IJCLA VOL. 3, NO. 1, JAN-JUN 2012, PP. 137–150
RECEIVED 28/10/11 ACCEPTED 09/12/11 FINAL 15/06/12

Subsymbolic Semantic Named Entity Recognition

RONALD WINNEMÖLLER

University of Hamburg, Germany

ABSTRACT

We present a novel application of our subsymbolic semantic TSR approach to named entity recognition and classification (NERC) in order to demonstrate the generic utility of the TSR approach beyond word sense disambiguation and language identification. Experimental results support our hypothesis that TSR techniques can successfully recognize named entities of different types in several languages. These experiments were based on a common framework infrastructure using different sets of features on two German and two English corpora.

1 INTRODUCTION

Named Entity Recognition and Classification (NERC) is an important topic: not only as subtask for natural language processing in terms of scientific research, but also for production environment applications such as web or intranet search, text mining, or content management software. Specifically, we found by analyzing some of our university's web servers log files that approx. 60% of all search queries are related to named entities of a particular kind (mainly personal or organizational data).

For several years, NERC tasks achieve good performance, especially for resource-rich languages. Still there is room for improvement regarding NERC for low-resource languages in named entity disambiguation, named entity fine-grained annotation and domain adaption. In the past, we have shown that the Text Sense Representations (TSR) approach described in Section 2 is well-suited for fine-grained word sense disambiguation. In this paper, we demonstrate that basic NERC also is a sensible topic for applying the TSR methodology thus paving the road for

TSR-based fine grained named entity disambiguation and classification (which, however, is beyond the scope of this paper).

Most current methods in the NERC field follow the statistical analysis paradigm: usually, a handful of selected lexico-syntactic features (e.g. bigram statistics, etc.) are fed into a machine learning algorithm trained on pre-annotated data. Sometimes, this process is augmented by applying handcrafted recognition rules.

In this paper, we will in principle also follow this path but instead of using lexico-syntactic features or applying explicit semantic databases such as wordnet, etc., we will introduce our own methodology of TSR-based subsymbolic semantics and pragmatics to the NERC field (cf. [1], [2]). From David Nadeaus great survey on NERC research and issues (cf. [3]) as well as our own literature work we derive that this indeed is a novel approach to NERC.

In the remainder of this paper, we will briefly describe the TSR approach and its application to NERC. We will then put our work in scientific context by relating to other researchers work in the field. Afterwards, we present some experimental results and conclude with a discussion about our findings.

2 About Text Sense Representations

In traditional Computational Linguistics, text meaning is usually encoded in a specific logical form, thus enabling semantic inferencing (cf. Allen [4]). Pragmatic aspects are often encoded as "heuristics" within a particular algorithm or even not handled at all. An alternative view, often attributed to the field of Text Engineering, uses implicit text meaning representation based on vector space models and accessible by specific algorithms, such as LSA (cf. Landauer *et al* [5]) and others. Other approaches associate lexical items to entities within a dictionary or an ontology in order to create a meaning oriented text knowledge representation, e.g. in the Microkosmos Machine Translation System (cf. Mahesh and Nirenburg [6]).

Our view of Text Meaning is more oriented toward pragmatics and general world knowledge and thus somewhat related to the field of prototype semantics (cf. Baerenfaenger [7], Meinhardt [8], Overberg[9] and others). For understanding "meaning", we rely on Wittgenstein's intuitive *family resemblance* notion of the *use* of language, particularly the usage of a word or text fragment in its context (cf. Wittgenstein's "Investigations" [10]). According to this theory, a human is - for example - able to recognize particular activities as “playing games” even though no single sharp common feature exists that is shared by every possible “game” (e.g. not every game is about winning; some, but not all games require teams, etc.). For a better understanding a visual interpretation of the semantic space of “game” is provided in figure 1.

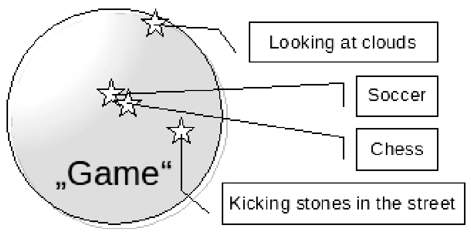

Fig. 1. Visual representation of a prototype for “game”

“Semantic spaces” can be regarded as prototypes in the sense described by Baerenfaenger [7] with some “typical”, i.e. central concepts located around a nucleus and others located at the periphery. These semantic spaces may also interfere with each other so that one concept might be close to the nucleus of one space and peripheral to other spaces.

Thus our notion of text meaning covers the ability of language to assign many pragmatic aspects of meaning to particular words or text fragments, some of which might be obvious, some might be rather unusual.

How words are used in language is analyzed by using the a.m. hierarchical system of *categories* so that a unique set of categories is associated with each particular word or text fragment. It is important to note that we are not assigning predefined “sense definitions” such as WordNet synsets (cf. Miller *et al* [11]) to particular word uses but rather associate data structures to word instances that contain hierarchical views of many possible uses of those words.

This hierarchical view is the basis of what we call a *Text Sense Representation* (TSR) and the foundation of our implementation of a text meaning representation (cf. [1]).

TSRs provide a methodology to represent semantic spaces in a uniform and fairly generally applicable way while being constructed in a fully automatic fashion prior to their use. The basic underlying data structures are tree hierarchies of labeled and weighted nodes, constructed from a web directory such as the Open Directory Project (ODP, cf. [12]).

Even though TSRs were described in greater detail by Winnemöller in [1] and [13], we will briefly and informally introduce the fundamentals of TSR tree construction and some operations on TSRs:

1. **Web directory data acquisition** Before building TSR Trees it is necessary to retrieve the underlying web directory data. In our case, we used the ODP RDF-like data dumps, but in principle other data sources like Yahoo [14] or Internet newsgroups data can be used as well. Our implementation uses the ODP directory data in order to create a TSR tree structure for each term that is contained in the ODP data (a term being a sequence of alphanumeric characters). The TSRs are constructed from the nodes of the web directory—a schematic excerpt of which is shown in Figure 2 on the next page.
2. **Sense node construction** Every text node of the web directory is analyzed into distinctive terms (usually stemmed words). Each term is then associated with the full entry path. For example, if a term *account* is found in the directory node */top/business/finance/banking*, then a respective TSR tree node *[account – /top/business/finance/ banking]* is created. Each node can be interpreted as "single sense" associated with the extracted term. Each node within a particular tree is therefore associated to a specific category of the ODP structure.
3. **TSR Tree construction** from sense nodes All nodes associated with a term are merged into a single TSR tree structure: every path is weighted by a $TFxIDF$-like formula calculating the occurrences of the TSR term against the overall number of terms within that ODP category.

A very simplistic example of a TSR is presented in Figure 3[1]. One might notice that the leaf nodes do not necessarily add up to 1 - that is because there is also some knowledge encoded in the branch nodes.

Furthermore, we have defined a number of basic TSR operations: a TSR relatedness measure can be used in order to compare two TSRs, a so-called *OR* operation will combine several TSRs into one by creating a TSR that consists of the union set of all input nodes (using the respective maximum weight) while a *AND* operation will create a merged TSR that consists of the intersection set of all input nodes (using the respective minimum weight).

[1] In a subsequent step, the ODP labels are exchanged for a node numbering for technical reasons.

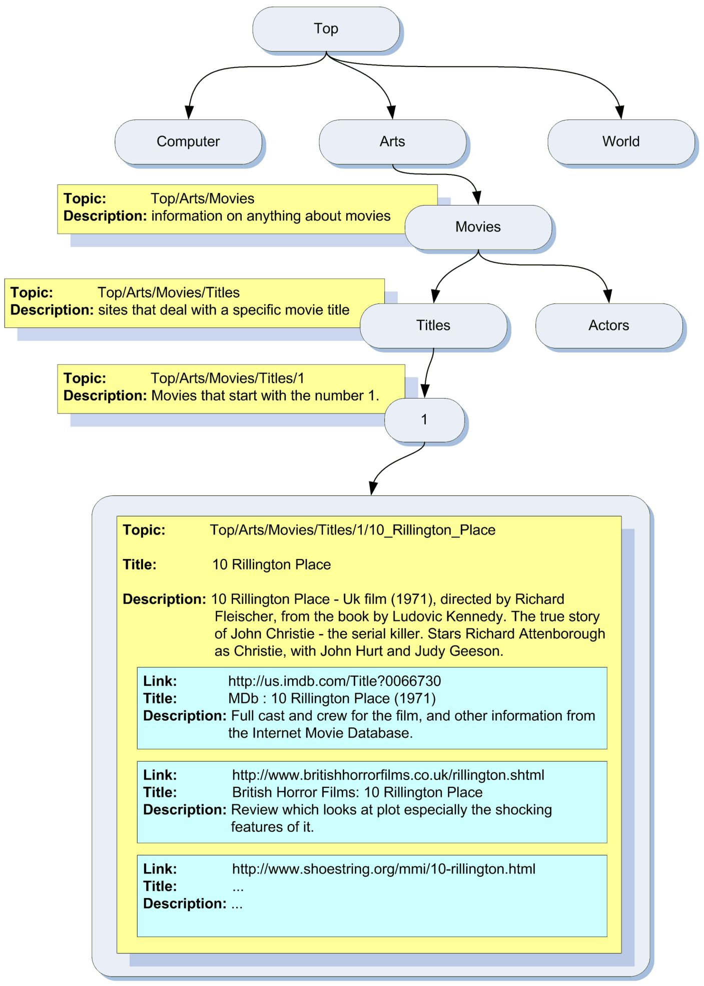
Top
Computer
Arts
World
Topic: Top/Arts/Movies
Description: information on anything about movies
Movies
Topic: Top/Arts/Movies/Titles
Description: sites that deal with a specific movie title
Titles
Actors
Topic: Top/Arts/Movies/Titles/1
Description: Movies that start with the number 1.
1
Topic: Top/Arts/Movies/Titles/1/10_Rillington_Place
Title: 10 Rillington Place
Description: 10 Rillington Place - Uk film (1971), directed by Richard Fleischer, from the book by Ludovic Kennedy. The true story of John Christie - the serial killer. Stars Richard Attenborough as Christie, with John Hurt and Judy Geeson.
Link: http://us.imdb.com/Title?0066730
Title: MDb : 10 Rillington Place (1971)
Description: Full cast and crew for the film, and other information from the Internet Movie Database.
Link: http://www.britishhorrorfilms.co.uk/rillington.shtml
Title: British Horror Films: 10 Rillington Place
Description: Review which looks at plot especially the shocking features of it.
Link: http://www.shoestring.org/mmi/10-rillington.html
Title: ...
Description: ...

Fig. 2. ODP excerpt (schematic)

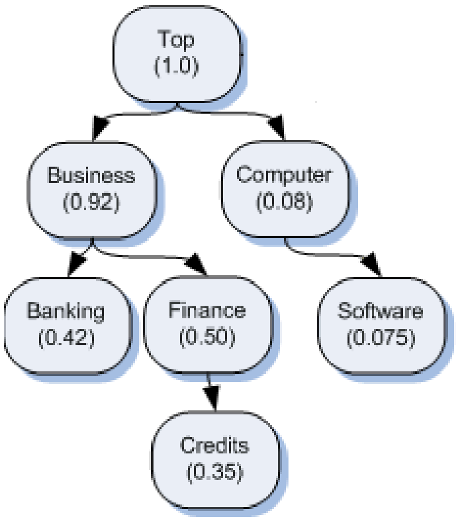

Fig. 3. TSR for the word "account"

In order to be able to compare TSRs with each other, we use a cosine function which consists of applying the well-known vector based cosine function onto the (vectorized) TSR tree node values.

Other operations include the ability of feature selection, as described by the author in [13] and tree minimization.

In summary, TSRs represent pragmatic aspects of text meaning via a combination of categorical text data from existing web directories and several semantic operations.

3 TSR BASED NERC

In the past, TSR vectors were used for unsupervised word disambiguation, language identification, categorizing emails and enriching the University of Hamburg telephone directory by web data (the latter two examples were internal projects and not published). The notion of semantic similarity in these cases was based on computing the cosine between vectors of serialized TSR nodes, i.e. TSR vectors.

Further work on TSRs derived the hypothesis that TSR vectors could be used as input data for machine learning algorithms just like term occurrences, n-grams or other term-related vector data. For this reason, we created a prototype system that reads raw text data, retrieves TSR vectors for the terms within those texts and submits those vectors (plus some TSR metadata such as *width*, *depth* and *size*) as feature vectors to some machine learning system.

Because the "meaning" of the textual terms is not represented by the term vectors themselves but rather by a complex data structure, i.e. a possibly large number of anonymous numerical elements which all together form the semantics of a particular term, we regard this approach "subsymbolic".

For this paper, we applied this methodology onto the problem of supervised NERC, being one important problem that can – to a certain extend – successfully be addressed by exchanging conventional term vector data input with TSR based vector data when training and testing a machine learning system.

4 Related Work

Much work in the NERC field is based on lexico-syntactic features and handcrafted syntactic rules but some approaches also address semantics-based NERC. In this section, we present a small subset of those approaches that we find representative in respect to this paper.

A good example for augmenting the conventional methodology by "semantic rules" was evaluated by Maynard *et al* – they used several lexico-syntactic features plus manually designed "semantic" rules in order to detect named entities (cf. [15]), reporting an F-score of about 60–65% for their experiments.

Even earlier, Cucchiarelli and Velardi report on a multi-stage unsupervised approach which consists of using several third party NERC taggers for creating a robust initial training corpus, then creating several syntactic context features off this corpus and subsequently apply a kind of wordnet-based contextualization (cf. [16]). An evaluation procedure results in about 88% precision and 84% recall on an italian corpus and about 94% precision and 90% recall on an english corpus derived from the Wall Street Journal. Here, an understanding of attribute-based semantics enter the process during the wordnet-related stage.

In addition to this kind of conceptualization technique, Boufaden *et al* employ a semantic NERC algorithm based on a handrafted ontology and similarity scores, derived by using the Lesk algorithm on the ontology and the wordsmyth thesaurus (cf. [17]). They report an F-score of about 86% on transcribed conversations from the domain of maritime search an rescue logs, despite the rather unusual genre (It might be argued, that the underlying knowledge domain is closed and quite restricted, though).

Using large volumes of web-based data, Gentile *et al* present a conceptual graph matching algorithm based on Wikipedia concepts (cf. [18]). Semantical relatedness measures are obtained by applying a random walk algorithm between concept nodes. Accurracy is reported to be about 94% which is comparable to state-of-the-art results. In this case, the system also was avaluated on a closed domain and genre corpus (in the english part, texts from the Wall Street Journal were used). Interestingly, the authors also employed a corpus bootstrapping method vaguely alike the one we used for this paper.

A similar approach was used by Richman and Schone, but they also researched the effect of using Wikipedia NERC on several different languages (cf. [19]). Because Wikipedia and the ODP share the property of addressing many languages, we recognize this work as supporting our notion of multi-language orientation.

5 Experiments

5.1 *Systems Setup*

For the experiments conducted for this paper, we used the OpenNLP framework as common basis for all experiments : *"OpenNLP is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution.* "[2] OpenNLP was used successfully by other researchers in the NERC field, e.g. recently by Abacha and Zweigenbaum (cf. [20]) because it has the advantages of being relatively simple, modular, open source and easy to extend.

In principle, we built three system setups using different types of feature sets:

1. The Baseline system setup – the "full feature" set[3]:
   (a) Window features of tokens : previous and next two tokens in context. According to the OpenNLP documentation, the current token is included unchanged while previous p- and next n-tokens are prefixed with p-distance and n-distance respectively

[2] Taken literally from the website `http://incubator.apache.org/opennlp/documentation/manual/opennlp.html`

[3] These are the features originally deployed with the OpenNLP NERC module

   (b) Window features of token classes. This feature behaves just like the token window feature except that it does not include the tokens themselves but the token "classes" (token classes are: lowercase-word, uppercase-word, alphanumeric-word, etc.)
   (c) Bigram feature: bigram tokens
   (d) Sentence feature: sentence begin and end tokens
   (e) Previous mapping feature: features using the outcome of previously occuring words[4]
2. The Baseline system setup – the so-called "realistic" feature set: this setup excludes all features from the above that make use of previous outcomes and in-sentence positioning – TSRs are functionally oriented towards subsymbolic text fragments rather than to words or sentences. It is more realistic than the full feature set setup because the TSR system setup cannot (yet) use such features in a similar way. In order to achieve comparable results, these features ((d) and (e)) have to be abandoned therefore.
3. The TSR system setup: this setup includes TSR related Features only: TSR width, size, depth plus the key indices and values of the TSR vector itself (the TSR vector was pruned by the TOP-algorithm leaving only the top-rated 100 elements, cf. [13]).
4. A combined Baseline+TSR system setup. This setup can not be used to evaluate the baseline versus the TSR based approach but it superficially shows the effect of combining both paradigms.

It is important to notice that only the setups (2) and (3) are fundamental for the cause of this paper (the others are interesting, but they are not suited to prove our point for the reasons given above).

Because we want to demonstrate the effectiveness of the TSR approach augmenting a given system rather than determining the "best" NERC system, we did not evaluate the different setups described here against other approaches.

5.2 *Corpus preparation*

Four corpora were prepared from randomly chosen wikipedia articles for experimenting:

1. *de-1K corpus* : german language; about 1 K sentences; 22 K words; 182 K bytes

[4] The OpenNLP code contains one more feature but this seems equivalent to this one

2. *en-1K corpus* : english language; about 1 K sentences; 30 K words; 214 K bytes
3. *de-10K corpus* : german language; about 10 K sentences; 222 K words; 1.738 K bytes
4. *en-10K corpus* : english language; about 10 K sentences; 294 K words; 2.073 K bytes

The corpora were annotated following roughly the approach described by Cucchiaralli and Velardi (cf. [16]) using a third party NERC tagger – in our case the balie baseline information extraction system (cf. [21]) – and afterwards manually corrected for obvious errors (e.g. one-and-two-character terms as well as non-alphanumeric terms were de-annotated when appropriate). Still, the corpora can be regarded as of relatively low-quality, but on the other hand there was little choice because we simply had no access to publicly licensed cost-free NERC-tagged english and german corpora.

5.3 *Results*

The results of the above explained experiments in terms of precision, recall and combined F-score are presented in tables 1, 2 on the next page and 3 on the facing page (respective best scores are set in boldface).

Table 1. Experimental Results: **Precision**

	german	**english**
1K Baseline System – full feature set	*0.89*	*0.86*
1K Baseline System – “realistic” feature set	0.74	0.75
1K TSR System	**0.88**	**0.91**
10K Baseline System – “realistic” feature set	0.85	0.85
10K TSR System	**0.92**	**0.91**
1K TSR + Baseline System	**0.96**	**0.94**

The results show that the TSR system performs better than the baseline system in terms of precision (here, the best score is actually achieved by the combination of TSR and conventional features) and recall on the german corpus (on the english corpus, the 10 K baseline is better, but only by a small percentage) and also by the combined F-score (TSR performs better than baseline in all experiments).

Table 2. Experimental Results: **Recall**

	german	**english**
1K Baseline System – full feature set	*0.69*	*0.75*
1K Baseline System – "realistic" feature set	0.44	0.58
1K TSR System	**0.51**	**0.60**
10K Baseline System – "realistic" feature set	0.58	**0.69**
10K TSR System	**0.66**	0.67
1K TSR + Baseline System	0.44	0.54

Table 3. Experimental Results: **F-score**

	german	**english**
1K Baseline System – full feature set	*0.78*	*0.80*
1K Baseline System – "realistic" feature set	0.55	0.65
1K TSR System	**0.65**	**0.72**
10K Baseline System – "realistic" feature set	0.69	0.76
10K TSR System	**0.77**	**0.77**
1K TSR + Baseline System	0.61	0.68

It is also worth mentioning that the baseline system using the full feature set would perform best in terms of recall and f-score but not for precision – but as we argued previously, this also means taking in account previous outcomes in context which is not (yet) available for TSR processing.

5.4 *Discussion*

Apart from the explicit outcome, there are a number of interesting points to note:

- High Precision in the case of a TSR based system indicates that the NEs that were detected usually we correct – only a few misclassifications were reported. This is a promising result that points future work to increasing recall rather than precision.
- Low Recall in the case of a TSR based system means that many NEs were missed. This is due to the fact that the intrinsic TSR methodology of learning new terms is not used within these experiments. Therefore only terms that occur within the word space of the ODP can be recognized. This situation can supposedly be remedied by

adding terms to the TSR index database through online TSR learning (using words-in-context for creating a corpus on-the-fly – these words can be taken from the original experiment corpus but can also be retrieved from the web using a common search engine) or offline TSR learning (using term features from wikipedia for creating words-in-context. These term features can be similar to the ones mentioned by Gentile *et al*, cf. [18]). Unfortunately, activating on- or offline learning of TSRs remains is out-of-scope of this paper and must remain topic of future research.

- The "realistic" baseline experiments tend to show a much better performance on english than on german data. Since the size of each training corpus within the experiment is approximately equal, it can be argued that the original tagging of the training corpus must be better in the english language case. This effect seems to vanish when using the TSR system on the larger 10K corpora which presumably relies on the fact that german is well represented in the TSR database.
- The problem of named entity ambiguity (cf. [22]) is not yet recognized; TSRs are in principle related to the surface form of words rather than to underlying attribute based semantics. Nonetheless, because of their rich internal hierarchical structure, they can be used to distinguish word senses on different levels of granularity and for many different domains.

6 Conclusions

We presented the novel application of TSR based techniques onto the research field of named entity recognition and classification, using a bootstrapped corpus for evaluating the (supervised) TSR approach against the conventional term based method.

In general, the TSR approach seems well suited to tackle NERC issues - either in a standalone fashion or in combination with conventional methods and features thus once more demonstrating the quite generic nature of the TSR methodology and concept.

Our findings include that the TSR setup led to good results in terms of precision but somewhat less desirable results in terms of recall on the account of several terms being not recognizable by the TSR system. Appropriate future work should therefore involve applying the different TSR learning techniques in order to increase recall scores.

We also showed that performance does not seem to degrade between languages (in our case: german and english) when using a large enough corpus of training data.

REFERENCES

1. Winnemöller, R.: Constructing text sense representations. In Hirst, G., Nirenburg, S., eds.: ACL 2004: Second Workshop on Text Meaning and Interpretation, Barcelona, Spain, Association for Computational Linguistics (July 2004) 17–24
2. Winnemöller, R.: Using meaning aspects for word sense disambiguation. In: 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Haifa, Israel (February 2008)
3. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Journal of Linguisticae Investigationes **30(1)** (2007) 3–26
4. Allen, J.: Natural Language Understanding. 2 edn. Benjaming/Cummings Publish. Corp, CA (1995)
5. Landauer, T.K., Foltz, P.W., Laham, D.: Introduction to latent semantic analysis. Discourse Processes **25** (1998) 259–284
6. Mahesh, K., Nirenburg, S.: A situated ontology for practical nlp. In: Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada (Aug 1995)
7. Bärenfänger, O.: Merkmals- und prototypensemantik: Einige grundsätzliche überlegungen. Linguistik online **12** (Mar 2002)
8. Meinhardt, H.J.: Invariante, variante und prototypische merkmale der wortbedeutung. Zeitschrift für Germanistik **5**(1) (1984) 60–69
9. Overberg, P.: Merkmalssemantik vs. prototypensemantik—anspruch und leistung zweier grundkonzepte der lexikalischen semantik. Master's thesis, Universität Münster (feb 1999)
10. Wittgenstein, L.: Philosophische Untersuchungen. In: Werkausgabe Bd. I. Suhrkamp Verlag, Frankfurt am Main (1984)
11. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. The MIT Press (1998)
12. Netscape Communications Corporation: Open directory project. dmoz.org (2004)
13. Winnemöller, R.: Knowledge based feature engineering using text sense representation trees. In: International Conference RANLP-2005, Borovets, Bulgaria (Sept 2005)
14. Yahoo Inc.: Yahoo. http://www.yahoo.com (01 2004)
15. Maynard, D., Bontcheva, K., Cunningham, H.: Towards a semantic extraction of named entities. In: In Recent Advances in Natural Language Processing. (2003)

16. Cucchiarelli, A., Velardi, P.: Unsupervised named entity recognition using syntactic and semantic contextual evidence. Computational Linguistics **27**(1) (2001) 123–131
17. Boufaden, N., Lapalme, G., Bengio, Y.: Extended semantic tagging for entity extraction. In: Beyond Named Entity Recognition Semantic labeling for NLP tasks Workshop held Jointly with LREC 2004, Lisbon, Portugal (may 2004) 49–54
18. Gentile, A.L., Zhang, Z., Xia, L., Iria, J.: Graph-based semantic relatedness for named entity disambiguation. Proceedings of S3T (2009) 13–20
19. Richman, A., Schone, P.: Mining Wiki Resources for Multilingual Named Entity Recognition. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. (2008) 1–9
20. Abacha, A.B., Zweigenbaum, P.: Medical entity recognition: A comparison of semantic and statistical methods. In: BioNLP 2011 Workshop of the 49th Annual Meeting of the ACL: Human Language Technologies, Portland, Oregon, USA, ACL (June 2011)
21. Nadeau, D.: Balie—baseline information extraction. Technical report, School of Information Technology & Engineering, University of Ottawa, Ottawa, Canada (January 2005)
22. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: EMNLP-CoNLL. (2007) 708–716

RONALD WINNEMÖLLER
REGIONAL COMPUTER CENTRE,
UNIVERSITY OF HAMBURG,
GERMANY
E-MAIL: <RONALD.WINNEMOELLER@RRZ.UNI-HAMBURG.DE>