

“All You Can Eat” Ontology-Building: Feeding Wikipedia to Cyc

Samuel Sarjant*, Catherine Legg†, Michael Robinson‡ and Olena Medelyan§
The University of Waikato, New Zealand

*Email: sjs31@cs.waikato.ac.nz †Email: clegg@waikato.ac.nz

‡Email: mdrar1@waikato.ac.nz §Email: olena@cs.waikato.ac.nz

Abstract

In order to achieve genuine web intelligence, building some kind of large general machine-readable conceptual scheme (i.e. ontology) seems inescapable. Yet the past 20 years have shown that manual ontology-building is not practicable. The recent explosion of free user-supplied knowledge on the Web has led to great strides in automatic ontology-building, but quality-control is still a major issue. Ideally one should automatically build onto an already intelligent base. We suggest that the long-running Cyc project is able to assist here. We describe methods used to add 35K new concepts mined from Wikipedia to collections in ResearchCyc entirely automatically. Evaluation with 22 human subjects shows high precision both for the new concepts’ categorization, and their assignment as individuals or collections. Most importantly we show how Cyc itself can be leveraged for ontological quality control by ‘feeding’ it assertions one by one, enabling it to reject those that contradict its other knowledge.

1. Introduction

The field of Artificial Intelligence was widely perceived to have stalled in the early 80s due to computers having no ‘common-sense’, causing an inescapable roadblock of brittle reasoning, inability to understand natural language and related problems. Thus the Cyc ontology project was conceived and funded massively by the US government. Its purpose was to codify in a giant general knowledge base, “the millions of everyday terms, concepts, facts, and rules of thumb that comprise human consensus reality”, to a point where the system could begin to learn on its own [1][2]. Doug Lenat, project leader, estimated in 1986 that this would take 5 years (350 person-years) of effort and 250 000 rules. After 22 years (> 900 person-years) of effort the goal is still unmet, and the project is widely perceived to have failed and produced a white elephant.

However we believe that Lenat’s vision was not misguided, his timeline was just too short. A crucial, unforeseen recent development on the Web is the vast, ever-improving, free *user-supplied* knowledge repositories, comprising far more than 900 person years’ effort, which have sprung up in the last few years — most notably the astounding

Wikipedia. In this paper we draw on cutting-edge research in mining Wikipedia to automatically augment the Cyc ontology at an unprecedented rate. We effectively increase the common-sense knowledge in ResearchCyc by 30%, whilst maintaining an impressive accuracy. A majority of our human evaluators judged that the placement of our new concepts within Cyc collections was ‘correct’ 88% of the time, and ‘correct’ or ‘close’ 90% of the time.

For better or worse Cyc maintains a strict metaphysical distinction between individuals (e.g. *Fido*) and collections (e.g. *Dog*), so any concepts automatically generated from Wikipedia, which does not make such distinctions, must be automatically assigned individual or collection status. We solve this problem via a mixed suite of heuristics.

Where other automated ontology-building projects merely ‘dump’ assertions together in a large resource, we obtain quality-control by ‘feeding’ assertions to Cyc one by one, enabling the system to reject those that contradict its other knowledge. We believe that Cyc is the only ontology currently sophisticated enough to be able to perform this function, which arises from the combination of its principled taxonomic structure and purpose-built inference engine. As our methods make use of Cyc’s common-sense knowledge for semantic disambiguation, the substantial addition we have now made to this common-sense knowledge can be bootstrapped to produce further results. Our results are freely available at <http://wdm.cs.waikato.ac.nz/cyc/portal/>.

2. Related work

Milne and Witten [3] use Wikipedia to obtain an automated measure of semantic relatedness for any two concepts. Thus *river bank* and *bank robber* are 0% related, *Barclays Bank* and *Bank of England*, 61%. Their method differs from other similar research in using only Wikipedia’s internal hyperlinks, as compared to, for example, the Explicit Semantic Analysis of Gabrilovich and Markovitch [4] which uses the full text of Wikipedia articles, and Ponzetto and Strube [5], which uses the Wikipedia category network. In Medelyan and Legg [6] this measure is used to map Wikipedia articles to Cyc terms. We build our work on their research.

The recent explosion in free user-supplied Web content has drawn many researchers into automated ontology building. An early project is YAGO [7][8]. The project maps

Wikipedia’s leaf categories onto the WordNet taxonomy of synsets, adding articles belonging to those categories as new elements, then defines a mixed suite of heuristics for extracting further relations to augment the taxonomy. Much useful information is obtained by parsing category names, for example extracting relations such as *bornInYear* from categories ending with *birth* (e.g., *1879 birth*). The result has good taxonomic structure, and may be queried online via SPARQL and Linked Data, but no formal evaluation of its accuracy seems to have been performed so far.

A much larger, but less formally structured, project is DBpedia [9][10][11]. This project transforms Wikipedia’s template information (most notably infoboxes) into a vast set of RDF triples (103M). The primary goal of this project is to provide a giant open dataset and to connect with other open datasets on the web. Thus there is no attempt to place facts in the framework of an overall taxonomic structure of concepts, and many semantic relationships are obscured. An example is the lack of a relationship between *New Zealand* the article and the category, or the use of many redundant versions of the same relation within different infobox templates, such as *birth_date*, *birth* and *born*.

Recently the DBpedia project has released the much more structured DBpedia Ontology. This was generated by manually reducing the most common 350 Wikipedia infobox templates to 170 ontology classes and the 2350 template relations to 940 ontology relations, which are asserted onto 882 000 separate instances. No formal evaluation of the accuracy of this resource has been reported so far in the literature so we perform one and evaluate our method against it (Section 6).

The European Media Lab Research Institute (EMLR) has been building an ontology from Wikipedia in stages, starting with Wikipedia’s category link network. First they identify and isolate the *isA* relations from the other links between categories (which they call *notIsA* relations) [5]. Then they divide the *isA* relations into *isSubclassOf* and *isInstanceOf* using a number of different overlapping heuristics [12]. They also divide the *notIsA* domain into more specific non-taxonomic relations (e.g. *partOf*, *bornIn*) by parsing category titles and also adding facts derived from the articles in those categories [13]. The final result consists of 9M facts indexed on 2M terms in 105K categories.¹ They evaluate accuracy of the first two stages of their project using ResearchCyc as a gold standard, reporting for the first stage a precision of 86.6%, and for the second, 82.4% for their best method.

Freebase, a collaborative knowledge base produced by Metaweb, also contains many concepts and relations automatically mined from Wikipedia. No information is available in the research literature on the algorithms that generate it,

or any formal evaluation, and no tools for inferencing over the data appear to exist currently.

If we turn now to projects which seek to automatically augment the Cyc ontology: there are some limited efforts from Cycorp. Early work [14] mapped in SENSUS, WordNet, SNOMED, the CIA World Factbook and some other similar resources, but required interactive clarification with subject-matter experts. Similarly, Matuszek et al. [15] extended Cyc by querying a search engine, parsing the results, and checking for consistency with the Cyc knowledge base. Each entry then required a human check, thus only 2000 new assertions were added. Taylor et al. [16], report interesting work on determining where to enter new facts into the Cyc microtheory structure automatically using machine learning techniques (both Bayesian and Support Vector machines). They report precision and recall of 98% although only over 30 microtheories (a tiny subset of the whole), and do not report actually entering any new facts into Cyc as a result of this research.

Outside Cycorp, however, uses of ResearchCyc in automatic ontology-building, apart from its use by the EMLR group as a testset mentioned above, were slow to arrive, until Medelyan and Legg [6] mapped 52 690 Cyc terms to semantically equivalent Wikipedia articles, with a precision of 93%, but we re-evaluate these mappings in our work (Section 6). The algorithm used in this research will be described in detail in the next section.

3. Improved mappings

The mappings described in Medelyan and Legg [6] are impressive, but have some areas of weakness which we address and improve upon.

3.1. Previous mapping algorithm structure

Medelyan and Legg proposed four separate stages, which we codename A, B, C, and D. Stages A–C were used to find mappings from each Cyc term to a Wikipedia article or articles, and Stage D was used to disambiguate mappings into a one-to-one relationship.

Stage A looks for a single article with a title which exactly matches the Cyc term (after the term is converted into regular English). Stage B looks for an article using the Cyc term’s synonyms and Wikipedia redirects. Stage C attempts to find a matching article using both synonyms and related Cyc terms. First a ‘context’ set of articles is created from mappings to Cyc terms immediately surrounding the key term in the Cyc ontology. Then the candidate article with the most semantic relatedness to that context set of articles is used as the result, drawing on methods developed by Milne and Witten [3].

Once all possible mappings are found for each Cyc term, Stage D is used to disambiguate multiple mappings of Cyc

1. Downloadable at <http://www.emlr-research.de/english/research/nlp/download/wikirelations.php>

Table 1. Manual accuracy results before applying Stage D for the new algorithm.

	Mapped	Correct	Precision
Total	221	172	77.8
Stage A	105	102	97.1
Stage B	70	44	62.9
Stage C	46	26	56.5

terms to a Wikipedia article. The first method is to remove all mappings that are significantly less semantically-related to the context than the best mapping. If there are still multiple mappings, a further test culls terms known to be disjoint with the most similar term. Disjointness is found using Cyc’s common sense knowledge concerning which classes can have no shared instances. For instance the class of coins, which are non-living physical tokens, is disjoint with the class of dogs, which are living creatures.

3.2. New mapping algorithm structure

We have improved the mapping algorithm, and added code for creating new Cyc terms from Wikipedia pages. The new algorithm utilizes the Wikipedia Miner,² which provides easy access to Wikipedia’s structured and semi-structured information.

Stage A: This stage still looks for a one-to-one match between a Cyc term and a Wikipedia article title, but was improved by checking for alternative capitalizations and removals of ‘The’ (the latter allowing terms such as #*\$Batman-TheComicStrip* to obtain a correct mapping to *Batman (comic strip)*).

Stage B: Mappings are still found using synonyms, but the chosen article is determined by majority vote among the possible results. This stage also utilises a method from Wikipedia Miner which finds the most likely Wikipedia article given a term, rather than finding articles by exact title matching.

Stage C: This stage still selects the most relevant article from a group of articles, but the relevancy and the group is determined in a different way, using a Wikipedia Miner method which compiles a weighted list of all semantically-related articles to a plain-text synonym of the Cyc term, using the ‘context’ articles defined in the previous method as guidance.

To find the single Stage C result, we compare the top three weighted articles against each related article individually, rather than compare synonyms against the related articles. The article with the highest average semantic-relatedness to each of the related articles is the chosen mapping.

The new mapping algorithm was manually tested on a small dataset of 700 concepts to estimate the effectiveness of each stage (Table 1). The results suggest that Stage A is

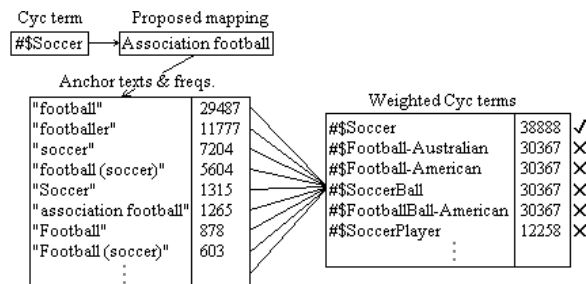


Figure 1. Finding a Cyc term from an article’s anchor terms.

not only the most frequently used Stage, but also the most accurate. In order to increase the accuracy of Stage B and C, a new Stage D was implemented.

Stage D: The new Stage D not only disambiguates multiple mappings to the same article, but also removes many incorrect mappings. It works by ‘double-checking’ the mappings obtained through Stage A, B, or C by finding the Cyc mapping from a given Wikipedia article. For example, the term #*\$DirectorOfOrganisation* incorrectly maps to the article *Film director*, but when we attempt to find a Cyc term from *Film director* we get #*\$Director-Film*. We only accept mappings which run both ways. Thus we employ the following new methods:

Reverse Stage A: This stage simply looks for a Cyc term with the same name as the article (after its title has been converted into Cyc form), adding ‘The’ if necessary.

Reverse Stage B: An article’s synonyms are found using its most frequent anchor terms (incoming link text) and finding a clear majority Cyc term among the synonym mappings. Each anchor term has a frequency of occurrence, which is used to weight the Cyc term/s found using the anchor text. The Cyc term that occurs in the majority of mappings and is clearly weighted more than the other majority terms is used as the result (Figure 1).

If the reverse mapping is found to be the same as the original Cyc term, the mapping is accepted. If it is different, ambiguous, or not found, the mapping is not accepted as correct, but is not completely discarded (see Section 4.2). This double-checking strategy increases precision but reduces the number of mappings by 43%. However, having accurate mappings is critical to the next step of the algorithm, which creates new Cyc terms as ‘children’ (hyponyms) of Cyc collections mapped to Wikipedia articles.

4. Augmenting the Cyc ontology

Having obtained mappings between concepts in Cyc and Wikipedia, the next step is to increase the size of Cyc using information from Wikipedia. We pursue this in two ways: increasing Cyc’s breadth by adding information from Wikipedia into known Cyc constants, and increasing Cyc’s depth by adding new terms into the ontology.

2. Wikipedia Miner homepage: <http://wikipedia-miner.sourceforge.net/>

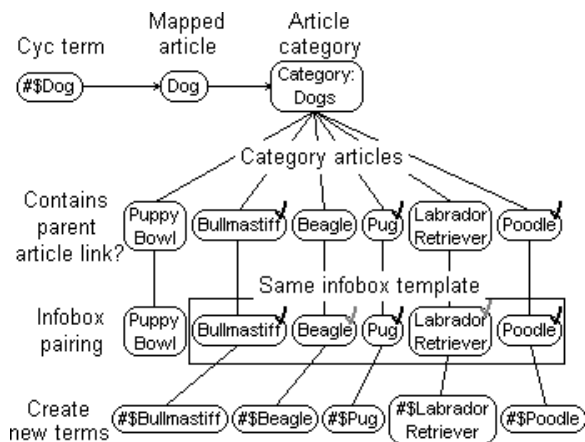


Figure 2. A simplified diagram of the child creation process. Each article is parsed for parent links and also classified by common infoboxes.

Toby Foster



Figure 3. Parsing possible parent links from an article's first sentence.

4.1. Finding possible children

We increase depth by creating new child terms for Cyc collections from Wikipedia articles harvested from relevant Wikipedia categories. Most of these categories were identified by finding Wikipedia articles which: **i)** are mapped to Cyc collections, **ii)** have equivalent Wikipedia categories. Many articles in Wikipedia have categories of the same or similar name. This applies to about 20% of the articles which map to Cyc collections. We also used title matching to identify some further relevant categories.

Wikipedia's category structure is not as well-defined as Cyc's collection hierarchy, containing many merely associatively-related articles. For example, the article *Dog* has an equivalent category: *Dogs*, but to say that every article within that category is a hyponym of the Cyc collection #Dog, would result in inclusion of *Fear of dogs* and *Puppy Bowl* in #Dog. We therefore cannot harvest articles blindly from categories, we need to identify the correct candidate children.

4.2. Identifying candidate children

If a Cyc term T_{parent} has a mapping to an article which has an equivalent category, then the articles under that category are potential candidate child articles A_{child} . If A_{child} is already mapped to a Cyc term T_{mapped} , we assert T_{mapped} as a child of T_{parent} . If there is no mapping, we

Table 2. The regular expressions used to parse an article's first sentence.

Regex format	Example
X are a Y	Bloc Party are a British...
X is one of the Y	Dubai is one of the seven...
X is a Z of Y	The Ariegeois is a breed of dog...
X are the Y	The Rhinemaidens are the three...
Xs are a Y	Hornbills are a family of bird...
Xs are Y	Bees are flying insects...
The X is one of the Y	The Achaeans is one of the collective...
X is a/the Y	Batman is a fictional character...
X was/were a Y	Kipchaks were an ancient Turkic...

apply the following tests to determine if the article is a candidate child, then, if successful, create a new Cyc term T_{child} and assert it as a child of T_{parent} (Figure 2).

Link parsing: The first and most common method for checking if a category article is a candidate child is to parse its first sentence for a link to the parent article. However we only check links that follow directly after a matching regular expression, give or take some punctuation and one or two arbitrary words. For example, in Figure 3, *Toby Foster* would be considered a child candidate if it was found under the *Comedians* category.

The set of regular expressions was created manually from the most frequently occurring structures seen in Wikipedia article first sentences (see Table 2). This technique identifies 57% of our candidate children.

Infobox pairing: Although link parsing finds the majority of candidate child articles, many other potential child articles are not considered because they do not contain an explicit link to the parent article. For example *Rat flea*, found under *Category: Fleas* begins: "The *Oriental rat flea* is a [[parasite]] of [[rodents]]", and although it is clearly a type of #Flea, it is not classified as a one. The article *Rat flea* contains an infobox of the same template (i.e. `template:taxobox`) as those found in other confirmed children (via link parsing) of #Flea. If at least 90% of the confirmed children share the same infobox as the current potential child article, then that article is classified as a child as well. This method identifies 43% of candidate children.

4.3. Determining individual/collection status

A key feature of Cyc is that it classifies all entities which are not relations as either *individuals* or *collections* (see Section 1). Wikipedia does not, so we must automatically determine an article's status before adding it as a new entity in Cyc. We do this by using an ordered list of heuristics, most of which are derived from [12].

Equivalent category: A basic method for determining if an article represents a collection is by checking if it has an equivalent category. For instance, the article *Dog* has an equivalent category: *Dogs*, therefore, it represents

Collection : [WrestlingRing](#)

Bookkeeping Assertions :

 ([myCreationTime](#) [WrestlingRing](#) 19920918) in [BookkeepingMt](#)


GAF Arg : 1

Mt : [UniversalVocabularyMt](#)

isa :  [ExistingObjectType](#)

geils :  [SportsPlayingArea](#)

Mt : [WikipediaToCycDataMt](#)

comment :  "A wrestling ring is the ring stage that professional wrestlers wrestle in."

salientURL :  "http://en.wikipedia.org/wiki/Wrestling_ring"

Mt : [WikipediaToCycLexicalMt](#)

 ([synonymousExternalConcept](#) [WrestlingRing](#) [Enwiki](#) 20080727 "5160881")

Figure 4. A Cyc stub (created in 1992) is found and documented by our algorithm.

a collection (of dogs). This method is somewhat fallible, however (e.g. *Category: New Zealand*). Approximately 8% of children were classified with this method.

Infobox relation: Each infobox in Wikipedia contains a set of relations (such as *'birth_date'* or *'species'*). Many of these relations only make sense when the article is an individual or collection. For example, the article *Dog* would not have the relation *'birth_date'* within its infobox, but may have *'species.'* We have manually created a set of mappings for the most frequent relations and their corresponding status values (either individual or collection). If an article has an infobox, each relation is scanned and the total number of status values is recorded. The status with at least 75% majority is assigned to the article. Approximately 41% of children were classified with this method.

Syntactic structure: We have also assigned an individual or a collection status to each of the regular expressions listed in Table 2, although most expressions cover both collections and individuals and cannot be used to determine status. Approximately 7% of children were classified with this method.

Article title: If previous methods are unsuccessful, we analyze the article's title. If the last word is capitalized, then the article is probably an individual (e.g. *Bill Gates*). If it is in lowercase, then the article is probably a collection (*Aircraft carrier*). This heuristic only applies if the title has two or more words. Approximately 31% of children were classified with this method.

If an article's status cannot be determined, then we default it as an individual and mark it for later manual assignment. This applies to approximately 13% of children.

4.4. Adding further information

We have increased the 'depth' of Cyc by identifying mappings between Wikipedia articles and corresponding Cyc terms. We have also identified the individuality status of

Wikipedia articles that are not in Cyc. Now we can increase the 'breadth' of Cyc with more information.

When a child is added to Cyc as a new entity, it is given a single parent collection. However, the first sentence of an article may contain more than one relevant link, allowing us to assign further parents to the new children. In Figure 3, if the article *Toby Foster* was found within the *Comedians* category, then it would be considered a candidate child of *Comedians*. But it could also be asserted that *Toby Foster* is *British*, an *actor* and a *radio personality*, if those linked articles are mapped to corresponding Cyc collections.

When a mapping is found, or a new child is created, the Cyc term gains a new `#$comment` (Cyc's documentation predicate) from the Wikipedia article's first sentence, and new synonyms (stored using the Cyc predicate `#$termStrings`) from the article title and any bold text present in the first paragraph. For example, the article *Jumping spider*, which begins, "The **jumping spider** family (*Salticidae*) contains..." would create two synonyms: "Jumping spider" and "Salticidae." We also link to the article itself using the Cyc predicates `#$synonymousExternalConcept` to record the article's internal ID, and `#$salientURL` to record the page URL.

In this way, as well as adding over 35K new concepts to the lower reaches of the Cyc ontology, each with an average of 7 assertions, we also flesh out many existing Cyc stubs. For instance approximately 50% of comments made on existing Cyc concepts add a comment where there was none previously (see Figure 4).

4.5. Bootstrapping

As noted above, our Stage C makes use of the ontology surrounding a given Cyc term to perform semantic disambiguation when finding mappings and new children. It follows that adding to the Cyc ontology might make further semantic disambiguation and yet further mappings and children possible. We tested this hypothesis by running our algorithm again, with no changes whatsoever, on a subset (10%) of the enlarged Cyc ontology, and were delighted to derive 1661 entirely new children. This extrapolates to an estimated 16K new children across the whole Cyc (approximately 46% of the size of the set derived by the first running of the algorithm). Achieving bootstrapping of a system's understanding is a long-held goal within AI research.

5. Cyc's ontological quality control

Much of the common-sense knowledge Cyc currently possesses lies in its knowledge of disjointnesses between its many collections, thus for instance if one tries to assert into Cyc that Bill Gates is a parking meter, it refuses

```

BillGates is known not to be an instance of
  ParkingMeter in mt WikipediaToCycDataMt.
sbhl conflict: (isa BillGates ParkingMeter) TRUE
               WikipediaToCycDataMt
because: (isa BillGates MaleHuman)
         True-JustificationTruth
(genls MaleHuman MaleAnimal) TRUE
(genls MaleAnimal Animal) TRUE
(genls Animal AnimalBLO) TRUE
(genls AnimalBLO BiologicalLivingObject) TRUE
(disjointWith BiologicalLivingObject
  Artifact-Generic) TRUE
(genls Technology-Artifact Artifact-Generic) TRUE
(genls MechanicalDevice Technology-Artifact) TRUE
(genls ParkingMeter MechanicalDevice) TRUE

```

Figure 5. Cyc uses its knowledge of disjointness among collections to reject an ontologically faulty assertion.

explaining that as Bill is a ‘biological living object,’ a parking meter is an ‘artifact,’ and those two collections are known to be disjoint, it cannot be true (see Figure 5). We initially attempted to build disjointness tests into the code to preprocess new assertions for accuracy. However these attempts were gradually relaxed in favour of just feeding assertions to Cyc and relying on it to ‘regurgitate’ those that were ontologically unsound.

As we added new knowledge to Cyc, we gathered all the assertions Cyc was rejecting in a file for inspection. We found that overall Cyc had rejected over 4300 assertions, roughly 3% of the total. Manual inspection suggests that 96% of these are true negatives, assertions which are actually incorrect, for example:

```

($#isa #$CallumRoberts #$Research)
($#genls #$Pony #$GymHorse)

```

This compares very favorably with the precision of the assertions which made it into Cyc, as measured by our formal evaluation.

6. Evaluation

We used an online form to evaluate both the new mappings and the new children created by the algorithm. The evaluation consisted of 400 questions, 200 for each task. 22 volunteers participated in the evaluation, each answering at least 100 questions. The authors of this paper did not participate in the evaluation.

We have assessed the *precision* in each evaluation scenario by calculating the number of positively rated questions in relation to the total number of answered questions in that scenario. It was infeasible to compute the recall, as this would imply manual inspection of over 2.5M concepts in Wikipedia and over 100 000 concepts in Cyc.

For both evaluation tasks, the concepts were hyperlinked to a page displaying the relevant Wikipedia page and a description of the new child or mapped term. The following sections describe the evaluation of the two scenarios, each putting the presented algorithm in contrast to previous work.

Table 3. Evaluation results for the mappings data.

Case	1	2	3	4	5	6
Old mappings	0.65	0.83	0.99	0.99	0.99	1.00
New mappings	0.68	0.91	1.00	1.00	1.00	1.00

The mappings created by Medelyan and Legg [6] were used as a baseline for the mapping evaluation task. As a baseline for the new children, we used child-parent pairs extracted randomly from the DBpedia Ontology. Note that both baselines are very strong, as the previous mappings were 93% accurate and the DBpedia samples were manually reviewed by human evaluators.

6.1. The inter-rater agreement

The inter-rater agreement was calculated using Cohen’s Kappa statistic (κ) [17]. The kappa values were computed pairwise and then averaged across all evaluators. To ensure reliable responses, for each evaluation scenario, responses from evaluators with $\kappa \leq 0$ were excluded, as such scores indicate agreement expected by chance in these settings.

The average inter-rater agreement was greatest for the new mappings: 0.26, or “fair” agreement. There was also “fair” inter-rater agreement for the old mappings: 0.22. There was only “slight” agreement on the assessment of the DBpedia pairs (0.18) and the new children (0.15) indicating that this evaluation task was less straightforward. The average inter-rater agreement for individual/collection assignment was 0.32, or “fair”. The fair and the slight agreement indicates that the evaluation was meaningful overall.

6.2. The quality of the new mappings

The possible answers per mapping were:

- “correct” — if the concepts were equivalent:
e.g.: ‘Bench seat is equivalent to CarSeat-Bench’
- “close” — if nearly equivalent, but not quite:
e.g.: ‘Schedule (workplace) is equivalent to Schedule’
- “incorrect” — if entirely unrelated:
e.g.: ‘Nissan Forum is equivalent to RomanForum’

We have considered six cases: The first three indicate the mappings were *correct*: as agreed by all evaluators (case 1), by over a half of evaluators or their majority (case 2), and by at least one evaluator (case 3). In further three cases, we computed how many evaluators assigned “correct” or “close”, indicating that the mappings were *not incorrect*: agreed by all (case 4), by the majority (case 5) and by at least one evaluator (case 6).

Table 3 shows the percentage of evaluators in each of these cases. The majority agreed that the new mappings were more accurate: strict full agreement on 68% new versus 65% old mappings. There was a greater difference in the agreement by the majority, with 91% of new mappings being

Table 4. Evaluation results for the child concept data.

Case	1	2	3	4	5	6
DBpedia children	0.58	0.81	0.99	0.98	0.99	0.99
New children	0.57	0.88	0.99	0.90	0.90	1.00

assigned correct by at least half of all evaluators, 83% for old mappings.

A paired samples t-test was performed comparing the responses for new mappings ($M = 66.9$, $SD = 8.89$) with the responses for old mappings ($M = 63.8$, $SD = 9.77$). This test was found to be statistically significant, $t(9) = 4.72$, $p < 0.01$. This indicates that the new mappings are significantly better than the old mappings.

6.3. The quality of the new child concepts

The second task was to evaluate the algorithm’s categorization. For each newly created Cyc entity, we asked the evaluators whether it belongs to its given Cyc collection, and whether the new concept is an individual or a collection.

The evaluators were presented with 100 new children and their Cyc parents, and 100 DBpedia child-parent pairs, and for each pair, asked to assign:

- “correct” — if a concept A was a child of concept B: e.g.: ‘WXJC (AM) is a RadioStation’
- “close” — if A was related to B, but not hierarchically: e.g.: ‘Lampsilis is a Species’ (Lampsilis is a genus)
- “incorrect” — if A and B were not semantically related: e.g.: ‘10 004 Igormakarov is a Planet’

We used the same six cases as in Section 6.2 to assess the quality of the new children.

Table 4 summarizes the results. 57% of the new children and 58% of the DBpedia pairs were assessed as “correct” by all evaluators, and 88% and 81% of the new children and DBpedia pairs respectively were assessed as “correct” by the majority. Overall, the DBpedia pairs were judged to be more accurate, however the precision of the new children was very close to that of the DBpedia pairs.

The evaluation of automatic categorization of entities into collections and individuals shows that: 29% of the new children’s status was assessed as correct by all evaluators, 68% by the majority, and 84% were assessed as correct by at least one evaluator.

A paired samples t-test was performed comparing the responses for new children ($M = 86.8$, $SD = 13.69$) with the responses for DBpedia ($M = 83.7$, $SD = 14.66$). This test was found to be statistically significant, $t(9) = 2.81$, $p < 0.05$. This indicates that the children terms assigned with our method are significantly better than those listed in DBpedia.

6.4. Discussion

The analysis of the results shows that DBpedia’s parent-child pairs were more accurate in cases 4 to 6: fewer

subjects thought that the pairs were incorrect. However, more subjects agreed that pairs produced by our method are correct (in cases 1 to 3). We have observed the following problem with DBpedia classifications: DBpedia is manually assembled, but its classes seem heavily reliant on the names of Wikipedia infobox templates, which are often very broad. For example, the template Planet is used to describe all kinds of astral bodies, and Species is used to describe both species and genera. As a result, DBpedia classifies Moon as a Planet and Aa (plant) as a Species.

Our algorithm does not suffer from the same limitations. Instead it makes use of many different aspects of a given Wikipedia page as well as Cyc’s rich ontological structure and inferencing capabilities. Newly created concepts are classified into more precise collections than DBpedia’s, e.g. Oregano is a #Herb-HumanUse; Pelusios is a #Turtle; Lapid is an #IsraeliSettlement.

Our results for classifying newly added terms into collections and individual do not compare well against those reported in [12], where a precision of 82.4% was achieved. However, in [12] the relations were tested on terms that already exist in Cyc and were encoded as collections and individuals by professional ontologists. Our method was tested on terms that are new to Cyc, which were evaluated by untrained human subjects. Deciding whether a concept is an individual or collection is a difficult task, and can be confusing for someone without ontology-building experience.

7. Future work

In the short-term, a relatively quick and easy first addition to our results would be to add the assertions contained in infoboxes on mapped Wikipedia articles (such infobox data being conveniently available in DBpedia) to the Cyc terms they are mapped to. We estimate that by these means we could add around 35K new assertions to Cyc. Secondly, in this research we have largely restricted ourselves to harvesting from categories which have equivalent and mapped articles, a subset of all categories which exist in Wikipedia. We could widen this to crawl over the whole Wikipedia gathering children by parsing the first sentence of every article and searching for mapped Cyc collections in appropriate links, though there could be a potential loss in accuracy. The bootstrapping potential we have demonstrated demands to be investigated further, in particular, how many iterations of the algorithm will continue to produce new results?

In the long-term, we would like to build a tool to automatically update Wikipedia edits into Cyc (the DBpedia project has already implemented something similar for their dataset). We would also like to explore the use of Wikipedia, and in particular the quantitative measure of semantic relatedness its vast size makes possible, to perform quality control and/or augmentation on Cyc. For instance we could

use quantitative measures of semantic relatedness to suggest that Cyc concepts that are too close in meaning be merged, or conversely to suggest that new disjointness assertions be made where concepts are entirely different.

On the evaluation side, it would be valuable to do a more thorough relative study of the accuracy of automatically built ontologies, which would also include YAGO and the EMLR group's recent efforts. The lack of overall comparative studies and a gold standard constitutes a current immaturity in this fascinating research field.

8. Conclusions

We have presented a new algorithm for mapping Wikipedia to Cyc terms and a new approach for extending Cyc with new concepts. Compared to previous work [6], the accuracy of mappings was improved from 83% to 91%. The Cyc ontology was entirely automatically extended by 35K new concepts mined from Wikipedia. Additionally each new concept was categorized as an instance or a subcollection. Most excitingly, Cyc itself was leveraged for ontological quality control by 'feeding' it assertions and allowing it to 'regurgitate' those that are ontologically unsound. Cyc is arguably the only ontology currently sophisticated enough to be able to perform such a 'digestive' function. Perhaps a traditional fixation of AI researchers on realizing the intelligence of the brain has caused us to overlook more humble yet genuine steps towards the AI vision which might be gained by realizing the intelligence of the stomach.

9. Acknowledgements

We would like to thank Professor Ian Witten, David Milne, Dr. Nicola Starkey, Associate Professor Bernhard Pfahringer, the University of Waikato Summer Research Scholarship Program, and all the human volunteers who took part in the evaluation.

References

- [1] D. B. Lenat and R. V. Guha, *Building large knowledge-based systems; representation and inference in the Cyc project*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [2] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Commun. ACM*, vol. 38, no. 11, pp. 33–38, 1995.
- [3] D. Milne and I. Witten, "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links," in *Proc., Workshop on Wikipedia and AI, AAAI08*, 2008.
- [4] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 6–12.
- [5] S. Ponzetto and M. Strube, "Deriving a large scale taxonomy from Wikipedia," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 1440.
- [6] O. Medelyan and C. Legg, "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.
- [7] F. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*. ACM New York, NY, USA, 2007, pp. 697–706.
- [8] —, "Yago: A large ontology from Wikipedia and Wordnet," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 3, pp. 203–217, 2008.
- [9] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a web of open data," *Lecture Notes in Computer Science*, vol. 4825, p. 722, 2007.
- [10] S. Auer and J. Lehmann, "What have Innsbruck and Leipzig in common? Extracting semantics from wiki content," *Lecture Notes in Computer Science*, vol. 4519, p. 503, 2007.
- [11] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia—A crystallization point for the web of data," *Journal of Web Semantics [Forthcoming]*, 2009.
- [12] C. Zirn, V. Nastase, and M. Strube, "Distinguishing between instances and classes in the Wikipedia taxonomy," *Lecture Notes in Computer Science*, vol. 5021, p. 376, 2008.
- [13] V. Nastase and M. Strube, "Decoding Wikipedia categories for knowledge acquisition," in *Proceedings of the AAAI*, vol. 8, 2008.
- [14] S. Reed and D. Lenat, "Mapping ontologies into Cyc," in *Proc. AAAI Conference 2002 Workshop on Ontologies for the Semantic Web*, 2002.
- [15] C. Matuszek, M. Witbrock, R. Kahlert, J. Cabral, D. Schneider, P. Shah, and D. Lenat, "Searching for common sense: Populating Cyc from the Web," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1430.
- [16] M. Taylor, C. Matuszek, B. Klimt, and M. Witbrock, "Autonomous classification of knowledge into an ontology," in *The 20th International FLAIRS Conference (FLAIRS), Key West, Florida*, 2007.
- [17] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, p. 37, 1960.