

Semantic Relatedness Approach for Named Entity Disambiguation

Anna Lisa Gentile^{1,*}, Ziqi Zhang², Lei Xia^{3,**}, and José Iria^{4,**}

¹ Department of Computer Science, University of Bari, Italy
al.gentile@di.uniba.it

² Department of Computer Science, The University of Sheffield, UK
{z.zhang,l.xia}@dcs.shef.ac.uk

³ Archaeology Data Service, University of York, UK
lx535@york.ac.uk

⁴ IBM Research - Zurich, Switzerland
jir@zurich.ibm.com

Abstract. Natural Language is a mean to express and discuss about concepts, objects, events, i.e., it carries semantic contents. One of the ultimate aims of Natural Language Processing techniques is to identify the meaning of the text, providing effective ways to make a proper linkage between textual *references* and their *referents*, that is, real world objects. This work addresses the problem of giving a sense to *proper names* in a text, that is, automatically associating words representing *Named Entities* with their *referents*. The proposed methodology for Named Entity Disambiguation is based on Semantic Relatedness Scores obtained with a graph based model over Wikipedia. We show that, without building a *Bag of Words* representation of the text, but only considering named entities within the text, the proposed paradigm achieves results competitive with the state of the art on two different datasets.

1 Introduction

Reading a written text implies the comprehension of the information that words are carrying. Comprehension is an intrinsic capacity for a human, but not for a machine. Providing machines with such ability, by anchoring meanings to words, can be considered a task with great significance for Artificial Intelligence.

The focus of this work is on proper names, that is, on such words within text that represent *entites*: we want to give a meaning to such pieces of text that carry high information potential. Many tasks could benefit of such added value, such as Information Retrieval (for example, lots of Web Search queries concern Named Entities).

We propose an automatic method to associate a *unique sense* (the *referent*, that will be also addressed in the remaining of this work as *meaning*, *concept* or simply *sense*) to each *entity* (the *reference* within the text), exploiting Wikipedia¹ as freely available

* The work was done while this author was at The University of Sheffield, UK as visiting researcher.

** The work was done while this author was working at The University of Sheffield, UK.

¹ <http://en.wikipedia.org/wiki/Wikipedia>

Knowledge Base, showing a novel solution for Named Entity Disambiguation (NED). We show the correctness of the proposed methodology with two experimental sessions, consolidating results obtained in [1].

Our contributions are twofold. Firstly, we use a random-walk model based Semantic Relatedness approach to NED. Graph-based models have previously been applied to Word Sense Disambiguation (WSD) [2,3,4,5] but not experimented for the problem of NED: to the best of our knowledge, previous approaches to NED were based on Vector Space model, treating *concepts* and context texts as a bag of words [6,7], while graph-based models have been exploited for a specific type of NED, which is Person Name Disambiguation [8], or for specific domains, such as bibliographic citations [9]. The solution proposed in this work exploits Semantic Relatedness Scores (calculated with a random walk on a graph) as input for disambiguation step. Secondly, we introduce a different way for representing the context for the target entity which, rather than consisting of surrounding words, is composed of only other named entities present in the text. Our approach has the advantage of using relatedness scores independently for the NED task, that means using semantic relations as input for NED: this is useful in terms of clearly separation of two different functions, which can be refined and ameliorated separately one from each other. Compared to the best result by Cucerzan [7], which is an accuracy of 91.4% and of 88.3%, our method achieves a competitive accuracy of 91.5 % and 89.8% respectively and it adds the benefit of having two clearly separate steps (relatedness scores, disambiguation), thus providing a glimpse of improving results in both directions.

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

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) [10]. The NED process aims to create a mapping between the *surface form* of an entity and its unique dictionary meaning. It can be assumed to have a dictionary of all possible entity entries. In this work we use Wikipedia as such a dictionary. Many studies that exploit Wikipedia as a knowledge source have recently emerged [11,12,13]. 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 [14] that, resembling more to a dictionary, has little coverage on named entities [12]. Lots of previous works exploited Wikipedia for the task of NER, e.g., to extract gazetteers [15] or as an external knowledge of features to use in a Conditional Random Field NER-tagger [16], to improve entity ranking in the field

of Information Retrieval [17]. 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 [6] and Cucerzan [7]. 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% [6]. 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. The best result for this approach is an accuracy of 91.4% [7]. Both described works 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.

Another method for NED based on Wikipedia is the WibNED algorithm [18]. This method is an adaptation of Lesk dictionary-based WSD algorithm [19], with the difference that in the WibNED algorithm the words to disambiguate are only those representing an entity. WibNED takes as input a document $d = \{w_1, \dots, w_j, e_{j+1}, w_{j+2}, \dots, w_h, e_{h+1}, w_{h+2}, \dots\}$ and returns a list of Wikipedia URIs $X = \{s_1, s_2, \dots, s_k\}$ in which each element s_i is obtained by disambiguating the *target entity* e_i on the ground of the information obtained from Wikipedia for each candidate URI (Wikipedia page content of the URI) and words in the *context* C of e_i . The *context* C of the target entity e_i is defined to be a window of n words to the left and another n words to the right, for a total of $2n$ surrounding words. If other entities occur in the context of the target entity, they are considered as words and not as entities. The main limitation of this approach is that it is merely based on word overlapping and there is no specific weight associated to different categories of words in the context (entities and common words are treated the same way).

Contrary to these works, we propose a novel method, which uses a graph model combining multiple features extracted from Wikipedia. We calculate Semantic Relatedness over this graph and we exploit obtained relatedness values to resolve the problem of NED.

Graph-based models have been applied to a subtask of NED, which is Person Name Disambiguation, usually benefiting from the social networks in people related tasks. Minkov et al. [8] 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 [20], 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 this 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 that 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 a large number of literature on computing semantic relatedness between words or concepts using knowledge extracted from Wikipedia, such as [12] and [21]. 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 approaches [22,14,23] 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*), that is, 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 e.g Yamcha [24]. 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 [12], or only those from the first paragraph [21] 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 [25].

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 [26]. 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 lies at its strength of seamlessly combining different features to arrive at one single measure of relatedness between two entities [27]. Specifically, we build an undirected weighted typed graph that encompasses all concepts identified in the page retrieval step and their extracted features. The graph is a 5-tuple $G = (V, E, t, l, w)$, where 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; $t: V \rightarrow T$ is the node type function ($T = \{t_1, \dots, t_{|T|}\}$ is a set of types, e.g., concepts, *title_words*, *cat_words*, ...),

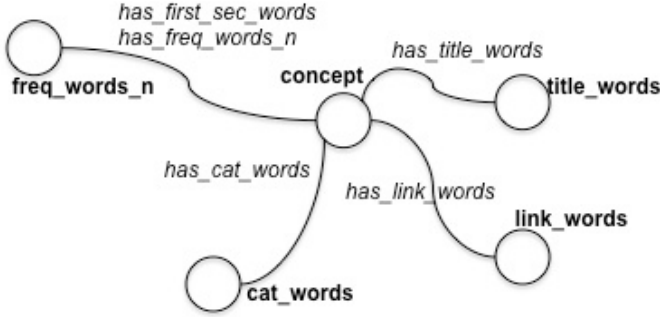


Fig. 1. The Graph representation model of concepts, features, and their relations. Circles indicate nodes (V) representing concepts and features; bold texts indicate types (T) of nodes; solid lines connecting nodes indicate edges (E), representing relations between concepts and features; italic texts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

$l: E \rightarrow L$ is the edge label function ($L = \{l_1, \dots, l_{|L|}\}$ is a set of labels that define relations between concepts and their features), and $w: L \rightarrow R$ is the label weight function that assigns a weight to an edge. Figure 1 shows a piece of the graph with types and labels described before. Concepts sharing same features will be connected via the edges that connect features and concepts.

We define weights for each edge type, which, informally, determine the relevance of each feature to establish the relatedness between any two concepts. Let $L_{t_d} = l(x, y): (x, y) \in E \cap T(x) = t_d$ 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

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

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 [28]. 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 next 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 [29], and empirically derived the optimum weight model for our chosen feature set.

To simulate the random walk, we apply matrix transformation using the formula $P^{(t)}(j | i) = [(D^{-1}W)^t]_{ij}$, as described by Iria et al. in [27], 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 [30], we are not interested in the stationary behavior of the model. The resulting matrix of this transition $P^{(t)}(j | i)$ 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:

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

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

$$Rel(i | j) = \frac{Rel(j | i)}{\max_k Rel(j | i)} \quad (3)$$

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 $S = \{s_1, \dots, s_n\}$ the set of *surfaces* in a document, $C = \{c_{1_k}, \dots, c_{m_k}\}$ (with $k = 1 \dots |S|$), the set of all their possible senses (*concepts*) and R the matrix of relatedness $Rel(i | j)$ 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 $f : S \rightarrow C$, that 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 $cand_{k_i}$ the list of candidates winner concepts for each surface, with i being the candidate concept for *surface* k ($k = 1 \dots |S|$); if some of *surfaces* k has more that one candidate winner, for each 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 $V = \{v_1, \dots, v_{|C|}\}$ the vector of such values, the function returns for each surface s_k the concept c_{i_k} that has the 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 usage 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 [7], which is publicly available². Test data consist of two different datasets. Each dataset consists of several documents containing a list of Named Entities, 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, that we will refer in what follows as *NEWS*, consists 20 news stories: for each story it is provided the list of all entities, annotated with the corresponding page in Wikipedia. The number of entities in each story can vary from 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* | *Genus* | *CSSAlabama* | *AlabamaRiver* | *Alabama(people)* | *Noctuidae* | *Harvest(album)* | *USSAlabama* | *Alabamalanguage* | *Alabama(band)* | *Moth* | *UniversityofAlabama* | *Alabama, NewYork*]

The second dataset, that we will refer in what follows as *WIKI*, consists of 350 Wikipedia entity pages selected randomly. The text articles contain 5,812 entity 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 [7].

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

² <http://research.microsoft.com/users/silviu/WebAssistant/TestData>

table 1 for the *NEWS* dataset and in table 2 for the *WIKI* dataset. Both tables also report figures obtained by Cucerzan on the same datasets [7].

In the first experiment the best result equals the best available system at the state of the art, with an accuracy of 91.4% 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 at the state of the art. The *highest method* achieves results below the state of the art of 91.4%, even if, 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 1. Comparison of proposed Named Entity Disambiguation Functions on *NEWS* dataset

Literature Systems	Accuracy	Function	Accuracy
Cucerzan baseline [7]	51.7%	Highest	82.2%
Cucerzan [7]	91.4%	Combination	91.5%
		Propagation	68.7%

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

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

Literature Systems	Accuracy	Function	Accuracy
Cucerzan baseline [7]	86.2%	Highest	87.1%
Cucerzan [7]	88.3%	Combination	89.8%
		Propagation	84.3%

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 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, giving value to 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 states 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.

A consideration must be done on the decision of using Wikipedia as *entity inventory*: in terms of lexical coverage, especially with reference to specific domains, the choice impose some limitations. It is the case of specific technical areas, such as Medicine, Biology, specific fields of engineering, etc., which 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. Gentile, A.L., Zhang, Z., Xia, L., Iria, J.: Graph-based Semantic Relatedness for Named Entity Disambiguation. In: Dicheva, D., Nikolov, R., Stefanova, E. (eds.) Proceedings of S3T 2009: International Conference on Software, Services and Semantic Technologies, Sofia, Bulgaria, October 28–29, pp. 13–20 (2009)
2. Agirre, E., Martínez, D., López de Lacalle, O., Soroa, A.: 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, pp. 585–593. Association for Computational Linguistics (July 2006)
3. Mihalcea, R.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, The Association for Computational Linguistics (2005)
4. Navigli, R., Lapata, M.: Graph connectivity measures for unsupervised word sense disambiguation. In: Veloso, M.M. (ed.) IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1683–1688 (2007)
5. Sinha, R., Mihalcea, R.: 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), pp. 363–369. IEEE Computer Society, Los Alamitos (2007)
6. Bunescu, R.C., Pasca, M.: Using Encyclopedic Knowledge for Named Entity Disambiguation. In: EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, The Association for Computer Linguistics (2006)
7. 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, pp. 708–716. Association for Computational Linguistics (June 2007)

8. Minkov, E., Cohen, W.W., Ng, A.Y.: Contextual search and name disambiguation in email using graphs. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Järvelin, K. (eds.) SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, pp. 27–34. ACM, New York (2006)
9. Kalashnikov, D.V., Mehrotra, S.: A probabilistic model for entity disambiguation using relationships. In: SIAM International Conference on Data Mining, SDM (2005)
10. Grishman, R., Sundheim, B.: Message Understanding Conference- 6: A Brief History. In: COLING, pp. 466–471 (1996)
11. Ponzetto, S.P., Strube, M.: Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In: Moore, R.C., Bilmes, J.A., Chu-Carroll, J., Sanderson, M. (eds.) Proceedings of HLT-NAACL, Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, ACL (2006)
12. Strube, M., Ponzetto, S.P.: 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 Conference, pp. 1419–1424. AAAI Press, Menlo Park (2006)
13. Zesch, T., Gurevych, I., Mühlhäuser, M.: Analyzing and Accessing Wikipedia as a Lexical Semantic Resource. In: Biannual Conference of the Society for Computational Linguistics and Language Technology (2007)
14. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 265–283. MIT Press, Cambridge (1998)
15. Toral, A., Munoz, R.: A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In: Workshop on New Text, 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy (April 2006)
16. Kazama, J., Torisawa, K.: Exploiting wikipedia as external knowledge for named entity recognition. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 698–707 (2007)
17. Vercoustre, A.M., Thom, J.A., Peheveski, J.: Entity ranking in Wikipedia. In: Wainwright, R.L., Haddad, H. (eds.) Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), pp. 1101–1106. ACM, New York (2008)
18. Gentile, A.L., Basile, P., Semeraro, G.: WibNED: Wikipedia Based Named Entity Disambiguation. In: Agosti, M., Esposito, F., Thanos, C. (eds.) Post-proceedings of the Fifth Italian Research Conference on Digital Libraries - IRCDL 2009: A Conference of the DELOS Association and the Department of Information Engineering of the University of Padua. Revised Selected Papers, DELOS: an Association for Digital Libraries, pp. 51–59 (2009)
19. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A. (ed.) CILing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)
20. Minkov, E., Wang, R., Cohen, W.W.: Extracting personal names from emails: Applying named entity recognition to informal text. In: Proceedings of HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, British Columbia, Canada (2005)
21. Zesch, T., Müller, C., Gurevych, I.: Using wiktionary for computing semantic relatedness. In: Fox, D., Gomes, C.P. (eds.) Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, pp. 861–866. AAAI Press, Menlo Park (2008)

22. Banerjee, S., Pedersen, T.: Extended Gloss Overlaps as a Measure of Semantic Relatedness. In: Gottlob, G., Walsh, T. (eds.) *IJCAI 2003, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pp. 805–810. M. Kaufmann, San Francisco (2003)
23. Resnik, P.: Disambiguating noun groupings with respect to WordNet senses. In: *Proceedings of the 3th Workshop on Very Large Corpora*, pp. 54–68. *ACL* (1995)
24. Kudo, T., Matsumoto, Y.: Fast Methods for Kernel-Based Text Analysis. In: Hinrichs, E., Roth, D. (eds.) *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 24–31 (2003)
25. Turdakov, D., Velikhov, P.: Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. In: Kuznetsov, S.D., Pleshachkov, P., Novikov, B., Shaporenkov, D. (eds.) *SYRCoDIS. CEUR Workshop Proceedings, CEUR-WS.org*, vol. 355 (2008)
26. Lovász, L.: Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty* 2, 353–398 (1996)
27. Iria, J., Xia, L., Zhang, Z.: Wit: Web people search disambiguation using random walks. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, Prague, Czech Republic, pp. 480–483. *ACL* (2007)
28. Nie, Z., Zhang, Y., Wen, J., Ma, W.: Object-level ranking: bringing order to web objects. In: *WWW 2005: Proceedings of the 14th international conference on World Wide Web*, pp. 567–574. *ACM, New York* (2005)
29. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppín, E.: Placing search in context: the concept revisited. *ACM Transactions on Information Systems* 20(1), 116–131 (2002)
30. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: *Seventh International World-Wide Web Conference (WWW 1998)*. (1998)