder name corpora, a news corpus acquired from a web search, containing four sub corpora each corresponding to one of four different ambiguous person names, already used by [28]. Results show an improvement to the state of the art on same corpora, with an F-measure of 94.03%.

Han et al. [29] consider the problem of author names ambiguity in publications or bibliographies. They propose the use of a K-means clustering algorithm based on an extensible Naive Bayes probability model. The algorithm is based on three features collected from citations: co-author names, the title of the paper and the title of the journal or proceedings. The work is based on the assumption that a researcher usually has research areas that are stable over a period and tends to co-author papers with a particular group of people during that period. The disambiguation system, given an author name, clusters the citations of different similar named entities. However, their method uses manually collected publications pages, where the correct publication pages are identified manually among the results returned by Google with a query consisting of the author name and "publication"as a keyword. The approach is evaluated on two names "J Anderson"(6) and "J Smith"(9) with accuracy of 70.6% and 73.6% respectively. The work has been improved further in [30] by using information about aliases and name invariants from a database. Two supervised learning approaches are proposed to disambiguate authors in the citations, one based on naive Bayes probability model, the other based on Support Vector Machines (SVMs). Both approaches utilize the same three types of citation attributes as previous work: co-author names, the title of the paper, and the title of the journal or proceeding and assume the existence of a citation database (training data) indexed by the canonical name entities, i.e., a name that is, the minimal invariant and complete name entity for disambiguation. Based on the observation that an author's citations usually contain the information of the author's research area and his or her individual patterns of coauthoring, the authors conjecture that a generative model like naive Bayes is a promising choice due to the fact that it can create other examples of the data and capture all authors' writing patterns. They also investigate the use of a discriminative model, such as Support Vector Machines, for this task. In the SVM approach citations are represented in a vector space and each author is considered as a class: a new citation is classified to the closest author. Unlike naive Bayes SVM can learn from both positive and negative training citations. Another difference between SVM and naive Bayes is that the first classifies a citation using distance measures while the second is based on probabilities. Experimental settings have been established as follows: given a full citation with the query name implicitly omitted the goal is to predict the most likely canonical name from the citation database. Two experimental datasets have been used: one collected from the web, the other collected from the DBLP citation databases. SVM outperforms nayve Bayes, except in the case of using coauthor information alone. The reason is that SVM can better capture and rank the features unique to a class, while naive Bayes assume all features to have the same distribution.

Mann et al. [31] present a set of algorithms for distinguishing personal names with multiple real referents in text, based on little or no supervision. The approach utilizes an unsupervised clustering technique over a rich feature space of biographic facts, which are automatically extracted via a language-independent bootstrapping process. The feature vectors for each document, have been generated using different methods, such as using all words (plain) or only Proper Nouns (nnp), using most relevant words (tf-idf), employing basic or extended biographical features. Performance is evaluated on a test set of hand-labeled multi-referent personal names and via automatically generated pseudonames.

Chen and Martin [28] attack the problem of automatically separating sets of news documents generated by queries containing personal names into coherent partitions. They propose clustering as a solution, with a range of syntactic and semantic features, beyond bag of words contextual features or biographical information. In particular they extract noun phrase features, named-entity feature, target entity, local entity (locally co-occurring entity with the target one), non-local entity. The proposed methodology follows a common architecture for named-entity disambiguation: the detection of ambiguous objects, feature extraction and representation, similarity matrix learning, and finally clustering. Evaluation over the Bagga and Baldwin corpus [27] and against their achieved results, shows an overall improved performance.

Pedersen et al. [32] make a parallel between name discrimination, the problem of grouping occurrences of a name based on the underlying entity's identity, and word sense discrimination, the process of examining a number of sentences that contain a given polysemous word, and then grouping those instances based on the meaning of that word (unlike word sense disambiguation, the process of assigning a sense to a polysemous word from a predefined set of possibilities). The authors present an unsupervised approach that resolves name ambiguity by clustering the instances of a given name into groups, each of which is associated with a distinct underlying entity. They use statistically significant bigrams that occur in the same context as the ambiguous name as features that represent the context of the ambiguous name. They generate a high dimensional "instance by word "matrix (by manipulating bigrams) and reduce it to its most significant dimensions by Singular Value Decomposition (SVD). The different "meanings"of a name are discriminated by clustering these second order context vectors with the method of Repeated Bisections. The proposed method has been evaluated over an ad-hoc built corpus of text containing ambiguous pseudo-names and proved to be more accurate than the majority classifier, and the best results are obtained by having a small amount of local context to represent the instance along with a larger amount of context for identifying features, or vice versa.

The problem of Entity Reference Resolution has been addressed from a graph based data analysis point of view by [8]. Usually knowledge about entities and the relationships among them resides in numerous documents and datasets distributed across a variety of data sources. Information Extraction has made possible to extract such entities and relationships automatically (at least in limited domains) which traditionally were manually collected from analysts. A key issue in constructing graphs from diverse sources is that of resolving the references to entities in these datasets. In the ideal world each entity would have a unique identifier and, whenever this entity is referred to, its unique identifier is specified so that there is no uncertainty. In the real world, however, entities are often referred to by descriptions that may be created by multiple persons and collected from various sources. Entities might be referred by different descriptions and also multiple entities might end up matching the same description. The authors compare two different kind of approaches to such problem: Feature Based Similarity (FBS) approach against their innovative Reference Disambiguation Approach (RDA). FBS uses features extracted from text while RDA builds a graph of entities and adopts a probabilistic model to estimate the strength of relationships between them. The authors use an undirected graph to represent entities and their relations.

The RDA algorithm consists of a Probabilistic Model (PM) over the graph that makes use of feature-based similarity and relationship-based similarity to calculate connection strength between two entities. Experiments have been carried out on the author matching problem, which considers authors and papers: the goal is authors disambiguation. Results over three datasets (one real and two synthetic) show high disambiguation accuracy.

Nuray and Turan [33] focus the attention on calibrating a connection strength (CS) measure from training data in the context of reference disambiguation problem, that is, identifying for each reference the unique entity it refers to. Given any two nodes u and v in the graph G, the connection strength c(u, v) returns how strongly u and v are interconnected in G. Generally, a domain expert determines a mathematical model to compute c(u, v), which describes the underlying dataset best. The authors propose a supervised learning algorithm that learns the importance of CS, among the classified entities and makes the approach self-tunable to any underlying domain. The algorithm uses a graphical methodology; the disambiguation decisions are made according to object features and inter-object relationships, including indirect ones that exist among objects. Experiments have been carried out on two synthetic datasets taken from two domains: Movies (from Stanford Movies Dataset) and Publications (CiteSeer dataset). Stanford Movies Dataset contains three different entity types: movies (11453 entities), studios (992 entities) and people (22121 entities) and five types of relationships: actors, directors, producers, producing studios and distributing studios. CiteSeer dataset contains four different types of entities: author, paper, department, and organization and three types of relationships: author-paper, author-department, and department-organization. The authors introduce uncertainty in both datasets manually. Results are expressed in terms of accuracy and show an increase of the quality of the disambiguation technique, compared to the state of the art Random-Walk model.

Malin [34] investigates unsupervised methods which simultaneously learn the number of entities represented by a particular name and observations corresponding to the same entity. The disambiguation methods leverage the fact that an entity's name can be listed in multiple sources, each with a number of related entity's names, which permits the construction of name-based relational networks. The author proposes two different methods which differ with respect to the type of network similarity exploited for disambiguation. The first method relies upon exact name similarity and employs hierarchical clustering of sources, where each source is considered a local network and represented as a boolean vector $s_i = [e_{i1}, \dots, e_{in}]$, $e_{ij} = 1$ if e_j is in source s_i and 0 otherwise. In contrast, the second method employs a less strict similarity requirement by using random walks between ambiguous observations on a global social network constructed from all sources, or a community similarity. The graph has been built with putting a node for every distinct name in S and an edge between two nodes if the names collocate in a source at least one time. Experiments have been conducted on a subset of the Internet Movie Database. Results demonstrate that community equivalence (the second method) provides an advantage over exact equivalence (first method) for measuring similarity and, subsequently, disambiguation.

Graph-based models have also been applied to Person Name Disambiguation, usually benefiting from the social networks in people-related tasks. Minkov et al.[7] consider extended similarity metrics for documents and other objects embedded in graphs, implemented via a lazy graph walk. They provide an instantiation of this framework for email data, where content, social networks and a timeline are integrated in a structural graph. The suggested framework is evaluated for two email-related problems: disambiguating names in email documents, and threading. Resolving the referent of a person name is also an important complement to the ability to perform named entity extraction for tasks like social network analysis or studies of social interaction in email. The authors model this problem as a search task: based on a name-mention in an email message m, they formulate query distribution V_q, and then retrieve a ranked list of person nodes. Experiments carried out on the Cspace corpus [35], manually annotated with personal names, show that reranking schemes based on the graph-walk similarity measures often outperform baseline methods, with a maximum accuracy of 83.8%.

The main differences between these method and the one we propose within this work is that our method is applicable to all kind of proper names because it does not rely on resources such as social networks and relatedness scores can be used offline after they have been calculated.

Semantic relatedness between words or concepts measures how much two words or concepts are related by encompassing all kinds of relations between them, such as hypernymy, hyponymy, antonymy and functional relations. There is plenty of literature on computing semantic relatedness between words or concepts using knowledge extracted from Wikipedia, such as [18] and [36]. However, the main limitation of these methods is that they only make use of one or two types of features; and they generally adapt WordNet-based [37, 20, 38] approaches by employing similar types of features extracted from Wikipedia. In contrast, we believe that other information content and structural elements in Wikipedia can be also useful for the semantic relatedness task; and that combining various features in an integrated model in the semantic relatedness task is crucial for improving performance. For this reason, we propose a random graph walk model based on a combination of features extracted from Wikipedia for computing semantic relatedness.

3. Methodology. Given a set of *surfaces* and their corresponding concept relatedness matrix R, our NED algorithm returns for each *surface* one *sense* (*concept*), collectively determined by other *surfaces* and their corresponding *concepts*. To achieve this goal the proposed method performs four main sequential steps: 1) each text is reduced to the list of *surfaces* of appearing entities; 2) for each *surface*, Wikipedia is used to retrieve all its possible *meanings* (also denoted as *concepts*) and build a feature space for each of them; 3) all *concepts*, their features and relations are transformed into a graph representation: a random graph walk model is then applied to combine the effects of features and derive a relatedness score; 4) for each *surface* a single *meaning* is chosen, taking into account *semantic relation* within the entity graph.

3.1. Concept Retrieval. In more details, as a starting point for the proposed methodology we assume that each text has been reduced to the list of its contained *named entity surfaces*, as it is simply obtainable with a standard NER system as Yamcha [39]. Then for each *surface* Wikipedia is used to retrieve all its possible *meanings* and build a feature space for each of them. More precisely we query Wikipedia using *surface* to retrieve relevant pages. If a *surface* matches an entry in Wikipedia, a page will be returned. If the *surface* has only one sense defined in Wikipedia then we have a single result: the page describing the concept that matches the surface form. We refer to this page as the *sense page* for the concept. Alternatively a *disambiguation page* may be returned if the *surface* has several *senses* defined in Wikipedia. Such a page lists different senses as links to other pages and with a short description for each one. For the purpose of this work, we deliberately choose the disambiguation page for every *surface*, which means we query Wikipedia by adding the string "(disambiguation)" to the surface words and follow every link on the page and keep all *sense pages* for that surface. This is done by appending the keyword "(disambiguation)" to a *surface* as a query. Thus, for every *surface*, we obtain a number of *concepts* (represented as *sense pages*) as input to our disambiguation algorithm. Once we have identified relevant *concepts* and their *sense pages* for the input *concept surface forms*, we use the *sense page* retrieved from Wikipedia for each *concept* to build its feature space. We identify the following features that are potentially useful:

1. Words composing the titles of a page *(title_words)*: words in the title of a sense page; plus words from all its redirecting links in Wikipedia (different *surfaces* for the same concept).
2. Top *n* most frequently used words in the page *(frequent_words_n)*: prior work makes use of words extracted from the entire page [18], or only those from the first paragraph [36]. In our work, we use the most frequent words, based on the intuition that word frequency indicates the importance of the word for representing a topic.
3. Words from categories *(cat_words)* assigned to the page: each page in Wikipedia is assigned several category labels. These labels are organized as a taxonomy. We retrieve the category labels assigned to a page by performing a depth limited search of 2, and split these labels to words.
4. Words from outgoing links on the page *(link_words)*: the intuition is that links on the page are more likely to be relevant to the topic, as suggested by Turdakov and Velikhov [40].

Thus, for each *concept*, we extract above features from its sense page, and transform the text features into a graph conforming to the random walk model, which is used to compute *semantic relatedness* between meanings belonging to different surfaces.

3.2. Random Graph Walk Model. A random walk is a formalization of the intuitive idea of taking successive steps in a graph, each in a random direction [41]. Intuitively, the harder it is to arrive at a given node starting from another, the less related the two nodes are. The advantage of a random-walk model concerns on seamlessly combining different features to arrive at one single measure of relatedness between two entities [42]. Specifically, we build an undirected weighted typed graph that encompasses all concepts identified in the page retrieval step and their extracted features. We build the graph following Definition 2.

Definition 2. *The graph is a 5-tuple $G = (V, E, t, l, w)$, where:*

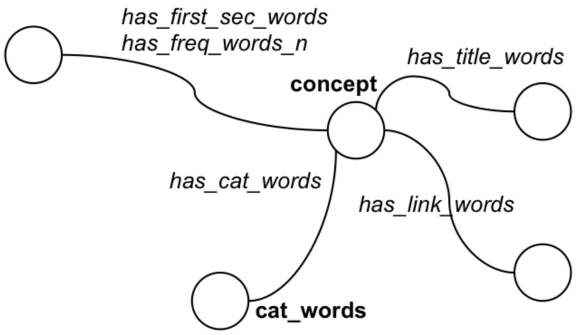

Fig. 1. The Graph representation model of concepts, features, and their relations. Circles indicate nodes (V) representing concepts and features; bold texts indicate types (T) of nodes; solid lines connecting nodes indicate edges (E), representing relations between concepts and features; italic texts indicate types (L) of edges. Different concepts may share features, enabling walks on the graph

V *is the set of nodes representing the concepts and their features;*

$E = V \times V$ *is the set of edges that connect concepts and their features, representing an undirected path from concepts to their features, and vice versa;*

$t\colon V \rightarrow T$ *is the node type function*
$T = \{t_1, \ldots, t_{|T|}\}$ *is a set of types (e.g., concepts, title_words, cat_words, ...);*

$l\colon E \rightarrow L$ *is the edge label function*
$L = \{l_1, \ldots, l_{|L|}\}$ *is a set of labels that define relations between concepts and their features;*

$w\colon L \rightarrow R$ *is the label weight function that assigns a weight to an edge.*

Figure 1 shows an abstract piece of the graph with types and labels described within Defintition 2, while Figure 2 shows a small portion of graph for the concept Paris, the capital of France. The concept node contains the Wikipedia URI for Paris, capital of France; other nodes contain information extracted from links, categories, descriptions and so on.

We define weights for each edge type, which, informally, determine the relevance of each feature to establish the relatedness between any two concepts.

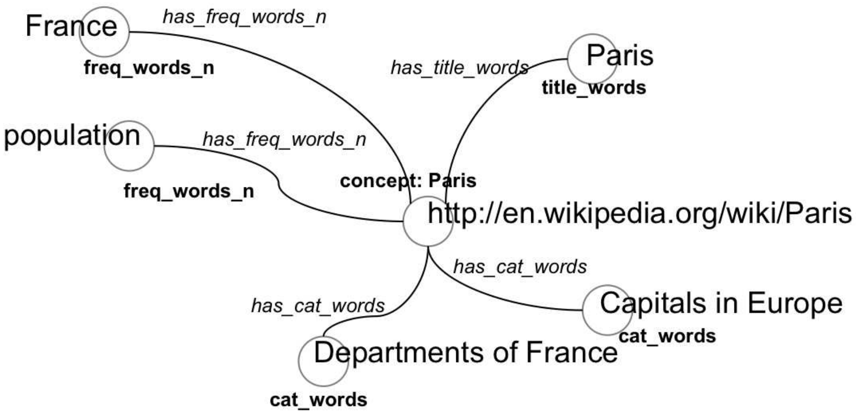

Fig. 2. A piece of graph for the concept Paris, capital of France

Let $L_{t_d} = l(x,y)\colon (x,y) \in E \cap T(x) = t_d$ be the set of possible labels for edges leaving nodes of type t_d. We require that the weights form a probability distribution over L_{t_d}, i.e.

$$\sum_{l \in L_{t_d}} w(l) = 1.$$

We build an adjacency matrix of locally appropriate similarity between nodes as

$$W_{ij} = \begin{cases} \sum\limits_{l_k \in L} \frac{w(l_k)}{|(i,\cdot)\in E\colon l(i,\cdot)=l_k|} & (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where W_{ij} is the i^{th}-row and j^{th}-column entry of W, indexed by V. The above equation distributes uniformly the weight of edges of the same type leaving a given node. The *weight model* (*wm*), that is, weights associated to each type of edges in the graph, has been determined applying a simulated annealing method [43]. The algorithm explores the search space of all possible combinations of feature weights and iteratively reduces the difference between a gold standard solution and that of our system. The algorithm allows us to run our method on one dataset in an iterative manner, where in each iteration, the algorithm generates a random *wm* for the feature set and scores our system results obtained with that model against the gold standard. If a *wm* obtained in the current iteration produces better results than the previous, the simulated annealing algorithm will attempt to adjust weights based on that model in iterations. Thus, by running simulated annealing for relatively large number of iterations, we expect the system performance to converge; by which we obtain the final optimum weight model for that feature set. This tuning has been done in advance using a standard testing dataset for semantic relatedness, the WordSimilarity-353 Test Collection [44], and empirically derived the optimum weight model for our chosen feature set.

To simulate the random walk, we apply matrix transformation using the formula $P^{(t)}(j \mid i) = [(D^{-1}W)^t]_{ij}$, as described by Iria et al. in [42], where D is the diagonal degree matrix given by $D_{ii} = \sum_k W_{ik}$, and t is a parameter representing the number of steps of the random walk. In our work, we have set $t = 2$ in order to compute the relatedness for walks that start in a concept and traverse one feature to arrive at another concept. Unlike PageRank [13], we are not interested in the stationary behavior of the model. The resulting matrix of this transition $P^{(t)}(j \mid i)$ is a sparse, non-symmetric matrix filled with probabilities of reaching node i from j after t steps. To transform probability to relatedness, we use the observation that the probability of walking from i to j then coming back to i is always the same as starting from j, reaching i and then coming back to j. Thus we define a transformation function as:

$$Rel(i \mid j) = Rel(j \mid i) = \frac{P^{(t)}(j \mid i) + P^{(t)}(i \mid j)}{2} \tag{2}$$

and we normalize the score to range {0, 1} using:

$$Rel(i \mid j) = \frac{Rel(j \mid i)}{\max Rel(j \mid i)}. \tag{3}$$

We will give a semplified example of relatedness score. Let us assume to have the following text: *"Robert Taylor went to Paris and then to Canada"*. In this small text we can spot three concept surfaces: Robert Taylor (**c1**), Paris (**c2**), Canada (**c3**).

Assuming that each of them has 2 possible meanings, we will use the following abbreviations:

- **c1.1**: Robert Taylor, actor
  c1.2: Robert Taylor, computer scientist
- **c2.1**: Paris, France
  c2.2: Paris, Virginia
- **c3.1**: Canada, country in northern North America
  c3.2: Canada, New France

Given the above six concepts we calculate the relatedness scores indicated in Table 1 (note that scores do not indicate actual values). Relatedness scores

Table 1. An example of relatedness score between the concepts c1.1, c1.2, c2.1, c2.2, c3.1, c3.2 associated to the surfaces c1, c2 and c3

	c1.1	**c1.2**	**c2.1**	**c2.2**	**c3.1**	**c3.2**
c1.1	—	—	0.7	0.3	0.1	0.8
c1.2	—	—	0.5	0.2	0.6	0.2
c2.1	0.7	0.5	—	—	0.9	0.5
c2.2	0.3	0.2	—	—	0.1	0.6
c3.1	0.1	0.6	0.9	0.1	—	—
c3.2	0.8	0.2	0.5	0.6	—	—

are calculated between concepts belonging to different surfaces (e.g. relatedness between Paris_France and Paris_Virginia is not to be calculated).

3.3. Named Entity Disambiguation. The final step of the methodology consists of choosing a single meaning (*concept*) for each *entity surface*, exploiting *semantic relatedness* scores derived by the graph. Given $S = \{s_1, \ldots, s_n\}$ the set of *surfaces* in a document, $C = \{c_{1_k}, \ldots, c_{m_k}\}$ (with $k = 1 \cdots \mid S \mid$), the set of all their possible senses (*concepts*) and R the matrix of relatedness $Rel(i \mid j)$ with each cell indicating the strength of relatedness between concept c_{i_k} and concept $c_{j_{k'}}$ (where $k \neq k'$, that is, c_{i_k} and $c_{j_{k'}}$ have different surface forms), we define the *entity disambiguation algorithm* as a function $f : S \rightarrow C$, given a set of *surfaces* S returns the list of disambiguated concepts, using the concept relatedness matrix R. We define different kind of such functions f and compare results in Section 4.

As first and simple disambiguation function we define the ***highest method***: we build the list of candidates winner concepts for each s_k surface in the text, $cand_{k_i}$, with i being the candidate concept for *surface* s_k ($k = 1 \cdots \mid S \mid$); if some of *surfaces* s_k has more that one candidate winner, for eachs_k *surface* with multiple i values, we simply pick the *concept* that among the candidates has the highest value in the matrix R.

The ***combination method*** calculates for each concept c_{i_k} the sum of relatedness with all different concepts $c_{j_{k'}}$ from different surfaces (such as $j \neq i$, $k' \neq k$). Given $V = \{v_1, \ldots, v_{|C|}\}$ the vector of such values, the function returns for each surface s_k the concept c_{i_k} with max v_i.

The ***propagation method*** works as follows: taking as seed the highest similarity value in the matrix R we fix the 2 concepts i and j giving that value:

for their surface form k and k' we delete rows and columns in the matrix R coming from other concepts for the same surfaces (all c_{t_k} and $c_{t_{k'}}$ with $t \neq i$ and $t \neq j$). This step is repeated recursively, picking next highest value in R. The stop condition consists of having one concept row in the matrix R for each surface form.

In the following section we present our experiments and evaluation.

4. Experiments. We performed the experiment with an "in vitro evaluation", which consists of testing systems independently of any application, using specially constructed benchmarks. What we want to prove is that the use of *semantic relatedness* scores is profitable for the issue of NED and that the graph of interconnections between entities is influent for the disambiguation decision. As benchmark to test our system we used data provided by Cucerzan in [6], which is publicly available[8]. Test data consist of two different datasets. Each dataset constists of several documents containing a list of Named Entites, labelled with the corresponding page in Wikipedia. As described in Section 3 we retrieve concepts for each surface and we build a graph with all identified possible concepts for each text. After running the Random Walk on the built graph and transforming the transition matrix in a relatedness matrix we obtain an upper triangular matrix with a score of relatedness between different concepts, belonging to different surfaces.

The first dataset we used for experiments, referred in what follows as *NEWS*, consists 20 news stories: for each story is provided the list of all entities, annotated with the corresponding page in Wikipedia. The number of entities in each story can vary form 10 to 50. Some of the entities have a blank annotation, because they do not have a corresponding page in the Wikipedia collection: among all the identified entities, 370 are significantly annotated in the test data. As input for our system we started from the list of entities spotted in the benchmark data and for each entity the list of all possible meaning is retrieved, e.g., for surface "Alabama" following concepts are retrieved:

Alabama $\longrightarrow [AlabamaClaims \mid Genus \mid CSSAlabama \mid AlabamaRiver \mid Alabama(people) \mid Noctuidae \mid Harvest(album) \mid USSAlabama \mid Alabamalanguage \mid Alabama(band) \mid Moth \mid UniversityofAlabama \mid Alabama, NewYork]$

The second dataset, referred in what follows as *WIKI*, consists of 350 Wikipedia entity pages selected randomly. The text articles contain 5,812 entity

[8] `http://research.microsoft.com/users/silviu/WebAssistant/TestData`

surface forms. We performed our evaluation on 3,136 entities, discarding all non-recallable surfaces, that is, all those surfaces having no correspondance in the Wikipedia collection. Indeed, an error analysis carried out by Cucerzan on this dataset showed that it contains many surface forms with erroneous or out-of-date links, as reported in [6].

We evaluate performance in terms of accuracy, that is, the ratio of number of correctly disambiguated entities on total number of entities to disambiguate. Results obtained applying all defined disambiguation functions to the relatedness matrix are shown in Table 2 for the *NEWS* dataset and in Table 3 for the *WIKI* dataset. Both tables also report figures obtained by Cucerzan on the same datasets [6].

In the first experiment the best result equals the best available system, with an accuracy of 91.5% (slightly higher) and all proposed functions are above the baseline of 51.7% (baseline returns always the first available result). Between three proposed methods, the *combination method* obtained the best result, equalling the best available system. The *highest method* achieves results below the state of the art of 91.4%, but with an accuracy of 82.2% is far over the baseline of 51.7% (baseline returns always the first available result). The motivation can be that it takes into account only the best relatedness score for each concept to decide sense assignment, without considering the rest of the scores. The *propagation method* works even worse because adds to the disadvantage of the first one also the propagation of errors. It reaches an accuracy of 68.7%, which is in the middle between the baseline and the state of the art.

Table 2. Comparison of proposed Named Entity Disambiguation Functions on *NEWS* dataset

Literature Systems	Accuracy	Function	Accuracy
Cucerzan baseline [6]	51.7%	Highest	82.2%
Cucerzan [6]	**91.4%**	**Combination**	**91.5%**
		Propagation	68.7%

To consolidate this result we conducted the same experiment on the *WIKI* dataset.

The second experiment definitively confirms the trend reported in the first one. The three proposed disambiguation functions have the same behavior on the *WIKI* dataset. The *combination method* is the best one: it achieves an accuracy of 89.8%, outperforming the accuracy of 88.3% reported by the state-of-the art system. The *highest method* is between the baseline and the state-of-the-art

Table 3. Comparison of proposed Named Entity Disambiguation Functions on *WIKI* dataset

Literature Systems	Accuracy	Function	Accuracy
Cucerzan baseline [6]	86.2%	Highest	87.1%
Cucerzan [6]	**88.3%**	**Combination**	**89.8%**
		Propagation	84.3%

system, with an accuracy of 87.1%. The *propagation method* is the worst one: with an accuracy of 84.3 % is under the baseline of 86.2%.

As expected and already assessed in the *NEWS* experiment, the *combination method* performs much better than others, outperforming the state of the art system on the *WIKI* dataset. The motivation can be found in the fact that it considers relatedness scores in their entirety, thus taking into account the interaction of all concepts instead of couples of concepts. We consider such value as an encouraging outcome for the proposed novel method: the second experiment reinforces results of the first one and affirms the correctness of the proposed methodology.

5. Conclusions. In this work we proposed a novel method for Named Entity Disambiguation. Experiments showed that the paradigm achieves significant results: the overall accuracy is 91.5% and 89.8% on two different datasets, which is a result competitive to the state of the art. The successful accuracy reached hints at the usefulness of *semantic relatedness* measures for the process of NED.

The decision of using Wikipedia as *entity inventory* imposes some limitations in terms of lexical coverage, especially with reference to specific domains. Specific technical areas, such as Medicine, Biology, specific fields of engineering, etc., can have little or no coverage within the Wikipedia source. However, the methodology is still applicable when the *entity inventory* is a different Knowledge Base: the only requirement is that we have coverage for entities of interest and that it is possible to calculate relatedness scores between them. We could easily change Wikipedia with a domain ontology, obtaining relatedness scores and then applying the same proposed NED step. Theoretically we might expect this to be true but obviously, experiments are needed to prove the efficacy of such approach and future work can follow this direction. Also, as future work, we plan to design new disambiguation functions over the relatedness matrix to achieve better results.

REFERENCES

[1] Agirre E., D. Martínez, O. López de Lacalle, A. Soroa. Two graph-based algorithms for state-of-the-art WSD. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, Association for Computational Linguistics, July, 2006, 585–593.

[2] Mihalcea R. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing HLT/EMNLP 2005, The Association for Computational Linguistics, 2005, 411–418.

[3] Navigli R., Lapata M. Graph connectivity measures for unsupervised word sense disambiguation. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence IJCAI 2007 (Ed. M. M. Veloso), 2007, 1683–1688.

[4] Sinha R., R. Mihalcea. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In: Proceedings of the First IEEE International Conference on Semantic Computing ICSC 2007, IEEE Computer Society, 2007, 363–369.

[5] Bunescu R. C., M. Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In: Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics EACL 2006, The Association for Computer Linguistics, 2006, 9–16.

[6] Cucerzan S. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning EMNLP-CoNLL, Prague, Czech Republic, Association for Computational Linguistics, June, 2007, 708–716.

[7] Minkov E., W. W. Cohen, A. Y. Ng. Contextual search and name disambiguation in email using graphs. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR 2006, (Eds E. N. Efthimiadis, S. T. Dumais, D. Hawking, K. Järvelin), ACM, Seattle, Washington, USA, 2006, 27–34.

[8] Kalashnikov D. V., S. Mehrotra. A probabilistic model for entity disambiguation using relationships. 2005, 1–22.

[9] Grishman R., B. Sundheim. Message Understanding Conference-6: A Brief History. In: Proceedings of the 16th International Conference on Computational Linguistics COLING 1996, Center for Sprogteknologi, Copenhagen, Denmark, August 5–9, 1996, 466–471.

[10] Peng Y., D. He, M. Mao. Geographic named entity disambiguation with automatic profile generation. In: Web Intelligence, IEEE Computer Society, 2006, 522–525.

[11] Aswani N., K. Bontcheva, H. Cunningham. Mining information for instance unification. In: Proceedings of the International Semantic Web Conference, Lecture Notes in Computer Science, Vol. **4273**, Springer, 2006, 329–342.

[12] García N. F., J. M. B. del Toro, L. Sánchez, A. Bernardi. IdentityRank: Named Entity Disambiguation in the Context of the NEWS Project. In: ESWC.(Eds E. Franconi, M. Kifer, W. May), Lecture Notes in Computer Science, Vol. **4519**, Springer, 2007, 640–654.

[13] Brin S., L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, **30** (1998), No 1–7, 107–117.

[14] Hassell J., B. Aleman-Meza, I. B. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. In: Proceedings of the International Semantic Web Conference, Lecture Notes in Computer Science,Vol. **4273**, Springer, 2006, 44–57.

[15] Nguyen H. T., T. H. Cao. Named entity disambiguation on an ontology enriched by Wikipedia. In: Proceedings of RIVF 2008, IEEE, 2008, 247–254.

[16] Harris Z. Distributional structure. *Word*, **10** (1954), No 23, 146–162.

[17] Ponzetto S. P., M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, New York City, USA, Association for Computational Linguistics, June, 2006, 192–199.

[18] Strube M., S. P. Ponzetto. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence, AAAI Press, 2006, 1419–1424.

[19] Zesch T., I. Gurevych, M. Mühlhäuser. Analyzing and Accessing Wikipedia as a Lexical Semantic Resource. In: Proceedings of the Biannual Conference of the Society for Computational Linguistics and Language Technology, 2007, 213–221.

[20] Leacock C., M. Chodorow. Combining local context and WordNet similarity for word sense identification. In: WordNet: An Electronic Lexical Database (Ed. C. Fellbaum), MIT Press, 1998, 265–283.

[21] Toral A., R. Munoz. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In:Proceedings of the Workshop on New Text, 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, April, 2006, 56–61.

[22] Kazama J., K. Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic, 2007, 698–707.

[23] Vercoustre A. M., J. A. Thom, J. Pehcevski. Entity ranking in Wikipedia. In: Proceedings of the 2008 ACM Symposium on Applied Computing – SAC (Eds R. L. Wainwright, H. Haddad), ACM, Fortaleza, Ceara, Brazil 2008, 1101–1106.

[24] Han X., J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge.In: Proceeding of the 18th ACM conference on Information and knowledge management CIKM '09, New York, NY, USA, ACM, 2009, 215–224.

[25] Sugiyama K., M. Okumura. Personal name disambiguation in web search results based on a semi-supervised clustering approach. In: ICADL. (Eds D. H. L. Goh, T. H. Cao, I. Sølvberg, E. M. Rasmussen), Lecture Notes in Computer Science, Vol. **4822**, Springer, 2007, 250–256.

[26] Srinivasan H., J. Chen, R. Srihari. Cross document person name disambiguation using entity profiles. In: Proceedings of the IJCAI 2009 Workshop on Information Integration on the Web, Pasadena, California, 2009.

[27] Bagga A., B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In: Proceedings of the 17th international conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics, 1998, 79–85.

[28] Chen Y., J. Martin. Towards robust unsupervised personal name disambiguation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, Association for Computational Linguistics, June, 2007, 190–198.

[29] Han H., H. Zha, C. L. Giles. A model-based k-means algorithm for name disambiguation. In: Proceedings of the 2nd International Semantic Web Conference ISWC-03, Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, 2003.

[30] Han H., C. L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In: JCDL (Eds H. Chen, H. D. Wactlar, C. chih Chen, E. P. Lim, M. G. Christel), ACM, 2004, 296–305.

[31] Mann G. S., D. Yarowsky. Unsupervised personal name disambiguation. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, Morristown, NJ, USA, Association for Computational Linguistics, 2003, 33–40.

[32] Pedersen T., A. Purandare, A. Kulkarni. Name discrimination by clustering similar contexts. In:CICLing (Ed. A. F. Gelbukh), Lecture Notes in Computer Science, Vol. **3406**, Springer, 2005, 226–237.

[33] Nuray-Turan R., D. V. Kalashnikov, S. Mehrotra. Self-tuning in graph-based reference disambiguation. In: DASFAA (Eds K. Ramamohanarao, P. R. Krishna, M. K. Mohania, E. Nantajeewarawat), Lecture Notes in Computer Science, Vol. **4443**, Springer, 2007, 325–336.

[34] Malin B. Unsupervised name disambiguation via social network similarity. In: Proceedings of the Workshop on Link Analysis, Counterterrorism and Security, 2005, Newport Beach, California, USA, 93–102.

[35] Minkov E., R. C. Wang, W. W. Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, Association for Computational Linguistics, October 2005, 443–450.

[36] Zesch T., C. Müller, I. Gurevych. Using wiktionary for computing semantic relatedness. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence AAAI 2008 (Eds D. Fox, C. P. Gomes), AAAI Press, 2008, 861–866.

[37] Banerjee S., T. Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence IJCAI-03 (Eds G. Gottlob, T. Walsh), M. Kaufmann, 2003, 805–810.

[38] Resnik P. Disambiguating noun groupings with respect to WordNet senses. In: Proceedings of the 3th Workshop on Very Large Corpora, ACL,Cambridge, Massachusetts, USA, 1995, 54–68.

[39] Kudo T., Y. Matsumoto. Fast Methods for Kernel-Based Text Analysis. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (Eds E. Hinrichs, D. Roth), Prague, Czech Republic, 2003, 24–31.

[40] Turdakov D., P. Velikhov. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. In: SYRCoDIS (Eds Kuznetsov S. D., P. Pleshachkov, B. Novikov, D. Shaporenkov), CEUR Workshop Proceedings. Vol. **355**, 2008. `http://www.CEUR-WS.org`

[41] Lovász L. Random walks on graphs: A survey. *Combinatorics*, **2** (1996), 353–398.

[42] Iria J., L. Xia, Z. Zhang. Wit: Web people search disambiguation using random walks. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, ACL, 2007, 480–483.

[43] Nie Z., Y. Zhang, J. Wen, W. Ma. Object-level ranking: bringing order to web objects. In: Proceedings of the 14th international conference on World Wide Web WWW '05, New York, NY, USA, ACM, 2005, 567–574.

[44] Finkelstein L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, E. Ruppin. Placing search in context: the concept revisited. *ACM Transactions on Information Systems.* **20** (2002), No 1, 116–131.

Anna Lisa Gentile, Ziqi Zhang
Department of Computer Science, University of Sheffield, UK
email: `a.gentile@shef.ac.uk, z.zhang@dcs.shef.ac.uk`

Lei Xia
Archaeology Data Service, University of York, UK
email: `lx535@york.ac.uk`

José Iria
IBM Research – Zurich, Switzerland
email: `jir@zurich.ibm.com`

Received February 28, 2010
Final Accepted May 31, 2010