Distinguishing Between Instances and Classes in the Wikipedia Taxonomy

Cäcilia Zirn[1,2], Vivi Nastase[1], and Michael Strube[1]

[1] EML Research gGmbH, Heidelberg, Germany
{zirn|nastase|strube}@eml-research.de
WWW home page: http://www.eml-research.de/nlp

[2] Department of Computational Linguistics
University of Heidelberg, Heidelberg, Germany
zirn@cl.uni-heidelberg.de

Abstract. This paper presents an automatic method for differentiating between instances and classes in a large scale taxonomy induced from the Wikipedia category network. The method exploits characteristics of the category names and the structure of the network. The approach we present is the first attempt to make this distinction automatically in a large scale resource. In contrast, this distinction has been made in WordNet and Cyc based on manual annotations. The result of the process is evaluated against ResearchCyc. On the subnetwork shared by our taxonomy and ResearchCyc we report 84.52% accuracy.

1 Introduction

The World Wide Web (WWW) is a latent repository of multi-lingual and multi-faceted knowledge. This knowledge is hard to get at: for humans because of the overwhelming quantity; for computers because of noise (unedited texts, conflicting information) and lack of structure. This leads naturally to the desire for more structured web content, which can be accessed, used and shared among software agents for a wide range of activities. This is the desideratum of the Semantic Web endeavour [1]. Enhancing web pages with semantic annotations requires a large knowledge resource to serve as reference and to provide a portal to a large network of organized information and of reasoning capabilities.

Manually created resources, such as WordNet [2] and Cyc [3], have been in use since the beginning of the 90's. The Natural Language Processing (NLP) community has now much experience working and using them in applications. This has revealed to us their strengths and weaknesses, thus providing guidelines for the development of new and better knowledge resources [4]. One weakness is coverage. Manually built repositories cannot – and are not supposed to – cope with the extremely large number of entities to be captured, as they are edited by a small number of qualified people. The web has provided a collaborative medium through which this laborious task of creating knowledge resources can be distributed among a large number of contributors. The downside is that not

all are computational linguists or lexicographers, and thus the task must be more flexible, and allow for the easy input of semi-structured knowledge. This is a very successful approach, as we have seen in the case of the online encyclopedia Wikipedia, which now covers more than 250 languages, 75 of which have more than 10,000 articles. It is therefore very appealing to try to induce from this semi-structured resource a large scale, multi-lingual ontology.

The first stage is to extract a taxonomy. Ponzetto and Strube [5] developed a method to accomplish this for the English Wikipedia, by inducing *isa* and *notisa* labels for the edges of the category network. An important feature of a useful taxonomy is differentiation between instances and classes – e.g. the authors of this paper are each an instance of the class PERSON[3]. This introduces a fundamental difference between elements in a resource:

- classes form the backbone of the network and form the actual ontology;
- classes are intensional descriptions of entities [6];
- in reasoning, instances are mapped to objects and classes to predicates;
- the class-class links are semantically different from the class-instance links;
- in texts, classes and instances have different syntactic behaviour – e.g. instances have no plural form, except in special cases when the speaker wants to emphasize some feature of the instance, as in the example: "I want it to be in a musically interesting catalogue (a label without compromises) so all the John Zorns and Mike Pattons of this globe are welcome to contact me."

The work presented in this paper focuses on developing methods to make the distinction between instances and classes automatically. In Section 2 we present a review of related work on distinguishing between instances and classes in large scale taxonomies. Section 3 reviews the methods for obtaining a taxonomy from the Wikipedia category network. Building upon this structure, we show in Section 4 our methods for labeling categories as instances or classes. The heuristics are evaluated against ResearchCyc, and the results are shown in Section 5.

2 Related Work

WordNet [2] is one of the most used lexical resources in NLP. It organizes open-class words in semantic networks: the nodes, *synsets*, represent senses, and contain a number of single or multi-word terms which have the same or very similar meaning; the edges represent different types of semantic relations, such as *hyponym-hypernym*, *meronym-holonym*, *antonymy*, *cause-effect*, *pertains to*. Among the members of a synset the *synonymy* relation holds. By far the most commonly used relation in WordNet is the *hyponym-hypernym*, which gives a taxonomic view of the resource. However, WordNet was not designed to be a taxonomy but rather a "map" of language. The organization has arisen as the

[3] We adopt the following notation conventions: `Sans Serif` for words and queries, *Italic* for relations, CAPITALS for Wikipedia pages and SMALL CAPS for concepts and Wikipedia categories.

links between synsets were added. Because of this, from the point of view of a formal taxonomy, WordNet is not perfect. One of the shortcomings is the lack of distinction between instances and classes [7, 8].

Miller and Hristea [9, 10] introduce manually the distinction between instances and classes in WordNet. The main features they used to identify instances are: (i) instances are proper nouns and therefore are capitalized; (ii) instances are unique, so they do not have instances themselves. Based on these criteria, the actual distinction was done manually by two judges. On 24,073 capitalized items, the agreement coefficient kappa was 0.75, indicating substantial, but not perfect, correspondence (for details on computing agreement statistics see [11]).

Cyc [3] is a large scale knowledge repository. It describes a wide variety of concepts and possible relations, and it is designed to be used for reasoning. Like WordNet, it was created manually, but there are functions to add non-atomic terms, e.g. "LiquidFN Nitrogen", automatically. From the beginning Cyc differentiated between instances and classes. There are two disjoint meta-concepts, `#$Individual` and `#$SetOrCollection`, based on which this distinction is made. All instances are linked through an *isa* chain to `#$Individual`, and all classes to `#$SetOrCollection`. Relations between two `#$SetOrCollection` items are labeled as *is generalized by (genls)*, whereas relations between an `#$Individual` item and a `#$SetOrCollection` item are labeled as *isa* (*isa*).

In both Cyc and WordNet the information about classes and instances is added manually, and is expensive both in terms of time and money.

Our literature review has not revealed any large scale resource in which this distinction is done automatically. The approach we present is based on the criteria identified by [9], and draws on the category structure of Wikipedia to assign automatically an instance or class label to categories and pages in this large network.

3 Wikipedia Taxonomy

Wikipedia is a free online encyclopedia, which grows through the collaborative efforts of volunteers over the Internet: anyone can contribute by writing or editing articles. As of March 2008, the English Wikipedia contains more than 2,300,000 articles[4]. The articles are organized in categories that can be created and edited as well. The categories themselves are organized into a hierarchy. Wikipedia's category and page network can be seen as a large semantic network. Categories in this network are connected by unlabeled links that represent different types of links: THUMB *isa* FINGER *part of* HAND.

Categories and pages linked with *isa* relations form a taxonomy. We show a sample in Figure 1. Ponzetto and Strube [5] describe methods to identify *isa* relations in Wikipedia's category network:

1. Filter out meta-categories used by Wikipedia for encyclopedia management using key words (e.g. `template`, `user`, `portal`).

[4] http://en.wikipedia.org/wiki/Wikipedia:About#Related_versions_and_projects

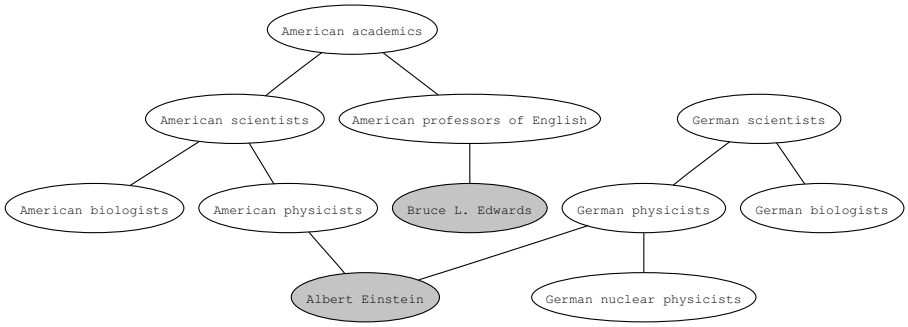

Fig. 1. Wikipedia Category Graph

2. Label as *is refined by* the relation between categories C_2 and C_1, if their names match the following patterns: C_1 = Y X and C_2 = X by Z – e.g. C_1 = Miles Davis Albums and C_2 = Albums by Artist.
3. Use two syntax-based methods:
   - Assign *isa* to the link between two categories if they share the same lexical head lemma – e.g. British Computer Scientists and Computer Scientists.
   - Assign *notisa* if one category contains the lemma of the lexical head of the other category in non-head position – e.g. Crime comics and Crime.

4a. Use structural information from the category network: for a category C, look for a page P with the same name. Take all of P's categories whose lexical heads are plural nouns $CP = \{C_1, C_2, ..., C_n\}$. Take all supercategories of $C_i, i = 1, ..., n$, $SC = \{SC_1, SC_2, ..., SC_k\}$. If the head lemma of one of C_i matches the head lemma of SC_j, label the relation between C and SC_j as *isa*. An example is provided in Figure 2. The category Microsoft has a homonymous page, categorized under Companies listed on NASDAQ which has the head lemma `companies`. Microsoft has a supercategory Computer and video game companies with the same head lemma. The link between Microsoft and Computer and video game companies is labeled as *isa*.

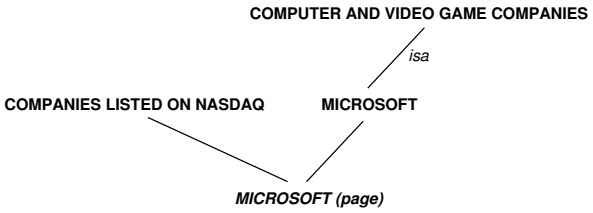

Fig. 2. Using structural information to induce *isa* links

4b. Assign *isa* label to the link between two categories if a page is redundantly categorized under both of them.
5. Use lexico-syntactic patterns in a corpus. This method uses two sets of patterns. One set is used to identify *isa* relations [12, 13] – for example `such X as Y`, X and Y take the values of categories and their subcategories respectively. The second set is used to identify *notisa* relations. This last group includes patterns for finding meronymic, locative, temporal and other relations.
   These patterns are used with a corpus built from English Wikipedia articles, and separately with the Tipster corpus [14]. The label is assigned by majority voting between the frequency counts for the two types of patterns.
6. Assign *isa* labels to links based on transitive closures – all categories along an *isa* chain are connected to each other by *isa* links.

These methods lead to a fully automatically generated large scale taxonomy from the Wikipedia version of September 25th, 2006 – 127,311 nodes, 267,479 links of which 106,258 *isa* links. The work described in this paper enhances this taxonomy with `instance` and `class` information for each node.

4 Methods

In this section we describe our methods for distinguishing automatically between instances and classes in the Wikipedia taxonomy. It may seem intuitive that categories in Wikipedia are all classes, but that is not the case. For example, United Nations, an instance of the class Organizations, appears both as a page and as a category in Wikipedia. The reason for this seems to be that it has many related concepts, which people naturally organize under this "umbrella". United Nations specialized agencies, United Nations charter, International Court of Justice, League of Nations are just some of the subcategories of the category United Nations. This is not a rare situation, as many instances – organizations, individuals, locations, time intervals, etc. – about which Wikipedia contributors have extensive knowledge are structured in this way.

4.1 Structure-based Method

Structure. The most important property of instances is that they are unique. As such, they do not have instances themselves [9]. Applying this to our taxonomy translates into finding categories which have no other subcategories or pages connected to them by *isa* relations. The reverse of this criterion is not true: if a category does not have other categories connected to it by *isa* relations, it is not necessarily an instance, it could also be a class that does not contain any other classes or instances. We use this criterion to determine categories that do have hyponyms, and which are therefore classes:

1. assign the `class` label to every category which has at least two hyponyms.

We adjust this rule to partially avoid erroneous *isa* labels introduced in the automatic *isa* links induction process, by introducing a second step:

2. assign the `class` label to every category which has exactly one hyponym, if this hyponym has more than one hyponym itself.

The remaining categories are processed based on clues from their names.

4.2 Category Name Analysis Methods

Named Entity Recognition (NER). Instances correspond to unique entities in the world, and are named entities. This fact is reflected in the Wikipedia category titles, which are accordingly capitalized. We then use an off-the-shelf named entity recognizer (NER), the CRFClassifier [15], and apply it to the category titles.

The category names consist of varied and complex noun phrase structures like Main kingdoms of the Puru clan or ... And You Will Know Us by the Trail of Dead albums. We pass onto the NER only the lexical heads, which we extract using the Stanford Parser [16]. The parser may return several heads for a phrase, depending on the connectivity of the parse structure it produces. The NER tags the heads with one of the following labels: `Person`, `Location`, `Organization`, if they are recognized as named entities, `Other` otherwise. If a majority of the heads of a category are tagged as `Other`, the category is labeled as a class, else it is marked as an instance.

Capitalization. Bunescu and Paşca [17] have developed an approach to detect whether Wikipedia pages represent named entities: Following the Wikipedia naming conventions[5], all content words are capitalized if they constitute a part of a named entity. If not, they are lowercased. As the Wikipedia naming conventions for pages and the ones for categories[6] share these rules, we can apply this heuristic to the category titles. For example in the title All India Council for Technical Education, all content words are capitalized, whereas in All America Football Conference coaches the last word is lowercased, because it does not belong to a named entity. In Wikipedia category titles the first word is always capitalized; this introduces ambiguity for single-word categories.

Our method proceeds as follows:

1. preprocess the first word with the CRFClassifier mentioned in section 4.2: if it is a named entity keep the spelling, if not, lowercase the word.
2. filter out all function words (closed class words such as prepositions and determiners)[7].
3. analyze the remaining words in the title: if all of them are capitalized, the title is classified as an instance.

[5] http://en.wikipedia.org/wiki/Naming_conventions
[6] http://en.wikipedia.org/wiki/Wikipedia:Naming_conventions_(categories)
[7] http://www.marlodge.supanet.com/museum/funcword.html

Plural. Instances are unique, therefore are mostly used in singular form. There are exceptions, as seen in the example: `The Millers are coming to our party.` This plural form is only used in particular situations. We therefore conclude that category titles that represent instances should be in singular. The grammatical number of the category title is the same as the number of its lexical head. To determine the heads and their numbers, we use the Stanford Parser and the part-of-speech tags assigned during parsing. If one of the category title phrase heads is marked as a plural noun (NNS, NNPS in the Penn Treebank Tagset used by this parser [18]), the category is labeled as `class`, otherwise as `instance`.

Page. Instructions for article authors in Wikipedia provide advice on creating categories: ARTICLES SHOULD BE PLACED IN CATEGORIES WITH THE SAME NAME. An article will therefore have a homonymous category. Because many articles refer to unique entities, the probability that a category containing a page with the same name is an instance is high. We use this as a heuristic to assign `instance` tags to categories.

5 Evaluation

The method presented was used with the Wikipedia version of September 25th, 2006. We use the same Wikipedia version as [5], whose work is briefly described in Section 3, as our research enhances the taxonomy they derived. The gold standard we use is ResearchCyc[8], in which the distinction between `#$Individual` and `#$SetOrCollection` is made for each entity in the repository.

Wikipedia and Cyc overlap on 7860 concepts, of which 44.35% (3486) are `#$Individual` and 55.65% (4374) are `#$SetOrCollection` (in Cyc). This constitutes our evaluation data set.

We first evaluated every method separately on the whole data set. Table 1 shows the results. We compute the reported scores as follows:

$$\text{Prec}_{indiv} = \frac{T_{indiv}}{T_{indiv}+F_{indiv}}$$

$$\text{Prec}_{coll} = \frac{T_{coll}}{T_{coll}+F_{coll}}$$

$$\text{Accuracy} = \frac{T_{indiv}+T_{coll}}{T_{indiv}+F_{indiv}+T_{coll}+F_{coll}}$$

where

- T_{indiv} is the number of nodes correctly classified as `instance` (they are classified as `#$Individual` in Cyc);
- F_{indiv} is the number of nodes incorrectly classified as `instance` (they are classified as `#$SetOrCollection` in Cyc);

[8] http://research.cyc.com/

T_{coll} is the number of nodes correctly classified as `class` (they are classified as `#$SetOrCollection` in Cyc);
F_{coll} is the number of nodes incorrectly classified as `class` (they are classified as `#$Individual` in Cyc).

Method	Prec_{indiv} (%)	Prec_{coll} (%)	Accuracy (%)
`NER`	85.23	76.84	79.69
`Page`	66.10	91.50	75.74
`Capitalization`	85.99	82.44	83.82
`Plural`	66.44	87.99	75.24
`Structure`	56.17	87.21	64.71

Table 1. Results for separate evaluation of methods

Using the results for individual methods, we have designed three different classification schemes, which we evaluate on 10 data sets – 5 rounds of binary random splits of the evaluation data, maintaining the `#$Individual`/`#$SetOrCollection` distribution – a form of cross-validation. We observe the standard deviation in performance over these splits, which will show the stability of performance.

A. Accuracy scheme:
We pick the method with the highest accuracy, and this will constitute a baseline for the evaluation of the combination methods. The best performing method is `Capitalization`.
B. Precision scheme:
We order the five methods according to their precision for correctly identifying instances respectively classes:
   1. `Page` – if the category does not have a corresponding page, it is classified as `class`.
   2. `Plural` – if the title is in the plural, the category is classified as `class`.
   3. `Structure` – if the category has hyponyms, it is classified as `class`.
   4. `Capitalization` – if the title is capitalized, the category is classified as `instance`.
   5. All remaining categories are assigned the `class` label.
C. Voting scheme:
We chose the two methods with the highest precision for classifying instances and the two methods with the highest precision for classifying classes. We combine them to a voting scheme, the remaining categories are classified by all methods ordered by precision.
   1. `Page` & `Plural` – if a category has no corresponding page and if the title is in the plural, the category is classified as `class`.
   2. `Capitalization` & `NER` – if a category title is capitalized and if it is a named entity, the category is classified as `instance`.

3. `Page` – if the page does not have a corresponding page, it is classified as `class`.
4. `Plural` – if the title is in the plural, the category is classified as `class`.
5. `Structure` – if the category has hyponyms, it is classified as `class`.
6. `Capitalization` – if the title is capitalized, the category is classified as `instance`.
7. All remaining categories are assigned the `class` label.

The precision and accuracy averages and the standard deviation scores are presented in table 2.

Method	Precision$_{indiv}$ (%)	Precision$_{coll}$ (%)	Accuracy (%)
A.	85.99±0.54	82.44±0.63	82.82±0.5
B.	90.92±0.41	77.36±0.52	81.64±0.42
C.	89.21±0.46	81.82±0.52	84.52±0.34

Table 2. Precision and accuracy ± standard deviation scores over 10 evaluation runs

5.1 Discussion

The methods used show very good results in distinguishing between classes and instances in Wikipedia category titles. The low standard deviation scores over the multiple runs indicate stable performance of the methods. This allows us to expect similar performance on the full Wikipedia category network.

It is interesting though to see where and why errors occur. A closer inspection of the erroneously classified categories reveal three main causes for these problems.

Preprocessing errors. Some of our methods rely on output of NLP tools – lexical heads of noun phrases and part-of-speech tags. The category titles are quite complex, and some of them pose particular problems to the tagger and parser. Category titles like All India Council for Technical Education are no challenge to be parsed, but it is expected too much to receive correct parsing results for titles like ...And You Will Know us by the Trail of Dead albums.

For an easier handling of the categories inside the system, all category titles are preprocessed by tokenization at the very beginning – All India Council for Technical Education becomes `[All] [India] [Council] [for] [Technical] [Education]`. Punctuation marks are treated as separate tokens. So the name of a Japanese franchise-company .hack becomes `[.] [hack]`. This not only leads to wrong parsing results, it also is not recognized as an instance by the method that checks the capitalization of the category titles.

Recognizing named entities. Named entity recognition is especially difficult when the components of the name are not named entities themselves. For example, BEE TRAIN is a Japanese animation studio, but the NER processes the two parts of the name separately, and neither `bee` nor `train` are tagged as named entities.

Concepts in Cyc. The gold standard we use, Cyc, is manually created. As agreement between human judges on assigning a class or instance label is not perfect, it is to be expected that there are some concepts in Cyc classified as instances respectively classes that do not match our definitions of instances and classes. As an example, we take the concept PHILOSOPHY. According to our definitions, PHILOSOPHY is a class, as it can contain subconcepts like ANALYTIC PHILOSOPHY, APPLIED PHILOSOPHY, EPISTEMOLOGY, etc. In Cyc, it is classified as an instance.

5.2 Resource

We applied classification scheme C that performed best in our evaluation to the 127,124 categories in the Wikipedia taxonomy built by [5]. It classified 15,472 categories as `instance` and 111,652 categories as `class`. To make the results accessible we converted the obtained taxonomy into an RDF Schema file using the Jena Semantic Web Framework[9]. The RDF Schema file includes all of the 111,652 categories labeled as `class` and 13,258 of the categories labeled as `instance`. The reason for the missing instances is that the RDF Schema specification requires for each instance its corresponding class, otherwise the instance cannot be included. The structure of the taxonomy is not perfect, and there are categories labeled as `instance` not directly connected to a category labeled as `class` by an *isa* relation. The taxonomy converted to RDF Schema is available on our web page (http://www.eml-research.de/nlp/download/wikitaxonomy.php).

6 Conclusions

We have presented methods to distinguish automatically between instances and classes in a large scale taxonomy derived from the Wikipedia category network. Towards this end we exploited both structural information from the taxonomy, and naming characteristics and conventions. We have implemented such methods, and evaluated them separately, and then combined them to obtain even higher performance. Combining methods based on individual precision in identifying either instances or classes results in an algorithm that identifies instances with 90.92% precision. The most balanced approach uses voting from several methods, and gives the highest accuracy of 84.52%. From the five methods we presented one can generate a large number of combinations ($\sum_{i=1}^{n} \frac{n!}{i!}$ when each

[9] http://jena.sourceforge.net/index.html

method is used at most once in a combination, n = 5 in our case). We have selected a few, based on individual performance of the methods. A more exhaustive analysis is left for future work.

The inter-judge agreement between human annotators for the task reported in [10] was $\kappa = 0.75$. This shows that the task is not easy, and that we have obtained very high performance with a fully automated approach.

There are multiple advantages of using Wikipedia: it is extremely up-to-date and it is multilingual. With slight modifications our methodology for distinguishing between instances and classes can be applied to other languages as well, provided a parser and a part-of-speech tagger are available.

Future work includes adding the Wikipedia article titles to the taxonomy and introducing the distinction between classes and instances at this level as well. Through the articles we can link to taxonomies in other languages, and ultimately create a huge, multi-lingual, knowledge base, to help users and software agents navigate through the World Wide Semantic Web.

Acknowledgements

We thank Simone Paolo Ponzetto for sharing his system for building the Wikipedia category network and for inducing the taxonomy, and for his feedback and input during various stages of this work. We thank the Klaus Tschira Foundation for financial support.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (2001) 34–43
2. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Mass. (1998)
3. Lenat, D.B., Guha, R.V.: Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Addison-Wesley, Reading, Mass. (1990)
4. Fridman Noy, N., Hafner, C.D.: The state of the art in ontology design: A survey and comparative review. AI Magazine **18**(3) (Fall 1997) 53–74
5. Ponzetto, S.P., Strube, M.: Deriving a large scale taxonomy from Wikipedia. In: Proceedings of the 22nd National Conference on Artificial Intelligence, *Vancouver, B.C., Canada, 22–26 July 2007.* (2007) 1440–1447
6. Woods, W.A.: What's in a link: The semantics of semantic networks. In Bobrow, D.G., Collins, A.M., eds.: Representation and Understanding. Academic Press, New York (1975) 35–79
7. Gangemi, A., Guarino, N., Oltramari, A.: Conceptual analysis of lexical taxonomies: The case of WordNet top-level. In: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, *Ogunquit, Maine, 17-19 October 2001.* (2001) 285–296
8. Oltramari, A., Gangemi, A., Guarino, N., Masolo, C.: Restructuring WordNet's top-level: The OntoClean approach. In: Proceedings of the Workshop on Ontologies and Lexical Knowledge Bases at LREC2002, *Las Palmas, Spain, 27th May, 2002.* (2002) 17–26

9. Miller, G.A., Hristea, F.: WordNet nouns: Classes and instances. Computational Linguistics **32**(1) (2006) 1–3
10. Miller, G., Hristea, F.: Towards building a WordNet noun ontology. Revue Roumaine de Linguistique **LI**(3-4) (2006) 405–413
11. Siegel, S., Castellan, N.J.: Nonparametric Statistics for the Behavioral Sciences. 2nd edn. McGraw-Hill, New York (1988)
12. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 15th International Conference on Computational Linguistics, *Nantes, France, 23-28 August 1992*. (1992) 539–545
13. Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, *College Park, Md., 20–26 June 1999*. (1999) 120–126
14. Harman, D., Liberman, M.: TIPSTER Complete. LDC93T3A, Philadelphia, Penn.: Linguistic Data Consortium (1993)
15. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, *Ann Arbor, Mich., 25–30 June 2005*. (2005) 363–370
16. Klein, D., Manning, C.D.: Fast exact inference with a factored model for natural language parsing. In Becker, S., Thrun, S., , Obermayer, K., eds.: Advances in Neural Information Processing Systems 15 (NIPS 2002). MIT Press, Cambridge, Mass. (2003) 3–10
17. Bunescu, R., Paşca, M.: Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, *Trento, Italy, 3–7 April 2006*. (2006) 9–16
18. Santorini, B.: Part of speech tagging guidelines for the Penn Treebank Project. http://www.cis.upenn.edu/∼treebank/home.html (1990)