Integrating YAGO into the Suggested Upper Merged Ontology

Gerard de Melo
Max Planck Institute
Saarbrücken, Germany
demelo@mpi-inf.mpg.de

Fabian Suchanek
Max Planck Institute
Saarbrücken, Germany
suchanek@mpi-inf.mpg.de

Adam Pease
Articulate Software
Angwin, CA, USA
apease@articulatesoftware.com

Abstract

Ontologies are becoming more and more popular as background knowledge for intelligent applications. Up to now, there has been a schism between manually assembled, highly axiomatic ontologies and large, automatically constructed knowledge bases. This paper discusses how the two worlds can be brought together by combining the high-level axiomatizations from the Standard Upper Merged Ontology (SUMO) with the extensive world knowledge of the YAGO ontology. The result is a new formal large-scale ontology, which provides information about millions of entities such as people, cities, organizations, and companies.

1. Introduction

Many modern information technology applications make use of ontological background knowledge, in fields as diverse as business information systems, bioinformatics, and information retrieval.

The Suggested Upper Model Ontology (SUMO) [8] is a large formal ontology with a wealth of axiomatized knowledge of general and domain-specific concepts, which makes it ideal for applications that need to draw conclusions with some kind of common sense. SUMO knows for example that every country has a capital or that humans communicate by talking. Now including a mid-level ontology and a variety of domain ontologies, it stands at around 20,000 terms and 70,000 axioms and is the largest open source formal upper ontology available. But the space of human knowledge is vast and SUMO has not emphasized capturing large numbers of simple facts. Thus, SUMO has only limited knowledge about cities, actors, or companies.

The YAGO ontology [11], on the other hand, is one of the largest resources of facts and entities available today. It combines the conceptual hierarchy of the WordNet lexical database [5] with the coverage of Wikipedia, the well-known Web-based encyclopedia. YAGO contains more than 1.7 million entities (politicians, countries, movies, etc.) and over 14 million facts about them. The latter include the Is-A hierarchy as well as non-taxonomic information. YAGO knows the birth dates of individuals, the locations of cities and the inflation rates of countries. YAGO is based on a clean logical model with a decidable consistency. However, YAGO itself only provides very rudimentary semantics based on merely five axioms, so only limited forms of reasoning are possible.

This paper investigates how the best of these two worlds can be brought together, revealing how millions of entities and facts from YAGO can rapidly be incorporated into SUMO by means of semi-automatic techniques.

2. Related Work

Numerous approaches have been proposed to construct general-purpose ontologies. One class of techniques focuses on extracting information automatically from text corpora [10, 4]. Despite good results, the quality remains below that of well-designed hand-crafted ontologies. Furthermore, the facts are not canonic, i.e. different identifiers are used for the same entity and no clearly defined relations exist.

The most successful ontologies are still assembled manually by human experts. These include domain-specific resources as well as general purpose ones such as Cyc [6]. Cyc's taxonomy is available freely, but the rules that define the terms in it are not. SUMO [8], by contrast, is a large general-purpose formal ontology that is freely available.

A number of projects have sought to construct fact repositories derived from Wikipedia. Most of these do not possess clear semantics. DBpedia [1], for instance, uses the words found in Wikipedia as relation names, so the same relationship can appear in multiple disguises (e.g. '*length-in-km*', '*length-km*'). Freebase [7] has defined a limited number of entity types and hence large amounts of entities lack class membership information. YAGO [11], in contrast, builds up a complete all-purpose knowledge base by drawing on Wikipedia as well as on the structural organization of WordNet [5]. Unlike the previously mentioned resources, it has a confirmed accuracy of more than 95%.

A large number of papers have studied the task of ontology mapping, which involves finding concepts or entities

that are shared by two ontologies. Our study considers the quite different task of merging two ontologies with very little overlap by discovering connections between them.

3. Integration of Entities

Both YAGO and SUMO aim at providing a conceptualization of what exists in the world in terms of entities or objects (construed in the broadest sense) and statements about them. YAGO is based on model-theoretic semantics, where entities are taken to include not only concrete individual objects but also classes and relations, for instance. SUO-KIF distinguishes individuals and classes, where the former is taken to include individual relations and functions.

3.1. Individuals

YAGO includes a plethora of entities such as organizations, products, places, events in history, and so forth, which can be integrated into SUMO. Three techniques are applied.

Semi-automatic matching: Although SUMO contains a comparably small amount of individuals, there is some overlap with YAGO. A weighted string similarity measure is applied to uncover such matches. We verified the matches manually and placed them in an equivalence table. This way, a portion of the YAGO identifiers is mapped explicitly to the corresponding SUMO identifiers. For example, YAGO's `Paris` is mapped to SUMO's `ParisFrance`.

Pruning: We attempt to avoid duplicate individuals from being included in the resulting ontology. Name similarity is a bad guide for duplicate entities, because similar names do not imply identical meaning and, likewise, two entities carrying differing names are not necessarily distinct. Hence, we generated an alternative abridged version of SUMO, where non-function, non-property, non-relational individuals are retained only if the corresponding YAGO entity is identified in the equivalence table mentioned above. In total, around 11,000 individuals (among them, over 9,000 airports) are removed. This is a relatively small portion of SUMO, whose main strength lies in the axiomatization of classes and predicates. Furthermore, the number of individuals omitted in the abridged SUMO version pales in comparison with the 1.7 million individuals from YAGO that emerge as new citizens of SUMO.

Name transformation: YAGO entities can then safely be added to SUMO. We construct a new, unique term name for each YAGO entity not listed in the equivalence table and add it to SUMO. This involves ensuring that the name has not already been used in SUMO, and that it abides to the rules of the SUO-KIF syntax specification.

3.2. Classes

When integrating YAGO's classes into SUMO, the goal is to transfer the YAGO taxonomy as precisely as possible while avoiding redundant duplicate classes and ensuring that newly imported classes are appropriately accommodated within SUMO's class hierarchy.

Merging Procedure to Remove Inconsistent Classes: In YAGO, we find for example that `BrownUniversity` is classified both as an instance of `College` and of `GroupOfPeople`, while in SUMO, an entity cannot be both a building and a group of people. At the top level, YAGO is partitioned into different branches, including artifacts, people, abstract entities, etc. If a YAGO individual is an instance in multiple branches, a voting procedure is used to determine the branch that most `type` facts lead to (breaking ties arbitrarily). These `type` statements are kept and all others are purged.

This decreases the number of `type` statements in YAGO by roughly 10% to four million. In return, each individual belongs to exactly one branch and potential errors in the

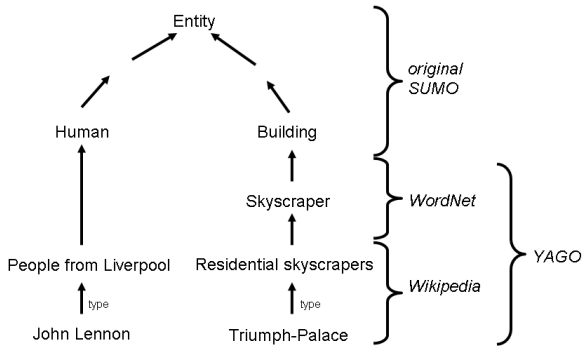

Figure 1: The Merged Taxonomy

Augmentation and Mapping Process: Most YAGO individuals are instances of classes derived from Wikipedia categories and have no corresponding term in SUMO (`John_Lennon`, e.g., is in the class `People_from_Liverpool`). We establish new SUMO terms for these classes and make the individuals instances of them. In YAGO, such classes are subclasses of classes derived from WordNet. For example, `People_from_Liverpool` is a subclass of the WordNet-derived class `person`. Using existing mappings from WordNet entries to SUMO [9], one can determine whether there exists an equivalent SUMO class. WordNet's `person`, for example, is mapped by an equivalence mapping to the SUMO class `Human`, so we can simply produce `(subclass PeopleFromLiverpool Human)`. In many cases, the WordNet mapping provides only a superclass, e.g. the `skyscraper` class is a subclass of SUMO's `Building` class. This impels us to add the WordNet class to SUMO and connect it to the existing superclass. Figure 1 exemplifies this process. In further cases, the WordNet mappings yield not a class, but a property or relation. For example, the WordNet class `Guitarist` is mapped to the property `Musician` in SUMO. In such cases, we add an axiom of the following form to SUMO:

```
(=> (instance ?ENTITY Guitarist)
    (property ?ENTITY Musician))
```

We then recursively move up YAGO's class hierarchy until an appropriate class or superclass is available in SUMO. This way, we can guarantee that each YAGO individual is integrated into SUMO's class hierarchy. Compared to YAGO alone, additional axioms thus become available for reasoning on them, e.g. SUMO explicitly formalizes that instances of `Human` can experience perceptions.

Quality Assessment: The knowledge in YAGO is subjected to a set of rigorous quality maintenance procedures. A human assessment study has shown that more than 95% of the statements are accurate [11]. This is guaranteed to carry over to the statements imported into SUMO, due to the use of hand-crafted transformation rules that will be described later on. A certain risk of decreased precision, however, cannot be ruled out at the nexus of YAGO and SUMO's class hierarchies.

For this reason, we conducted an additional human evaluation of this weakest part of our transformation. For a random sample of 300 new individuals, we moved up the class hierarchy until we found the most specific genuine SUMO class it is assigned to (e.g. `Building` for the `Triumph-Palace` instance). These assignments were then verified manually and the Wilson interval [2] at $\alpha = 5\%$ was used to generalize our findings on the sample to the whole ontology. We found that with a probability of 95%, the overall accuracy of links between entities and SUMO classes is in the range of $92.67\% \pm 2.98\%$. Given that we cannot surpass YAGO's 95%, this is a highly reassuring result that confirms the validity of our approach.

3.3. Semantics of Terms in Ontologies

An ontology usually has an intended denotation, i.e. an intended correspondence between its terms and real world objects. However, the fewer constraints the ontology imposes, the more denotations are possible. Moreover, unless one relies on externally defined *primitive* terms, it is not possible to exhaustively define all terms without interdependencies. This is much like a dictionary that defines Mandarin words using other Mandarin words, which is of little use to people lacking a basic understanding of at least some of the words. This indeterminacy is particularly pronounced for many OWL ontologies, where, replacing the often English-like names with more arbitrary identifiers, one often ends up solely with information of the form: `c87` is a subclass of `c34` and `c34` is a subclass of `c0`.

In a highly axiomatized ontology as SUMO, the problem is less severe since large numbers of axioms characterize the relationships between entities, so more unintended denotations can be ruled out. Even more can be ruled out if the denotation of certain terms is assumed to be fixed externally. For example, if the meaning of `representsInLanguage` and `EnglishLanguage` is taken to be properly defined, it becomes possible to ground the meaning of terms using statements such as `(representsInLanguage "Immanuel Kant" ImmanuelKant EnglishLanguage)`. Given that the interpretation of the constant `"Immanuel Kant"` is predetermined as simply being the respective symbolic string of characters, this tells us that the entity `ImmanuelKant` is one which is represented as '*Immanuel Kant*' in written English. The large number of YAGO entities described in this way then also aid in further fixing the meaning of the classes they are members of by characterizing them extensionally.

3.4. Literals

In YAGO, each literal is an instance of one of several hierarchically organized literal classes, e.g. the number `5` is an instance of the `PositiveInteger`. SUMO assumes a universe of discourse containing real numbers and finite symbolic character strings, so YAGO's number and string literals trivially correspond to the respective SUMO entities.

YAGO also knows dimensioned literals, which combine a number and a unit of measurement (e.g. `3.0#m^2`). SUMO defines the function `MeasureFn`, which takes a constant number and a unit and yields an instance of `ConstantQuantity`. For example, YAGO's `3.0#m^2` becomes `(MeasureFn 3.0 SquareMeter)`. In YAGO, each quantity exists exactly once and is represented uniformly using a predetermined unit, usually an SI unit, whereas SUMO models dependencies between different units using general axioms.

For time intervals, YAGO uses simple literals, while SUMO contains functions that yield classes representing the intervals. Thus, YAGO's `1961-11-28` is rewritten as `(DayFn 28 (MonthFn 11 (YearFn 1961)))`, and `147#` (the 1470's) is recast as `(DayFn ?DAYNO (MonthFn ?MONTHNO (YearFn ?YEARNO)))` where `?DAY`, `?MONTH`, and `?YEAR` are existentially quantified variables and `?YEAR` is constrained as follows:

```
(greaterThanOrEqualTo ?YEARNO 1470)
(lessThanOrEqualTo ?YEARNO 1479)
```

4. Integrating Factual Knowledge

Apart from the taxonomical relations mentioned earlier, YAGO also extracts a substantial amount of world knowledge from the infoboxes on Wikipedia pages. This includes for instance biographical information such as the birth date of a person and economic facts about a country. Around 100 different types of relations are currently used to capture such facts. The intended semantics of these relations vary quite considerably and are not specified formally in YAGO, so explicit conversion rules need to be established for each relation when integrating this knowledge into SUMO.

4.1. Transformation Rules

In certain cases, a direct correspondence between YAGO relations and SUMO ones can be found, so the statements are amenable to trivial mappings. For instance, for

YAGO's `hasCapital`, the inverse relation `capitalCity` has been defined in SUMO. In other cases, new relations need to be introduced to SUMO to reflect the intended semantics of the relation in YAGO. These have to be constrained appropriately by axioms. For instance, YAGO's `establishedOnDate` can be defined as follows:

```
(instance establishedOnDate BinaryRelation)
(domain 1 establishedOnDate Agent)
(domain 2 establishedOnDate TimeInterval)
(=> (establishedOnDate ?OBJ ?TIME)
    (exists (?FOUNDING) (and
        (instance ?FOUNDING Founding)
        (result ?FOUNDING ?OBJ)
        (overlapsTemporally
          (WhenFn ?FOUNDING) TIME))))
```

In order to make the knowledge from YAGO more useful in practical applications, we added further new axioms to SUMO to enable additional common sense reasoning. For instance, that people cannot act before being born:

```
(=> (and (birthdate ?HUMAN ?DAY)
         (agent ?PROCESS ?HUMAN))
    (beforeOrEqual
      (BeginFn ?DAY)
      (BeginFn (WhenFn ?PROCESS))))
```

For more details on this transformation, see [3].

4.2. Reification

YAGO relies heavily on *reification*, where statements are treated as entities and hence higher-order statements, i.e. statements about statements, can be expressed. To a large extent, reification in YAGO is used to convey information about the knowledge extraction process such as sources and techniques used to garner knowledge. Such data is not of interest in the transformation to SUO-KIF.

Reification in YAGO is also used to express relations with an arity higher than two. For instance, Plato is called '*Platone*' in Italian. In YAGO, this ternary statement is decomposed as a reified statement and one or more higher-order statements: `(Plato isCalled "Platone") inLanguage Italian_language`. Where such n-ary relations exist in SUMO, we can take advantage of them as `(representsInLanguage "Platone" Plato ItalianLanguage)`. The only case where this is not possible is for YAGO's time qualifications using `since`, `until`, `during`, which are rewritten e.g. as

```
(exists (?INTERVAL) (and
    (beforeOrEqual(BeginFn (YearFn 1867))
                  (BeginFn ?INTERVAL))
    (beforeOrEqual(EndFn ?INTERVAL)
                  (EndFn (YearFn 1918)))
    (holdsDuring ?INTERVAL
       (instance AustriaHungary Nation))))
```

5. Conclusions

The complementary nature of SUMO and YAGO has led us to establish a means of reconciling the different conceptualizations, thereby giving rise to a fruitful symbiosis that combines the axiomatic formalization manifested in SUMO with the massive body of knowledge accumulated in YAGO. The unification rests on semi-automatic techniques that recast the content of YAGO in the formal framework of SUMO, yielding an ontology of nearly two million entities and several million facts and axioms about them, thereby increasing the number of entities in SUMO by multiple orders of magnitude. Future work includes continuing to expand the number of axioms in SUMO to make more forms of inferences possible on the entities.

With the combined force of the two ontologies, an enormous, unprecedented corpus of formalized world knowledge is available for automated processing and reasoning. We anticipate that this will foster a wide range of new, intelligent applications in numerous domains.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A nucleus for a web of open data. In *Proc. ISWC*, volume 4825 of *LNCS*. Springer, 2007.

[2] L. D. Brown, T. T. Cai, and A. DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–133, 2001.

[3] G. de Melo, F. M. Suchanek, and A. Pease. Integrating YAGO into the Suggested Upper Merged Ontology. Research Report MPI-I-2008-5-003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, 2008.

[4] O. Etzioni, M. J. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in KnowItAll. In *Proc. WWW*, 2004.

[5] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

[6] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira. An introduction to the syntax and content of Cyc. In *Proc. AAAI Spring Symposium*, 2006.

[7] Metaweb Technologies. The Freebase project. *http://www.freebase.com/*.

[8] I. Niles and A. Pease. Toward a Standard Upper Ontology. In C. Welty and B. Smith, editors, *Proc. 2nd Intl. Conf. on Formal Ontology in Information Systems (FOIS)*, 2001.

[9] I. Niles and A. Pease. Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proc. IEEE IKE*, pages 412–416, 2003.

[10] F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from Web documents. In *Proc. KDD*, 2006.

[11] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *Proc. WWW*, New York, NY, USA, 2007. ACM Press.