Classifying the Wikipedia Articles into the OpenCyc Taxonomy

Aleksander Pohl*

Jagiellonian University
Department of Computational Linguistics
ul. Łojasiewicza 4, 30-348 Kraków, Poland
aleksander.pohl@uj.edu.pl

Abstract. This article presents a method of classification of the Wikipedia articles into the taxonomy of OpenCyc. This method utilises several sources of the classification information, namely the Wikipedia category system, the infoboxes attached to the articles, the first sentences of the articles, treated as their definitions and the direct mapping between the articles and the Cyc symbols. The classification decision made using these methods are accommodated using the Cyc built-in inconsistency detection mechanism. The combination of the best classification methods yields 1.47 millions of classified articles and has a manually verified precision above 97%, while the combination of all of them yields 2.2 millions of articles with estimated precision of 93%.

1 Introduction

The primary goal of this paper is a description of a method for a classification of the Wikipedia articles into the OpenCyc taxonomy. This research is motivated by the fact that the proper classification of entities into types is indispensable for any Information Extraction (IE) system (c.f. Moens [11]).

The strength of IE systems versus traditional text processing might be easily illustrated with the Google Trends service[1]. It allows for a comparison of trends for terms that people enter into the Google search engine. Suppose a person wishes to compare two programming languages: *Ruby* and *Python*. If they are entered, a plot concerning them will be presented. But a quick survey of the results will show, that the comparison covers not only the programming languages, but, due to the ambiguity of Ruby and Python terms, also other meanings. What one could expect from such a system would be at least an option to select only the interesting meanings. In a more sophisticated version of the system the selection should be done automatically based on their shared type – that is a *programming language*.

To fulfil such requirements it is required that during the processing of the text, the terms are disambiguated against some reference resource providing

* This work is partially sponsored by the Faculty of Management and Social Communication of the Jagiellonian University.

[1] http://www.google.com/trends/

meaning for them. What is more, that resource should also provide fine grained types for the disambiguated terms, to allow for the realization of the second part of the scenario. Although we all know that there is such a resource – namely Wikipedia – and that there exists systems such as DBpedia Spotlight [9], AIDA [19] and Wikipedia Miner [10], that disambiguate unstructured text against it, the types that are determined for the Wikiepdia entities in resources such as DBpedia [8] and YAGO [16] are still not perfect. So the aim of this research is to provide better classification of the entities using the OpenCyc taxonomy as the reference resource.

2 Related work

The DBpedia [1, 2, 8] project concentrates on producing RDF triples[2] representing various facts about the Wikipedia entities, such as their categorisation, date of establishment or birth, nationality, sex, occupation and the like. These data are mostly extracted from Wikipedia infoboxes that describe the facts in a structured manner. It also provides its own ontology [2] used to classify the extracted entities. The classification is achieved via manual mapping of the infoboxes into the corresponding DBpedia ontology classes.

YAGO (Yet Another Great Ontology) [16] in its core is much similar to DBpedia – it converts Wikipedia to a knowledge base that may be queried for various facts using a sophisticated query language. The primary difference between these resources is the reference ontology used to categorise the entities. In the case of DBpedia it is its own hand-crafted, shallow ontology, in the case of YAGO these are WordNet [4] and SUMO (Suggested Upper Merged Ontology) [12].

The classification of Wikipedia entities into WordNet is done via the Wikipedia category system, which helps the Wikipedia users to discover related articles. YAGO exploits this system by syntactically parsing the category names and determining their syntactic heads. If the head is in plural, it is mapped to a corresponding WordNet synset. As a result the entity in question is supposed to be an instance of the concept that is represented by the synset.

A different approach is taken by Sarjant et al. in the experiment described in [15]. At the first stage the authors (following [7]) map the Wikipedia articles into symbols from the Cyc ontology [6] and in the next stage, some of the Wikipedia entities that lack corresponding Cyc symbols are classified into the Cyc taxonomy. The mapping is based on various transformations of the article names as well as transformations of the Cyc symbol names. Then a disambiguation is performed based on the semantic similarity measure described in [18]. In the next stage several heuristics (exploiting information encoded in infoboxs and introductory sentences) are used to determine the classification of the articles. At the last stage, the Cyc inconsistency detection mechanism is used to filter out false positives. The first stage yields 52 thousands of mapped entities, while

[2] http://www.w3.org/RDF/

the last 35 thousands of classified entities. As a result approx. 87 thousands of the Wikipedia articles are classified into the taxonomy of Cyc.

3 Current limitations

The short description of DBpedia Spotlight claims that the system is able to recognise 3.5 millions of things and classify them into 320 classes. However, only a half of the Wikipedia articles has an infobox[3] attached and as a result only 1.7 millions of articles are classified withing the DBpedia ontology.

The other thing which is assumed about DBpedia is its perfect classification precision. But this is true only to some extent. E.g. in DBpedia *Algol* is classified both as[4] a *dbpedia-owl:Writer* and a *yago:FlamsteedObjects*. From its description one may find that the entity is a star, but there is a *Writer* infobox in the contents of the article, so the DBpedia classification mechanism assigns a *dbpedia-owl:Writer* class and some other derived classes (such as *foaf:Person* and *dbpedia-owl:Person*).

The DBpedia ontology, with its 320 classes is definitely a small one. Even though a typical IE system defines only a few classes (such as *person*, *organisation*, *place*, etc. cf. [13] for a list of such types), when one wishes to perform moderately-sophisticated IE tasks, such as an automatic cleaning of the extracted data, that ontology is simply too shallow. What is more, the concepts defined in the ontology are not well balanced (e.g. *CelestialBody* has three subclasses: *Planet*, *Asteroid* and *Galaxy* but lacks *Star*).

YAGO seems to be on the opposite end of the ontology spectrum. The conversion of all Wikipedia categories with plural heads into YAGO classes yielded an ontology with 365 thousands of classes[5]. Although this is really an impressive number, most of the classes are over-specified. Consulting the entry for *Gertrude Stein* one will find the following results: *American autobiographers*, *American feminists*, *American poets*, *Feminist writers*, *Jewish American writers*, *Jewish feminists* and more. On the one hand many of the classes in the above example are overlapping, on the other the categories are not decomposed, so searching for *Jews* in YAGO will not yield Gertrude Stein. What is even worse, there is no such category in the ontology[6].

Further investigation into the class system of YAGO will also reveal that the category based classification is also error-prone. Although its authors used some heuristics devised for the removal of the contradicting classifications [3], such contradictions are still present in YAGO. For example *Gertrude Stain* has a type of *Works by Gertrude Stein* and via transitivity of the *type* relation she is classified

[3] The facts concerning Wikipedia are obtained using Wikipedia Miner fed with the Wikipedia dump from 22^{th} of July, 2011, containing 3.6M articles. Statistics for the (latest) DBpedia might be different.

[4] http://dbpedia.org/page/Algol – accessed on 25^{th} of July, 2012

[5] http://www.mpi-inf.mpg.de/yago-naga/yago/statistics.html

[6] Probably due to the fact, that in Wikipedia the *Jews* category includes only subcategories.

as *artifact*, *end product* and *oeuvre*. These are definitely wrong classifications. Even if the majority of the classifications are correct, such inconsistencies should be totally removed, since they introduce contradicting facts into the knowledge base.

To sum up – the available knowledge sources, that classify Wikipedia articles into ontologies still lack some features required from a fully-fledged IE systems. The classification of Wikipedia articles into Cyc was very limited, while the classification provided by DBpedia and YAGO could still be improved.

4 Solution

The proposed solution follows [15] – namely the goal is to classify as many of the Wikipedia articles into the Cyc taxonomy [6] and then use its inconsistency detection mechanism to filter out inconsistent classifications. The primary difference between them is that the first method covers less than 100 thousands of the Wikipedia articles, while the method presented in this paper yields more than 2 millions of classified entities.

The important feature of Cyc (compared to other ontologies like YAGO and DBpedia) is its efficient inferencing engine, which allows for querying the ontology for various sophisticated facts. This makes the development of any Cyc-based system simpler, since there are many built-in API calls, covering navigation through the taxonomy, indexing and inferencing, that would have to be otherwise implemented from scratch.

The contents of Cyc might be roughly divided into two ontological categories: *collections* and *individuals*. The entities from the second category might be instances of entities of the first category and might not have their own instances. They roughly correspond to the entities which are referred to by their proper names. On the other hand the entities of the first category have instances, but might be also instances of other collections. It might be assumed that the first order collections (whose instances are only individuals) correspond to classes (such as *books*, *people*, *numbers* etc.). In these terms Cyc contains approx. 71 thousands of classes[7].

The difference between the DBpedia ontology and Cyc is rather obvious – there are simply more classes (and relations) available. The difference between Cyc and YAGO is more subtle. Cyc also uses reification to a large extent, the feature that was criticised in YAGO, but the reification level is much lower in Cyc, than in YAGO, so none of the classes found in YAGO that describe *Gertrude Stein* will be found in Cyc. As a result the classification might be expressed in canonical form, where each component of the classification type is separated.

But that what makes Cyc particularly helpful for the classification task is the *disjointWith* relation with the corresponding *collections-disjoint?* API call[8]. The information encoded using this relation allows for straightforward detection of inconsistencies in object classification. Assuming that given object is classified

[7] The statistics are provided for OpenCyc version 4.0, released in June 2012.
[8] The description of the Cyc API is available at `http://opencyc.org/doc/opencycapi`

into several Cyc collections, by calling *any-disjoint-collection-pair* one can check if that classification is consistent. In the case that one of the classes is more trusted, the other classes might be accepted and rejected pair-wise.

5 Classification algorithm

5.1 Introduction

The goal of the classification algorithm is a consistent assignment of one or more first order Cyc collections to every Wikipedia article. The nature of the classification depends on the ontological status of the entity described in the article – if it is an object, the classification is interpreted as *instanceOf* relation, if it is a concept, the classification is interpreted as *subclassOf* relation. Telling apart objects from concepts is out of scope of this algorithm.

Unlike the other algorithms that were used to classify Wikipedia articles into an ontology using only one method, this algorithm tries to maximise its coverage by combining several classification methods: *category*-based, *infobox*-based, *definition*-based and *mapping*-based. It is assumed that the results of the methods will overlap, allowing for a reconciliation of the results using both the well developed Cyc taxonomy and the inconsistency detection mechanism.

5.2 Categories

The primary source of classification data is the category system of Wikipedia. The category names are split into segments and the first plural noun is detected. That noun, together with its preceding modifiers (if they exists) is assumed to be the name of a parent *semantic* category (i.e. a category that subsumes the category in question) of the category. Then the ancestor categories of the category are consulted and the category with the same name (if it exists) is selected as a parent semantic category.

Although inspired by YAGO classification algorithm, this method diverges from it in several places. First of all the name is not parsed using a link-grammar parser. The second difference is the more sophisticated semantic parent determination algorithm. It stems from the fact that the single-segment expressions used in YAGO are more ambiguous than the multi-segment expressions. The last difference concerns the inspection of parent/ancestor categories. Although not yet realized, in future this will allow for an extension of the Cyc ontology with meaningful categories defined in Wikipedia.

When the set of root semantic categories is determined, these categories are mapped semi-automatically into Cyc symbols. The name of the category is converted into singular form and then the methods of Wikipedia-Cyc names mapping described in [7] are applied. Usually this will lead to an ambiguous mappings. The Cyc symbols that are not first order collections are filtered. Still in most of the cases there is more than one candidate mapping. Although it is possible to create a method of automatic selection of the best candidate,

since this mapping is the key element of the classification algorithm, the proper mapping is determined manually. This also allows for ignoring mappings that are valid but not meaningful – e.g. words such as *group*, *system* or *collection* have very broad meaning in Cyc and they are not mapped.

5.3 Infoboxes

The second source of article classification were the infoboxes. The author used the classification provided by DBpedia and the mapping between Cyc and DBpedia ontology that is available in the Cyc Semantic Web service[9]. It turned out that many of the DBpedia classes were not mapped into Cyc symbols, so the author manually mapped the remaining classes.

This classification procedure was augmented with a category-based heuristic used to identify people. All the articles that were lacking an infobox, but belonged to the *Living people* category or categories ending with *births* or *deaths* were classified as *Person*. This simple heuristic gave 500 thousands of classifications with confidence comparable to the original infobox method.

The infobox and people-related category classification heuristics have very high level of confidence, since the infoboxes, the categories and the infobox-class mappings are determined manually and the chance for a misclassification is low. This is the reason why the mapping between the Cyc symbols and Wikipedia categories was first tested against the results of this method.

5.4 Definitions

The third method that was used to classify the Wikipedia articles was inspired by methods used to extract *hyponymy* relation from machine-readable dictionaries. Following Aristotle, the definitions in such dictionaries are constructed by indication of *genus proximum* and *differentia specifica*, that is the closest type and the specific feature of the defined entity. This allows for a construction of patterns devised for the extraction of the type of the entity (cf. [5], [15]).

The method used to determine the location of the entitie's type is as follows: the first sentence of the short description of the article that is extracted using the DBpedia extraction framework is tagged using Stanford POS tagger [17]. Then a continuous sequence of adjectives, nouns, determiners and (optional) *of* preposition that follow the first occurrence of *to be* or *to refer* verb is marked as the probable location of the type name. This expression is disambiguated using the improved Wikipedia Miner disambiguation algorithm [14], taking as the disambiguation context all the articles that are linked from the source article.

This method does not follow [15] in using the existing links that are usually present in the first sentence of the definition, since first, there are many articles which lack a link to the article's type in the first sentence and second, the links not always indicate their type (e.g. only the type constituens like *life* and *system* in the *living system* type).

[9] http://sw.opencyc.org

After defining the type-articles, the articles which are not semantically related to any other article with the same type (i.e. their semantic relatedness measure [18] with each article is 0) and lacking Wikipedia categories that include the type name in their names are rejected as false entity-type mappings.

At the end of the procedure the type-articles are mapped to Cyc symbols. In the first step candidate Cyc symbols are generated with the *dentotation-mapper* Cyc API call. This call maps given string to all its interpretations in Cyc. It is called for the name of the article and if it does not succeed the names of the links that have the article as their target are used, in descending frequency order. Only the symbols that are first order collections are registered as candidates.

In the next step the articles that have a Cyc type assigned via the *infobox* classification method are used to order the candidate type-article mappings. In the first pass the equality and in the second pass the subsumption tests are performed. The symbol with the largest number of positive matches is selected.

As the last resort the mapping between the type and the Cyc symbols was determined on the basis of the generality of the Cyc collection (determined as the sum of subsumed collections and covered instances). If a collection was proposed for any of the type names, the most general was selected. If there was no such mapping, but for the covered articles there were any *infobox*-based collections determined, their most specific generalisation was selected.

As a final remark it should be noted that the definition-based classification was applied only to the articles that were not classified as *Person* in the infobox-based classification.

5.5 Cyc mappings

The Cyc-mapping based classification utilizes the direct mappings between Wikipedia articles and Cyc symbols obtained with the methods described in [15] (excluding the cross-validation step, which is performed using the category-based classification). The mapping assumes various types of transformations of the names of Cyc symbols and Wikipedia articles as well as disambiguation strategies. The author used the original results of Sarjant et al. so the reader is advised to consult [15] in order to check the details of the method.

5.6 Cross-validation

The cross-validation of the results generated by the different classification methods allows for a consistent assignment of the types to the articles. It assumes that they have different accuracy and it takes into account the fact, that the number of classified articles varies between the methods. What is more, the results obtained with the more accurate methods are reused by the weaker methods. The methods are cross-validated pair-wise and their order is as follows:

1. categories vs. infoboxes
2. categories vs. definitions
3. categories vs. Cyc mappings

In the first case it is assumed, that the infobox-based classification is more accurate than the category-based one. In the two remaining cases this assumption is inverted. The structure of the cross-validation is as follows:

1. selection of the Cyc symbol that is assigned as the type by the second method (i.e. not category-based) to the Wikipedia article; this is the *primary* Cyc type
2. selection of Cyc symbols assigned to the article by the category-based classification, which are *compatible* with the primary type
3. generalisation of the symbols that were compatible with the primary type; this is the *secondary* Cyc type
4. *compatibility-check* between the secondary type and the Cyc symbols that are assigned to the categories of the article

The selection of the primary Cyc type is usually straightforward – it is the type that was assigned by the method. If there are many such types, the first type of the most specific types is taken. Even though in some cases this leads to a lose of information, the problem is reduced by the generalisation step (3) and usually the category-based classification spans more types than the alternative methods.

The compatibility of the symbols in the second step is determined using subsumption and instantiation relations, via *genls?* and *isa?* Cyc API calls[10]. In the case of the subsumption relation the types are marked as compatible disregarding the fact which of the types is the subsumed and which is the subsuming.

The generalisation of the types that are compatible with the primary type is performed using the *min-ceiling-cols* Cyc API call, which computes the most specific generalisation of a set of collections. The results are filtered using a black-list of types such as *SolidTangibleThing* and *FunctionalSystem* that are too abstract for this task. The black list is created empirically to forbid generalisations that do not posses discriminative power.

The goal of the fourth step is to select the types that will be assigned to the article. This is performed using both the subsumption relation and the disjointness relation. If the category-based type is subsumed or subsumes the secondary type, it is marked as *compatible*. If it is disjoint with the type, it is marked as *incompatible*. Still is status might be *undetermined* if none of the situations occurred.

The side effect of the cross-validation of individual entities is validation of the mappings between the Cyc symbols and the Wikipedia categories. Although the mapping of the root categories was manual, the mapping of the other categories was automatic, thus it introduced errors. Thanks to the cross-validation such erroneous mappings were removed and not exploited in the next cross-validation scenario. Furthermore, the mappings that turned out to be positively verified were used as a sole source of classification for the entities that did not have any types assigned in any of the cross-validation scenarios.

[10] The second call is used only for direct Cyc mappings, since in all other cases the types are always collections.

6 Results

Each variant of the cross-validation procedure yielded a different number of types that were determined as compatible and incompatible for the respective concepts. Table 1 summarizes these numbers. The total number of concepts is the number of concepts for which given classification method assigned at least one type. The number of cross-validated concepts is the number of concepts that have the type determined by the method and at least one category-based type[11]. The classifications denoted as *valid*, were the classifications for which the cross-validation procedure found at least one compatible type and as *invalid* – the classifications that have only incompatible types determined.

The last column indicates the number of valid classifications that were produced by the method for concepts that were not classified by the previous methods. It also indicates the number of classified concepts that were incorporated in the final result.

Table 1. The number of classifications (in thousands) with the respective status produced by each variant of the cross-validation procedure. C_t – total number of classified concepts. C_c – number of classifications that were cross-validated. C_v – number of valid classifications. C_i – number of invalid classifications. Δ – number of classifications included in the final result.

Variant	C_t	C_c	C_v	C_i	Δ
Infoboxes	2188	1712	1471	67	**1471**
Definitions	406	247	154	60	**154**
Cyc mappings	35	25	14	5	**3**
Categories	2470	742	593	—	**593**
Total					**2221**

The results of the classification were verified by two subjects (excluding the author of the article) with some ontological and linguistic training (one being a PhD student of philosophy and the other a person with a bachelor degree in linguistics). Each variant was verified on a distinct set of 250 randomly selected cross-validated classifications with equal number of compatible and incompatible types. The subjects were presented with the names of the entities and their respective types, supplemented with their short descriptions – the first paragraph of the Wikipedia article in the case of the entities and the comment attached to the Cyc symbol in the case of the types.

The subjects had three answers to choose from when deciding if the classification is correct: *yes*, *no* and *not sure*. The third option was left for cases when it was hard to decide if the classification is correct, due to the mismatch of description accuracy level between Wikipedia and Cyc.

[11] In the case of category-only classification, these were the types that were recognized as valid in previous cross-validation scenarios

As a result the precision and recall measures are given separately for cases when both of the subject were confident about their choice and only one of them was confident. In the first case the answer was used to compute the precision and the recall only if both answers were the same, that is there was no adjudication procedure implemented. The precision and the recall were defined as follows:

$$P = \frac{c_{tp}}{(c_{tp} + c_{fp})} \qquad R = \frac{c_{tp}}{(c_{tp} + c_{fn})}$$

where:

- c_{tp} – the number of types determined as *compatible* by the cross-validation procedure and marked as *valid* by *both* of the subjects
- c_{fp} – the number of types determined as *compatible* by the cross-validation procedure and marked as *invalid* by both of the subjects
- c_{fn} – the number of types determined as *incompatible* by the cross-validation procedure and marked as *valid* by both of the subjects

Table 2. The results of the verification of the cross-validated classifications carried out by two subjects on 250 classifications (for each of the cross-validation variants). P – precision for classifications with agreed answer. R – recall for classification with agreed answer. $P_{1/2}$ – precision for classifications with one uncertain answer. $R_{1/2}$ – recall for classifications with one uncertain answer. A – agreement between the subjects. $C_{1/2}$ – percentage of classifications that were confusing for one of the subjects. C – percentage of classifications that were confusing for both of the subjects. # – number of classified concepts (in thousands).

Variant	P	R	$P_{1/2}$	$R_{1/2}$	A	$C_{1/2}$	C	#
Infoboxes	**97.8**	77.2	90.0	78.0	**92.5**	9.7	2.1	**1471**
Definitions	93.5	69.4	**93.9**	68.6	89.0	**5.2**	**0.0**	154
Cyc mappings	94.0	76.4	89.1	71.5	86.1	10.8	**0.0**	3
Categories	81.9	**80.4**	82.1	**78.7**	90.5	10.9	0.8	593
Overall (est.)	**93.3**	**77.5**	**88.2**	**77.5**	**91.7**	**9.7**	**1.6**	**2221**

The results of the verification are presented in Table 2. The testers agreed approx. in 90% of the answers, which means that the verification procedure was meaningful. In approx. 10% of the answers one of the subjects was confused with the classification. It shows that the ontology-based classification is not an easy task, especially if the reference resource is Cyc, making very strict and well defined distinctions, which are sometimes hard to accommodate with fuzzily defined Wikipedia entities.

Comparing the results of the classification to YAGO[12] shows that the combination of the best methods has almost the same precision as in YAGO. However,

[12] http://www.mpi-inf.mpg.de/yago-naga/yago/statistics.html

it should be noted that in the case of the presented algorithm there should be no inconsistent classifications and no compound types, which are both present in YAGO. What is more, Cyc collections are defined more strictly than WordNet synsets.

Comparing the results to DBpedia shows that with a moderately hight precision (93%) we can assign types to more than 2.2 millions of entities, going far beyond the infobox-based classification.

The comparison with the results of Sarjant et al. [15] is harder, since the evaluation procedure was more sophisticated in the second case. However, they reported that the classification was indicated as strictly correct by the majority of evaluators in 91% of the cases. Assuming this is a fair comparison, the presented method surpasses their results both in precision (93% vs. 91%) and coverage (2.2 millions of classified concepts vs. 87 thousands).

7 Conclusions

The precision of the Cyc-based method used to classify the Wikipedia articles depends strongly on the source of the classification information. It is apparent that it is possible to achieve very good classification results (with precision above 97%) for a large number (1.47 millions) of articles using the best method (infobox-based) and also, that with a moderately high precision (93%) we can extend the coverage of the classification.

The sample results of the classification together with the handcrafted mappings are available on the Internet: `https://github.com/apohllo/cyc-wikipedia`. The full result is available upon request. The results of the classification are incorporated into an Information Extraction system, that utilizes the improved Wikipedia Miner algorithm [14]. This system is available at `http://textplainer.com`.

As a final remark we can conclude that Cyc is well suited for the task of detecting the inconsistencies in the classification. The author is going to further utilize this feature in cross-linguistic classification of the Wikipedia articles and automatic, type-base validation of the information extraction results.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. The Semantic Web pp. 722–735 (2007)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia-A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009)
3. De Melo, G., Suchanek, F., Pease, A.: Integrating yago into the suggested upper merged ontology. In: Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on. vol. 1, pp. 190–193. IEEE (2008)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)

5. Kazama, J., Torisawa, K.: Exploiting wikipedia as external knowledge for named entity recognition. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 698–707 (2007)
6. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
7. Medelyan, O., Legg, C.: Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. In: Proceedings of the WIKI-AI: Wikipedia and AI Workshop at the AAAI. vol. 8 (2008)
8. Mendes, P., Jakob, M., Bizer, C.: Dbpedia for nlp: A multilingual cross-domain knowledge base. LREC-to appear (2012)
9. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. ACM (2011)
10. Milne, D., Witten, I.: Learning to link with Wikipedia. In: Proceeding of the 17th ACM conference on Information and knowledge management. pp. 509–518. ACM (2008)
11. Moens, M.: Information extraction: algorithms and prospects in a retrieval context, vol. 21. Springer-Verlag New York Inc (2006)
12. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. pp. 2–9. ACM (2001)
13. NIST: Automatic Content Extraction 2008 Evaluation Plan (ACE08) (2008), `http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf`
14. Pohl, A.: Improving the Wikipedia Miner Word Sense Disambiguation Algorithm. In: Proceedings of Federated Conference on Computer Science and Information Systems 2012. IEEE (to appear)
15. Sarjant, S., Legg, C., Robinson, M., Medelyan, O.: All you can eat ontology-building: Feeding wikipedia to cyc. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01. pp. 341–348. IEEE Computer Society (2009)
16. Suchanek, F., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
17. Toutanova, K., Manning, C.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13. pp. 63–70. Association for Computational Linguistics (2000)
18. Witten, I., Milne, D.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA. pp. 25–30 (2008)
19. Yosef, M., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: Aida: An online tool for accurate disambiguation of named entities in text and tables. Proceedings of the VLDB Endowment 4(12) (2011)