Graph-based Semantic Relatedness for Named Entity Disambiguation

Anna Lisa Gentile[1], Ziqi Zhang[2], Lei Xia[2], and José Iria[2]

[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, j.iria}@dcs.shef.ac.uk

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

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. The goal of SemanticWeb is to provide machines with such ability, anchoring meanings to the words. 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. We propose an automatic method to associate a *unique sense* to each *entity*, exploiting Wikipedia[3] as freely available Knowledge Base, showing a novel solution for Named Entity Disambiguation (NED).

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 [1, 10, 11, 15] 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 [3, 4]. 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

[3] http://en.wikipedia.org/wiki/Wikipedia

means using semantic relations as input for NED. Compared to the best result by Cucerzan [4], which is an accuracy of 91.4%, our method achieves the same results using the same dataset, but it adds the benefit of having two clearly separate steps (relatedness sores, disambiguation), thus providing a glimpse of improving results in both directions.

The work is structured as follows: Section 2 proposes an overview of the Named Entity Disambiguation 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) [5]. 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 [13, 16, 20]. 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 [9] that, resembling more to a dictionary, has little coverage on named entities [16]. Lots of previous works exploited Wikipedia for the task of NER, e.g. to extract gazetteers [17] or as an external knowledge of features to use in a Conditional Random Field NER-tagger [7], to improve entity ranking in the field of Information Retrieval [19]. 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 [3] and Cucerzan [4]. 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 ofWikipedia 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% [3]. 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% [4]. 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. Contrary to these works, we propose 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.

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 [16] 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 [2, 9, 14] 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*), 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.

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 [8]. 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*, 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 [16], 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 [18].

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.

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 (Lovsz, 1993). 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 [6]. 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 x 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 → 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 : E → 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), and w : L → 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.

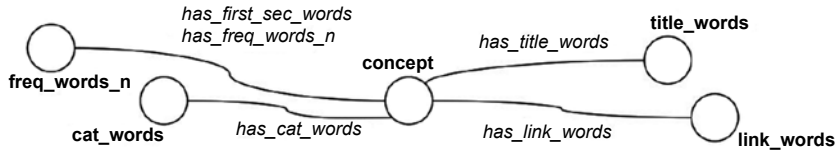

Fig. 1. The Graph representation model of concepts, features, and their relations. Circles indicate nodes (V) representing concepts and features; bold texts indicate types (T) of nodes; solid lines connecting nodes indicate edges (E), representing relations between concepts and features; italic texts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

$$\sum_{l \in L_{t\,d}} w\,(l) = 1$$

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

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

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. To tune the weight of different features we use a simulated annealing optimization method as described in [12]. 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 [6], 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 $t = 2$ to prevent smoothing out the graph . 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:

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

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 $k = 1 \cdots |\,S\,|$), 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 $f : S \rightarrow C$, 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 ($k = 1 \cdots | S |$); 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 $V = \{v_1, \ldots, v_{|C|}\}$ 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 $c_{i_{k'}}$ with $t \neq i$ and $t \neq j$). This step is repeated recursively, picking the 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 [4], which is publicly available[4]. In particular texts proposed are 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 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 → [*AlabamaClaims* | *Genus* | *CSSAlabama* | *AlabamaRiver* | *Alabama*(*people*) | *Noctuidae* | *Harvest*(*album*) | *USSAlabama* | *Alabamalanguage* | *Alabama*(*band*) | *Moth* | *UniversityofAlabama* | *Alabama, NewYork*]

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.

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 1, where are also reported figures obtained by

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

Cucerzan on the same dataset [4]. 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 82.21% 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 68.68%, which is in the middle between the baseline and the state of the art.

Table 1. Comparison of proposed Named Entity Disambiguation Functions

Literature Systems	Accuracy		Function	Accuracy
Cucerzan baseline [4]	51.7%		Highest	82.21%
Cucerzan [4]	**91.4%**		**Combination**	**91.46%**
			Propagation	68.68%

As expected, the *combination method* performs much better than others, rivaling the state of the art system. 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 result for the proposed novel method.

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 91.46%, which is comparable with the state of the art. The competitive accuracy reached hints at the usefulness of Semantic Relatedness measures for the process of Named Entities Disambiguation.

As future work, we plan to conduct experiments on other corpora and investigate if more precise relatedness scores could improve results.

References

1. E Agirre, D. Mart´ınez, O. L´opez de Lacalle, and 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, pages 585–593, Sydney, Australia, 2006. ACL.
2. S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In: G. Gottlob and T. Walsh, editors, IJCAI, pages 805–810. M. Kaufmann, 2003.
3. R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In: EACL. ACL, 2006.
4. S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In: EMNLP 2007: Empirical Methods in Natural Language Processing, June 28-30, 2007, Prague, Czech Republic, 2007.
5. R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In: COLING, pages 466–471, 1996.

6. J. Iria, L. Xia, and Z. Zhang. Wit: Web people search disambiguation using random walks. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pages 480–483, Prague, Czech Republic, 2007. ACL.
7. J. Kazama and K. Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 698–707, 2007.
8. T. Kudo and Y. Matsumoto. Fast methods for kernel-based text analysis. In: Erhard Hinrichs and Dan Roth, editors, Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pages 24–31, 2003.
9. C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In: C. Fellbaum, editor, WordNet: An Electronic Lexical Database, pages 265–283. MIT Press, 1998.
10. R. Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: HLT/EMNLP. ACL, 2005.
11. R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In: M. Veloso, editor, IJCAI, pages 1683–1688, 2007.
12. Z. Nie, Y. Zhang, J. Wen, and W. Ma. Object-level ranking: bringing order to web objects. In: WWW '05: Proceedings of the 14th international conference on World Wide Web, pages 567–574, New York, NY, USA, 2005. ACM.
13. S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In: R. C. Moore, J. A. Bilmes, J. Chu-Carroll, and M. Sanderson, editors, HLT-NAACL. ACL, 2006.
14. P. Resnik. Disambiguating noun groupings with respect to WordNet senses. In: Proceedings of the 3th Workshop on Very Large Corpora, pages 54–68. ACL, 1995.
15. R. Sinha and R. Mihalcea. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In: ICSC, pages 363–369. IEEE Computer Society, 2007.
16. M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In: AAAI, pages 1419–1424. AAAI Press, 2006.
17. A. Toral and R. Munoz. 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., 2006.
18. D. Turdakov and P. Velikhov. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. In: S. D. Kuznetsov, P. Pleshachkov, B. Novikov, and D. Shaporenkov, editors, SYRCoDIS, volume 355 of CEURWorkshop Proceedings. CEUR-WS.org, 2008.
19. A. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in wikipedia. In: R. L. Wainwright and H. Haddad, editors, SAC, pages 1101–1106. ACM, 2008.
20. T. Zesch, I. Gurevych, and M. Muhlhauser. Analyzing and accessing wikipedia as a lexical semantic resource. In: Biannual Conference of the Society for Computational Linguistics and Language Technology, 2007.
21. T. Zesch, C. M¨uller, and I. Gurevych. Using wiktionary for computing semantic relatedness. In: D. Fox and C. P. Gomes, editors, AAAI, pages 861–866. AAAI Press, 2008.