WibNED: Wikipedia Based Named Entity Disambiguation

Anna Lisa Gentile, Pierpaolo Basile, and Giovanni Semeraro

Dipartimento di Informatica, Università di Bari
Via E. Orabona, 4 - 70125 Bari - Italia
{al.gentile, basilepp, semeraro}@di.uniba.it

Abstract. Natural Language is a mean to express and discuss concepts, which are taken to be abstractions from perceptions of the experienced real world: what texts describe consist of objects and events. Objects of the real world are identified by proper names, which are words, thus raising the problem of proper linkage between the textual reference and the real object. This work adresses the problem of automatically association of meanings to words within an unstructured text and focuses the attention on words representing Named Entities. The proposed solution consists of a Knowledge based algorithm for Named Entity Disambiguation: we used an *ad hoc* builded corpus, extracted form Wikipedia's articles to prove the soundness of the algorithm.

1 Introduction

A proper name is a word or a list of words that refers to a real world object. According to Frege, a proper name has a reference (Bedeutung) and a sense (Sinn) [6]. The reference is the object that the expression refers to (different linguistic expressions can have the same reference). The sense is the *cognitive significance*, the way by which the referent is presented. Linguistic Expressions with the same reference may have different senses, so it is necessary to disambiguate between them. In Natural Language Processing field Named Entity Disambiguation is the task that aims to solve this issue. NLP operations include text normalization, tokenization, stop words elimination, stemming, Part Of Speech tagging, lemmatization. Further steps, as Word Sense Disambiguation (WSD) or Named Entity Recognition (NER), are aimed at enriching texts with semantic information. Named Entity Disambiguation (NED) is the procedure that solves the correspondence between real-world entities and mentions within text. The proposed approach automatically associates each entity in a text with a unique identifier, a URI from Wikipedia[1], which is used as an "entity-provider". The contribution of this work is twofold: firstly a novel knowledge based approach for NED is proposed; secondly the work shows a method to build a testbed dataset from Wikipedia. The suggested solution is completely knowledge-based, with the advantage that no training data is needed: indeed, manually annotated data for this task is not easily available, so acquiring such data can be expensive.

[1] http://www.wikipedia.org

The work is structured as follows: Section 2 proposes an overview of Named Entity Disambiguation task, together with a description of available solutions to exploit Wikipedia for the issue of Named Entities. Section 3 presents the proposed Wikipedia based Named Entity Disambiguation algorithm, named WibNED. Section 4 presents the dataset used for experiments, which are then described in section 5. Conclusions close the paper.

2 Related Work

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) [8]. 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 [12, 13, 16]. 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 [5] that, resembling more to a dictionary, has little coverage on named entities [13]. Lots of previous works exploited Wikipedia for the task of NER, e.g. to extract gazetteers [14] or as an external knowledge of features to use in a Conditional Random Field NER-tagger [9], to improve entity ranking in the field of Information Retrieval [15]. 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 of Wikipedia entries, together with correlations between pages learned from the structure of the knowledge source (mostly using Wikipedia Categories assigned to the pages). 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. 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. The proposed solution, differently to previous methods, exploits words in the context of an entity in a simple way, calculating the gloss overlapping between context and dictionary entries.

For the task of NED little resources and benchmark data are publicly available. On the other hand lots of data is available for the task of Named Entity Recognition: multilingual benchmarking and evaluations have been performed within several events, such

as the Message Understanding Conferences (MUC) series organized by DARPA, the International Conference on Language Resources and Evaluation (LREC), the Computational Natural Language Learning (CoNLL) workshops, the Automatic Content Extraction (ACE) series organized by NIST, the Multilingual Entity Task Conference (MET), the Information Retrieval and Extraction Exercise (IREX). The problem with data shared within these events is that entities are not labelled with a URI, but only classified within a set of predefined entity classes, which means that is not directly reusable for the task of NED. Some useful data for NED has been provided by Cucerzan, but the dataset only contains information about Named Entities and not all the text, so it is not useful for the purpose of this work. Moreover we wanted to evaluate our algorithm for the italian language and the available dataset is in english. For these reasons a specific dataset has been built to validate the proposal, automatically extracted from italian Wikipedia articles containing ambiguous entities, maintaing both entities and other words within the text.

3 WibNED: Wikipedia based Named Entity Disambiguation

The goal of a WSD algorithm consists in assigning a word w_i occurring in a document d with its appropriate meaning or sense s, by exploiting the *context* C in which w_i is found. The context C for w_i is defined as a set of words that precede and follow w_i. The sense s is selected from a predefined set of possibilities, usually known as *sense inventory*.

The Lesk algorithm is a classical algorithm for Word Sense Disambiguation for all words in unrestricted text. It was introduced by Mike Lesk in 1986 [11]. The basic assumption is that words in a given neighbourhood will probably share a common topic. Apart from knowledge about the context (the immediate surrounding words), the algorithm requires a machine readable dictionary, with an entry for each possible sense for a word. The original algorithm takes into account words pairwise and computes overlap among sense definitions: the sense pair with the highest overlap score is chosen.

The proposed algorithm, named WibNED, is an adaptation of Lesk dictionary-based WSD algorithm [1]. In the WibNED algorithm the words to disambiguate are only those representing an Entity.

WibNED takes as input a document $d = \{w_1, \ldots, w_j, e_{j+1}, w_{j+2}, \ldots, w_h, e_{h+1}, w_{h+2}, \ldots\}$ and returns a list of Wikipedia URIs $X = \{s_1, s_2, \ldots, s_k\}$ 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. We define the *context* C of the target entity e_i to be a window of n words to the left and another n words to the right, for a total of $2n$ surrounding words. In the current version of the algorithm if other entities occur in the context of the target entity, they are considered as words and not as entities.

Algorithm 1 describes the complete procedure for the disambiguation of entities. The input for the algorithm is an ordered list W

$$W = (w_1, \ldots, w_j, e_{j+1}, w_{j+2}, \ldots, w_h, e_{h+1}, w_{h+2}, \ldots)$$

of words in a document, processed with a NLP tool: w_i are common words while e_i are tagged as Named Entities.

The NLP tool used to process the document collection is META [2], that performs the following operations:

- tokenization;
- part of speech (POS) tagging;
- stop words elimination;
- word sense disambiguation by using WordNet;
- Entity Recognition performed with Yamcha [10], a NER annotator which uses a Support Vector Machine techniques.

Each document of the collection is then transformed in an ordered list of common words w_i and Named Entities e_i:

$$W = (w_1, \ldots, w_j, e_{j+1}, w_{j+2}, \ldots, w_h, e_{h+1}, w_{h+2}, \ldots)$$

The list W is the input for the main procedure, named WibNED, that finds the proper Wikipedia URI for each polysemous entity e_i in W. WibNED uses several subprocedures. The subprocedure QueryWiki finds all possible Wikipedia pages answering the query e_i. Each element of the returning set contains the page URI and a short textual description of the page. The subprocedure pickPage computes the overlap between the context of the target entity and the description obtained from the Wikipedia page for each candidate sense. It returns the candidate entity which maximizes the overlap.

4 Dataset

The dataset used for Experiments consists of 752 documents extracted from italian Wikipedia. The evaluation of a NED algorithm which gives as output Wikipedia URIs needs a dataset containing ambigous entites, already tagged with the correct URI belonging to Wikipedia.

We implemented a procedure to build an automatic annotated corpus, starting from a list of ambigous entities (i.e. entities whose surface form has an associated *Disambiguation Page*[2] in Wikipedia). We used a list of 100 ambiguos surface form, taken from italian Wikipedia, $A = (a_1, \ldots, a_{100})$. For each a_i we accessed the related *Disambiguation Page* on Wikipedia and we picked the most significative senses for a_i, $s_{a_i} = (s_1, \ldots, s_j)$, with j $\leq$ 4, considering only senses referring to Named Entities and using heuristics to reject poor senses. For example, considering the *disambiguation page* for the italian word "mosca", the sense referring to *Mosca (Moskva), the capital and the largest city of Russia*, has been stored whereas the sense referring to *Muscomorpha, a group of flies*, has been ignored because it is a common noun word. Starting from $S = (s_{a_1}, \ldots, s_{a_t}, \ldots, s_{a_{100}})$ for each s_j in s_{a_t} we choosed up to 5 generic Wikipedia articles that contain at least a link to the sense s_j. Each article has been processed using META [2] and has been stored as a single file, using a IOB like format, similar to CoNLL 2003 Named Entity Recognition corpus[3]: each row contains a token, its Part Of Speech tag, its lemma and a final tag which is valued as O if the token has not been

[2] http://en.wikipedia.org/wiki/Category Disambiguation_pages

[3] http://www.cnts.ua.ac.be/conll2003/ner/

Algorithm 1 $WibNED$, the algorithm for the disambiguation of entities

procedure $WibNED(W)$
▷ finds the proper Wikipedia URI for each polysemous entity e_i in the ordered list $W = (w_1, \ldots, w_j, e_{j+1}, w_{j+2}, \ldots, w_h, e_{h+1}, w_{h+2}, \ldots)$
W is the list of words in a document, processed with a NLP tool.
w_i are common words while e_i are tagged as Named Entities.
  $S \leftarrow W$
  for all $e_i \in S$ **do**
    $Context_{e_i} \leftarrow \{w_{i-n}, ..., w_{i+n}\}$
    $Candidate_{e_i} \leftarrow QueryWiki(e_i)$
    $s_{e_i} \leftarrow pickPage(Context_{e_i}, Candidate_{e_i})$
    $S.replace(e_i, s_{e_i})$
  end for
  return S
end procedure

procedure $QueryWiki(e_i)$
▷ finds all possible Wikipedia pages answering the query e_i. Each element of the returning set contains the page URI and a short textual description of the page.
  $Candidate \leftarrow \{\}$
  $WikiResults \leftarrow allpagesfromWikipediaansweringthequery e_i$
  for all $wikis_i \in WikiResults$ **do**
    $c_i.uri \leftarrow wikis_i$
    $c_i.description \leftarrow describe(wikis_i)$
▷ $describe(wikis_i)$ builds a short textual description of the page.
    $Candidate.add(c_i)$
  end for
  return $Candidate$
end procedure

procedure $pickPage(Context, Candidate)$
  $maxOverlap \leftarrow 0$
  $bestPage \leftarrow$ **null**
  for all $c_i \in Candidate$ **do**
    $currentOverlap \leftarrow computeOverlap(c_i.description, Context)$
    if $currentOverlap > maxOverlap$ **then**
▷ $computeOverlap(c_i.description, Context)$ calculate the number of overlapping words between $Context$ and $c_i.description$
      $bestPage \leftarrow c_i.uri$
      $maxOverlap \leftarrow currentOverlap$
    end if
  end for
  return $bestPage$
end procedure

recognized as an entity, B-<Wikipedia URI for the entity> if the token is the beginning of an entity and I-<Wikipedia URI for the entity> if the token continues an entity.

Each text contains on the average 89 entities. In table 1 we report a piece of a document to show an example of text. It is a document included in the corpus, specifically it is an article taken from italian Wikipedia about *The Beastie Boys*. In this piece of text there is only one entity, represented by the two words John Berry. Together with entity annotations, the corpus also contains the Part Of Speech and the stem for each word, thus allowing to refine and improve computation over the corpus.

Table 1. Corpus example

Il	RS	Il	O
nome	SS	nome	O
”	XPO	”	O
Beastie	SP	beastie	O
”	XPO	”	O
,	XPW	,	O
inventato	VSP	inventare	O
dall	SN	dall	O
’	XP	’	O
ex	SN	ex	O
componente	SS	componente	O
John	SPN	John	B-http:it.wikipedia.org/wiki/John_Berry
Berry	SPN	Berry	I-http:it.wikipedia.org/wiki/John_Berry
,	XPW	,	O
é	VI	essere	O
l	SN	l	O
’	XP	’	O
acronimo	AS	acronimo	O
della	ES	della	O
frase	SS	frase	O
”	XPO	”	O
Boys	YF	Boys	O
Entering	YF	Entering	O
Anarchistic	YF	Anarchistic	O
States	YF	States	O
Towards	YF	Towards	O
Inner	SPN	Inner	O
Excellence	SPN	Excellence	O
”	XPO	”	O

5 Experiments

We performed the experiment following the methods generally used to evaluate Word Sense Disambiguation (WSD) algorithms. The entity WSD is not an end in itself but

rather an intermediate task which contributes to an overall task such as information retrieval, ontology building, etc. This opens the possibility of two types of evaluation for WSD work (using terminology borrowed from biology): in vitro evaluation, where WSD systems are tested independently of any application, using specially constructed benchmarks; and evaluation in vivo, where, rather then being evaluated in isolation, results are evaluated in terms of their contribution to the overall performance of a system designed for a particular application (e.g., information retrieval). In this instance we adopt in vitro evaluation in order to evaluate the accuracy and the potentialities of the algorithm in an independent way. In vitro evaluation, despite its artificiality, enables close examination of the problems plaguing a given task. In its most basic form this type of evaluation involves comparison of the output of a system for a given input, using measures such as precision and recall. Alternatively, in vitro evaluation can focus on study of the behavior and performance of systems on a series of test suites representing the range of linguistic problems likely to arise in attempting WSD. Considerably deeper understanding of the factors involved in the disambiguation task is required before appropriate test suites for typological evaluation of WSD results can be devised. The in vitro evaluation demands the creation of a manually sense-tagged reference corpus containing an agreed-upon set of sense distinctions. The idea is to build a corpus, find the entity into the corpus and annotate the entity with relative sense. In our experiments we use Wikipedia as Sense Inventory for the entities because it's the same used by the WibNED algorithm.

WibNED is implemented in JAVA, by using Lexical Collector web service [7] as accessing point to the Wikipedia Sense Inventory. We run the WibNED algorithm on the dataset described in Section 4 in order to evaluate its effectiveness. We used the accuracy metric that describes the ratio between entities correctly labelled by WibNED and total number of entities within the document.

Experimetal results are showed in figure 1: on the x-axis are reported single documents while on the y-axis is reported the accuracy of WiBNED for each document. Documents on x-axis are ordered according to ascending accuracy.

Total Number of Entities	67029
Total Number of correctly labelled Entities	19131
Average number of Entities per Document	89
Average number of Correctly labelled Entities per Document	25
Total Accuracy	0.285
Average Accuracy on single Document	0,282
Minimal Accuracy	0,000
Maximal Accuracy	0,833

Table 2. WibNED Results

Some statistics are reported in table 2. The table provides the total number of entities within the corpus, the total number of correctly labelled entities by WiBNED

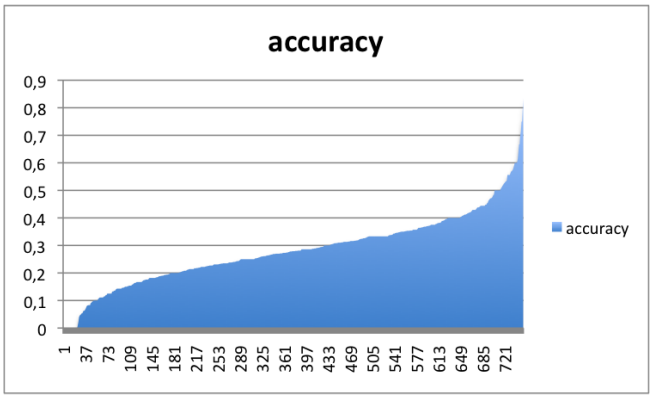

Fig. 1. Accuracy of WiBNED on 752 documents

algorithm, the average number of entities on single documents, the average number of correctly labelled entities per document. Results about accuracy are showed: total accuracy which is calculated considering the total number of entities and total number of correctly labelled entities, regardless of distribution between documents, average accuracy considering the mean of accuracy of all documents. The minimal and maximal values reported for accuracy are listed in the table.

The results are quite encouraging if we consider that WibNED is a very simple algorithm based only on the string-matching between the words in Wikipedia definition and the words within the context of the target entity. The algorithm achieves on average 28,2% of accuracy. Taking into account more informative features borrowed from Wikipedia, such as category labels associated to each article or internal links, could improve results of the algorithm.

6 Conclusions

In this paper we presented the WibNED algorithm for Named Entity Disambiguation and we evaluated it on italian language using an *ad hoc* builded corpus, automatically annotated with Wikipedia URIs.

The task of disambiguating Named Entities within a text and the problem of identity and reference are important issues for many research field, most of all for NLP: the WibNED algorithm associates unique references to words and uses popular URLs (Wikipedia's) as "canonical" URIs.

An ongoing work is focused on improving accuracy of WiBNED algorithm, relying on more Wikipedia features, such as links and categories instead of using only words within the disambiguation process.

References

1. S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing'02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 136–145. Springer-Verlag, 2002.
2. P. Basile, M. de Gemmis, A.L. Gentile, L. Iaquinta, P. Lops, and G. Semeraro. Meta multilanguage text analyzer. In *Proceedings of the Language and Speech Technology Conference LangTech 2008*, pages 137–140, 2008.
3. R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16. ACL, 2006.
4. S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP 2007: Empirical Methods in Natural Language Processing*, pages 708–716, 2007.
5. C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
6. G. Frege. Uber sinn und bedeutung. In Mark Textor, editor, *Funktion - Begriff - Bedeutung*, volume 4 of *Sammlung Philosophie*. Vandenhoeck & Ruprecht, Göttingen, 1892.
7. A. L. Gentile, P. Basile, L. Iaquinta, and G. Semeraro. Lexical and semantic resources for nlp: From words to meanings. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5179/2008 of *Lecture Notes in Computer Science*, pages 277–284. Springer Berlin / Heidelberg, 2008.
8. R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *COLING*, pages 466–471, 1996.
9. 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.
10. 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.
11. M. E. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the Fifth International Conference on Systems Documentation*, pages 24–26, Toronto, CA, 1986. ACM.
12. S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In Robert C. Moore, Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, *HLT-NAACL*. ACL, 2006.
13. M. Strube and S.P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, pages 1419–1424. AAAI Press, 2006.
14. 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, EACL 2006, 11th Conference of the European Chapter of the Association for Computational Linguistics*. ACL, 2006.
15. 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.
16. T. Zesch, I. Gurevych, and M. Mühlhäuser. Analyzing and accessing wikipedia as a lexical semantic resource. In *Biannual Conference of the Society for Computational Linguistics and Language Technology*, 2007.