Integrating YAGO into the Suggested Upper Merged Ontology

Gerard de Melo
Fabian M. Suchanek
Adam Pease

Authors’ Addresses

Gerard de Melo
Max-Planck Institute for Informatics
Campus E1.4
66123 Saarbrücken
Germany

Fabian M. Suchanek
Max-Planck Institute for Informatics
Campus E1.4
66123 Saarbrücken
Germany

Adam Pease
Articulate Software
Angwin, CA
USA
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 report 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. On the theoretical side, it analyses the differences between the knowledge representation in YAGO and SUMO. On the practical side, this report explains how the two resources can be merged. This yields a new large-scale formal ontology, which provides information about millions of entities such as people, cities, organizations, and companies. This report is the detailed version of our paper [15].

Keywords

YAGO, SUMO, Ontology, Integration, Merging
Contents

1 Introduction 2

2 Related Work 4

3 Knowledge Sources 6
   3.1 The Suggested Upper Merged Ontology ................. 6
   3.2 Wikipedia ............................................ 6
   3.3 WordNet .............................................. 7
   3.4 YAGO .................................................. 7
   3.5 Mappings from WordNet to SUMO ....................... 8

4 Integration of Entities 9
   4.1 Individuals ............................................ 9
   4.2 Classes ............................................... 10
   4.3 Semantics of Terms in Ontologies ...................... 12
   4.4 Literals .............................................. 13

5 Integrating Factual Knowledge 16
   5.1 Simple Transformation Rules ............................ 16
   5.2 Advanced Transformation Rules ........................ 17
   5.3 Reification ........................................... 18

6 Conclusions 19
1 Introduction

Many modern information technology applications make use of ontological background knowledge, in fields as diverse as business information systems, bioinformatics, information retrieval, and Semantic Web applications. Machine translation [10], word sense disambiguation [7], and query expansion [25, 20, 45] exploit lexical knowledge, information retrieval and document classification benefit from taxonomical information [23], and question answering relies strongly on background knowledge [22, 24]. Furthermore, ontological knowledge structures play an important role in data cleaning [11], record linkage (entity resolution) [13], and information integration in general [35]. In addition, there are emerging trends towards entity- and fact-oriented Web search and community management [5, 8, 9, 12, 16, 24, 26, 31, 32], which can build on rich knowledge bases.

The Suggested Upper Model Ontology (SUMO)[33] is a large formal ontology with detailed axiomatization of general and domain-specific concepts. Its wealth of axiomatized world knowledge 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. 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 the cities, actors, or companies of this world.

The YAGO ontology[43], on the other hand, is one of the largest resources of facts and entities available today. It contains more than 1.7 million entities (such as politicians, countries, movies, etc.) and millions of facts about them. YAGO knows the birth dates of individuals, the locations of cities, and the inflation rates of countries. However, YAGO provides only a rudimentary axiomatization. Thus, only limited forms of deduction are possible on YAGO.
This report 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 [15]. For this purpose, we also examine the different conceptualizations in YAGO and SUMO and indicate how they can be reconciled.
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 [2, 14, 42, 17, 36]. Despite their reasonable results, the quality remains significantly below that of well-designed hand-crafted knowledge bases. Furthermore, the facts are not canonic, i.e. different identifiers are used for the same entity. This is because a common frame of reference is missing. For similar reasons, no clearly defined relations exist.

Due to these limitations, 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 [28] and SUMO [33]. Cyc is a commercially developed ontology. Its taxonomy is available freely, but the rules that define the terms in it are not. SUMO, by contrast, is a large general-purpose formal ontology that is freely available. SUMO has been reviewed by a community of experts and has been subjected to formal verification with automated theorem provers, thus fulfilling the highest quality standards. Yet, continuous human effort is needed to keep it up to date with new entities.

This problem of constant maintenance calls for intelligent automatic approaches to discover vast amounts of entities and facts. A number of projects have sought to construct knowledge repositories by deriving explicit facts from Wikipedia. Most of these knowledge bases do not possess clear semantics, let alone an axiomatization. DBpedia [4], for instance, uses the words found in Wikipedia as relation names, so the same relationship can appear in multiple disguises (e.g. ‘length’, ‘length-in-km’, ‘length-km’). The Freebase project [29] aims to produce a shared database of the world’s knowledge by extracting information from existing sources and inviting volunteers to contribute. However, Freebase has defined only a limited number of entity types so far and hence large amounts of entities lack class membership information. Ponzetto et al. [41] use heuristics to derive a taxonomy from Wikipedia categories, but do not aim at a full general-purpose knowledge base. Isolde [46] extracts class candidates from a specific domain corpus, using background
knowledge from Wikipedia and Wiktionary. YAGO [43], in contrast, builds up a complete all-purpose knowledge base by drawing on Wikipedia as well as on the structural organization of WordNet [18]. Unlike the previously mentioned knowledge bases, it has a confirmed accuracy of more than 95%, making it the perfect choice for extending SUMO.

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. This involves reconciling two different knowledge representations.
3 Knowledge Sources

In this section, we present and compare the ontologies and related knowledge sources used for our knowledge integration process.

3.1 The Suggested Upper Merged Ontology

The Suggested Upper Merged Ontology began in the year 2001 as a formal upper ontology of roughly 1,000 terms and 4,000 axioms. Now including a mid-level ontology (MILO) 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.

The axioms are expressed in SUO-KIF [38], a variant of the Knowledge Interchange Format [19]. SUMO also has an associated reasoning and development system called Sigma [37], which has recently been extended with many theorem provers from the CADE ATP System Competition [40, 44].

<table>
<thead>
<tr>
<th></th>
<th>SUMO (core)</th>
<th>MILO</th>
<th>Domain ontologies</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>1,120</td>
<td>2,167</td>
<td>18,171</td>
<td>21,458</td>
</tr>
<tr>
<td>Axioms</td>
<td>4,522</td>
<td>4,394</td>
<td>64,069</td>
<td>72,985</td>
</tr>
<tr>
<td>Rules</td>
<td>794</td>
<td>528</td>
<td>1,679</td>
<td>3,001</td>
</tr>
</tbody>
</table>

Over the years, SUMO has been augmented with several domain ontologies for areas as diverse as economy, engineering, geography, and transportation. However, much of this knowledge is still relatively high level, defining each domain of discourse without concentrating on facts that would appear in a database. There is still a need for detailed encyclopedic knowledge about particular entities, such as for example information about politicians, animal species, or significant events in history.

3.2 Wikipedia

Wikipedia [1] is a multilingual, Web-based encyclopedia that is written collaboratively by volunteers and is available for free. Currently, the English
version of Wikipedia comprises more than two million articles. Each is presented on a separate Web page and usually describes a single entity or topic.

The majority of Wikipedia pages have been manually assigned to one or multiple categories. The page about Elvis Presley, for example, is in the categories ‘American rock singers’, ‘1935 births’, and over 30 others. A Wikipedia page may include a so-called infobox. Infoboxes are standardized tables with information about the entity described in an article. For example, the standardized infobox for people provides the birth date, profession, and nationality of a person. Similar ones exist for cities, musical artists, companies, etc. However, Wikipedia’s category system and infoboxes are based on subject areas rather than on ontological criteria, so this information cannot be used directly as an ontology.

3.3 WordNet

WordNet [18] is a lexicon for the English language that captures information about the senses of words. A set of words that express the same sense is called a synset. Terms and synsets are organized as a network of nodes linked by various lexico-semantic relations.

The hyponymy relation can be defined as one that “holds between a more specific, or subordinate, lexeme and a more general, or superordinate, lexeme, as exemplified by such pairs as ‘cow’:‘animal’, ‘rose’:‘flower’” [27]. Hypernymy is the respective inverse relation, which, in WordNet, spans a directed acyclic graph between synsets, with a single root node. If a term’s intension is taken as determining its extension, hyponymy would entail subsumption relationships between the respective classes of entities being referenced. Similarly, WordNet’s meronymic relations between synsets correspond closely to the respective mereological part/whole relations between their real-world referents.

WordNet is based on linguistic criteria. For instance, a term is considered a hyponym of another one if and only if “native speakers accept sentences constructed from such frames as ‘An x is a (kind of) y’” [30]. Unfortunately, formal analysis has unveiled examples of where this leads to hyponymic relationships that do not strictly imply subsumption [21].

3.4 YAGO

YAGO [43] is an ontology that combines the coverage of Wikipedia with the conceptual hierarchy of WordNet [18]. YAGO builds on entities and relations and currently describes more than 1.7 million entities and 14 million facts. The latter include the Is-A hierarchy as well as non-taxonomic relations between entities (such as hasWonPrize). There are currently 100 different
binary relationships in YAGO. The entities and facts about them are mainly extracted from Wikipedia’s category system and infoboxes, whereas the class hierarchy is derived from WordNet.

YAGO is based on a clean logical model with a decidable consistency. However, YAGO itself only provides very rudimentary semantics based on merely five basic axioms, so only limited forms of reasoning are possible. Furthermore, its upper level relies entirely on WordNet, which, as elaborated earlier, has certain limitations when conceived as a formal ontology.

3.5 Mappings from WordNet to SUMO
SUMO has been linked by hand to all of the WordNet lexicon [34, 39]. Three kinds of relationships between the SUMO concept and the WordNet synset are distinguished:

1. Synonymy: The WordNet synset is equivalent in meaning to the SUMO concept. For example, the WordNet synset \{'artificial satellite', 'orbiter', 'satellite'\} corresponds exactly to SUMO’s ArtificialSatellite.
2. Subsumption: The lexical concept corresponding to the synset is subsumed by the SUMO concept. For example, \{'elk'\} maps to the more general SUMO term HoofedMammal. WordNet is considerably larger than SUMO and so many synsets are mapped to the same more general formal term.
3. Instance relationships: The object referred to by the WordNet synset is an instance of the SUMO concept. For example, the synset \{'George Washington', 'President Washington', 'Washington'\} ”first President of the United States” refers to an instance of the SUMO’s conception of Human.
4 Integration of Entities

The available resources suggest that the axiomatic knowledge provided by SUMO can be extended semi-automatically with the large number of entities and facts in YAGO. To achieve this, certain conceptual differences in their modelling of the world need to be overcome. The integration process will be described in a bottom-up manner, beginning with entities and then proceeding at the level of statements in Section 5.

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. Hence, the majority of YAGO’s entities can be integrated into SUMO.

4.1 Individuals

YAGO includes a plethora of entities such as people, organizations, products, geographical entities, cultural artifacts, events in history, and so forth, covering virtually all areas of human inquiry.

We use three different techniques to integrate the YAGO entities into SUMO:

**Semi-automatic matching:** Although SUMO is only aware of a comparably small amount of individuals, some degree of overlap between YAGO and SUMO exists, e.g. SUMO includes a few hundred countries and cities. A weighted string similarity measure is applied to uncover such matches. To guarantee highest accuracy, we verified these match candidates manually. Correct matches are placed 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:** While SUO-KIF does not postulate any unique names assump-
tion, it may make sense to preclude duplicate instances from being part of the ontology. There is undoubtedly no fail-safe method for detecting such duplicates automatically. Similar names do not imply identical meaning, e.g. YAGO’s GreekLanguage refers to the Greek language in all its variants, whereas SUMO’s GreekLanguage refers to Modern Greek only. Likewise, two entities carrying differing names are not necessarily distinct (e.g. Tonne and MetricTon). To avoid duplicate entities in spite of these difficulties, we generate alternative abridged versions of SUMO’s domain ontologies: 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), and, by extension, around 33,000 statements involving them 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: The remaining YAGO individuals can then safely be transferred 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.

4.2 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: Many terms of human languages can quite regularly be used in a number of metonymically related senses [3], e.g. the term ‘university’ can be used to refer to the institution, the faculty and students, or the campus. As this polysemy is also reflected in Wikipedia and hence YAGO, we find that BrownUniversity is classified both as an instance of College and of GroupOfPeople. In SUMO, however, an entity cannot be both a building and a group of people. In some cases, the double classification in YAGO is erroneous. For example, Abraham Lincoln is an instance of twelve subclasses of the class person, such as lawyer and president. However, he is also falsely listed as an instance of the class cabinet.

At the top level, YAGO is partitioned into different branches, including
locations, artifacts, people, other physical entities, and abstract entities. Our merging algorithm identifies these branches. 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 YAGO taxonomy are removed.

**Augmentation and Mapping Process:** Most YAGO individuals are instances of classes that have been derived from Wikipedia categories and have no corresponding term in SUMO, e.g. John Lennon is in the class People from Liverpool. We establish new SUMO terms for these classes and the individuals are made instances of the newly created classes. In YAGO, such classes are subclasses of classes that have been derived from WordNet synsets. For example, People from Liverpool is a subclass of the WordNet-derived class person. Using the SUMO-WordNet-mappings, 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

$$\text{(subclass PeopleFromLiverpool Human)}$$

In many cases, the WordNet mapping yields only a superclass. For example, the skyscraper class is not equivalent to the SUMO class Building, but is a subclass of it. This impels us to add the WordNet class as a new class to SUMO and connect it to the existing superclass. Figure 1 exemplifies this process.

In further cases, the mapping yields 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:

$$\Rightarrow$$

$$(\text{instance ?ENTITY Guitarist})$$

$$(\text{property ?ENTITY Musician})$$

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 connected to at least one class in 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 [43]. 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: YAGO uses heuristics to assign individuals to Wikipedia classes and to link the Wikipedia classes to WordNet classes, which are finally connected to SUMO classes using the WordNet-SUMO-mapping.

For this reason, we conducted an additional human evaluation of this weakest part of our transformation. We determined for each individual its most specific SUMO class, that is, for a given individual, we move up the class hierarchy until a genuine SUMO class is encountered, for example Building for the Triumph-Palace instance. A random sample of 300 such pairs was then verified manually. We computed the Wilson interval [6] at $\alpha = 5\%$ to generalize our findings on the sample to the whole ontology. We find 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.

4.3 Semantics of Terms in Ontologies
An ontology usually has an intended denotation, i.e. an intended correspondence between its terms and real world objects. For example, in YAGO, the intended denotation of the term George Washington is the first US president. 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 interdepen-
dencies. This is much like a monolingual dictionary that defines Mandarin words using other Mandarin words. Such a dictionary is of little use to someone who does not already possess an understanding of at least some of the words. This indeterminacy is particularly pronounced for many of the concept names in public OWL ontologies, where concepts are frequently only characterized as being subsumed by some other concept, which in turn is similarly underspecified. Replacing the often English-like names with more arbitrary identifiers, one ends up with information of the form: c87 is a subclass of c34 and c34 is a subclass of c0, so the formalization alone does not reflect the intended semantics very well.

In a highly axiomatized ontology as SUMO, this problem is less pronounced because large numbers of axioms characterize the relationships between entities, so more unintended denotations can be ruled out. The number of denotations can be further pruned 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 individual terms using statements of the following form:

\[(\text{representsInLanguage} \ "\text{Immanuel Kant}\" \ \text{ImmanuelKant} \ \text{EnglishLanguage})\]

Given that the interpretation of the string constant "Immanuel Kant" is predetermined as simply being the respective symbolic string of characters, this information from YAGO allows us to characterize the entity ImmanuelKant as one which is represented as the string of characters 'Immanuel Kant' in written English. When names are ambiguous, providing such symbolic strings for multiple languages can further reduce the range of possible interpretations. The large number of new entities described in this way then also aid in further fixing the meaning of the classes they are members of by characterizing them extensionally.

### 4.4 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 class PositiveInteger, which is a subclass of Integer. SUMO assumes a universe of discourse containing real numbers and finite symbolic strings of characters, so YAGO’s number and string literals trivially correspond to the respective entities in SUMO.
YAGO also knows dimensioned literals, which combine a number and a unit of measure (e.g. 3.0#m^2). These include physical dimensions like length and time, but also others such as monetary currencies. In SUMO, dimensioned quantities are instances of the class `PhysicalQuantity`, which contains measures of quantifiable aspects of the world that need not be material ones. A `ConstantQuantity` is a `PhysicalQuantity` that does not change its value. To express dimensioned quantities, SUMO defines the function `MeasureFn`, which takes a constant number and a unit of measurement and yields an instance of `ConstantQuantity`. For example, YAGO’s 3.0#m^2 becomes `(MeasureFn 3.0 SquareMeter)`.

Given that different units of measurements exist, the same dimensioned quantity could be represented in multiple ways. In YAGO, the quantity exists exactly once and is represented uniformly as a literal with a predetermined unit, usually an SI unit. Using reification (see Section 5.3), it is possible to describe the relation between a quantity and its value in different units:

```
(1000#g hasValue 1000) inUnit gram
(1000#g hasValue 1) inUnit kilogram
```

SUMO models such identities using axioms that capture general dependencies between units, for example:

```
(=)
   (instance ?NUMBER RealNumber)
   (equal
      (MeasureFn ?NUMBER Kilogram)
      (MeasureFn
         (MultiplicationFn ?NUMBER 1000)
         Gram))
```

As SUMO’s representation is more expressive, a smooth integration of YAGO’s literals into SUMO is possible. A similar observation holds for time intervals: YAGO uses simple literals, while SUMO uses functions that yield classes representing the intervals. Thus, YAGO’s 1961-11-28 is rewritten as

```
(DayFn 28 (MonthFn 11 (YearFn 1961)))
```

YAGO also knows literals with wild cards that represent longer time intervals, such as 147# for ‘the 1470’s’. Depending on the context, this wild card literal may be 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)

SUMO’s modelling thus explicitly formalizes the relationships between different entities, whereas YAGO tends to capture such information much more succinctly, however without formal characterization.
5 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.

5.1 Simple 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 example YAGO’s bornIn relation corresponds directly to SUMO’s birthplace. Similarly, for YAGO’s hasCapital the inverse relation capitalCity has been defined in SUMO, so an additional adaptation of the argument order suffices to rephrase the statements. In cases such as the projection of bornOnDate to birthdate, the date specifications additionally need to be acknowledged as explained earlier. For example, YAGO’s

HerveyDeStanton bornOnDate 127##-###-##

is rendered as

(exists ?YEARNO ?MONTHNO ?YEARNO
   (and
     (birthdate HerveyDeStanton
      (DayFn ?DAYNO
       (MonthFn ?MONTHNO
        (YearFn ?YEARNO)))
     (greaterThanOrEqualTo ?YEARNO 1270))
   )
)
5.2 Advanced Transformation Rules

In some cases, there was no straightforward correspondence between relations in the two resources, because the respective domains had not been sufficiently addressed by SUMO. For example, YAGO contains a wealth of information on entertainment-related entities such as the genre and director of a movie, or the creator of a music album cover.

In such circumstances, 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:

\[
\begin{align*}
\text{(instance establishedOnDate BinaryRelation)} \\
\text{(domain 1 establishedOnDate Agent)} \\
\text{(domain 2 establishedOnDate TimeInterval)} \\
\Rightarrow \\
\text{(establishedOnDate ?OBJ ?TIME)} \\
\text{(exists (?FOUNDING)} \\
\text{(and)} \\
\text{(instance ?FOUNDING Founding)} \\
\text{(result ?FOUNDING ?OBJ)} \\
\text{(overlapsTemporally (WhenFn ?FOUNDING) TIME))}
\end{align*}
\]

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

\[
\begin{align*}
\Rightarrow \\
\text{(and)} \\
\text{(birthdate ?HUMAN ?DAY)} \\
\text{(agent ?PROCESS ?HUMAN)} \\
\text{(beforeOrEqual (BeginFn ?DAY) (BeginFn (WhenFn ?PROCESS)))}
\end{align*}
\]

Fortunately, re-merging YAGO and SUMO in the future merely requires adding a few additional rules in case YAGO has added new relations or updating existing ones in the rare case that SUMO terms have been renamed or redefined. Apart from this, the new merges can be generated automatically.
5.3 Reification

Another important aspect of YAGO is that it relies heavily on reification. Reification treats statements as entities and therefore allows making higher-order statements, i.e. statements about statements. To a large extent, reification in YAGO is used to convey information about the knowledge extraction process, keeping track of the Wikipedia pages and techniques used to garner information. Such data is not of interest for the SUMO integration process.

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:

\[(\text{Plato}\ \text{isCalled}\ "\text{Platone}"),\]
\[
\text{inLanguage}\ \text{Italian\_language}
\]

Here, the relation \text{inLanguage} takes as its first argument the fact that Plato is called ‘Platone’. This way, YAGO can express many \(n\)-ary relations with binary predicates. Where such \(n\)-ary relations exist in SUMO, we can take advantage of them as follows:

\[(\text{representsInLanguage}"
\text{Platone}"
\text{Plato}
\text{Italian\_Language})\]

The only case where this is not possible is for time qualifications in YAGO such as

\[(\text{Austria\_Hungary}\ \text{type}\ \text{country})
\text{since}\ 1867-##-##\]
\[(\text{Austria\_Hungary}\ \text{type}\ \text{country})
\text{until}\ 1918-##-##\]

These statements (and similar ones with \text{during}) are rewritten as

\[(\text{exists}\ (?\text{INTERVAL})\]
\[(\text{and})\]
\[(\text{beforeOrEqual}(\text{BeginFn}\ (\text{YearFn}\ 1867))\]
\[(\text{BeginFn}\ ?\text{INTERVAL}))\]
\[(\text{beforeOrEqual}(\text{EndFn}\ ?\text{INTERVAL})\]
\[(\text{EndFn}\ (\text{YearFn}\ 1918)))\]
\[(\text{holdsDuring}\ ?\text{INTERVAL}\]
\[(\text{instance}\ \text{Austria\_Hungary}\ \text{Nation}))))\]
6 Conclusions

Our study has disseminated the different approaches to modelling world knowledge embraced by YAGO and SUMO. While YAGO focuses on broad coverage of entities with support for efficient querying, SUMO’s main strengths are the clean, expressive formal model and the axiomatic representation of common sense knowledge. The complementary nature of the two 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. This amounts to catapulting SUMO from the level of an upper ontology focusing on general concepts to the level of a full-fledged all-purpose knowledge base. During the course of this study, we further identified and resolved several issues in YAGO, SUMO, and in the WordNet-SUMO mappings, e.g. errors resulting from inaccurate word sense disambiguation in YAGO and inappropriate mappings. 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.
Bibliography


Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from ftp.mpi-sb.mpg.de under the directory pub/papers/reports. Most of the reports are also accessible via WWW using the URL http://www.mpi-sb.mpg.de. If you have any questions concerning ftp or WWW access, please contact reports@mpi-sb.mpg.de. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: library@mpi-sb.mpg.de

<table>
<thead>
<tr>
<th>Report Number</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI-I-2008-5-003</td>
<td>G. de Melo, F. Suchanek, A. Pease</td>
<td>Integrating YAGO into the Suggested Upper Merged Ontology</td>
</tr>
<tr>
<td>MPI-I-2008-5-001</td>
<td>F. Suchanek, G. Kasneci, M. Ramanath, M. Sozio, G. Weikum</td>
<td>STAR: Steiner Tree Approximation in Relationship-Graphs</td>
</tr>
<tr>
<td>MPI-I-2008-1-001</td>
<td>D. Ajwani</td>
<td>Characterizing the performance of Flash memory storage devices and its impact on algorithm design</td>
</tr>
<tr>
<td>MPI-I-2007-RG1-002</td>
<td>T. Hillenbrand, C. Weißenbach</td>
<td>Superposition for Finite Domains</td>
</tr>
<tr>
<td>MPI-I-2007-5-003</td>
<td>F.M. Suchanek, G. Kasneci, G. Weikum</td>
<td>Yago: A Large Ontology from Wikipedia and WordNet</td>
</tr>
<tr>
<td>MPI-I-2007-4-008</td>
<td>J. Gall, T. Brox, B. Rosenhahn, H. Seidel</td>
<td>Global Stochastic Optimization for Robust and Accurate Human Motion Capture</td>
</tr>
<tr>
<td>MPI-I-2007-4-006</td>
<td>C. Dyken, G. Ziegler, C. Theobalt, H. Seidel</td>
<td>GPU Marching Cubes on Shader Model 3.0 and 4.0</td>
</tr>
<tr>
<td>MPI-I-2007-4-004</td>
<td>C. Stoll</td>
<td>A Volumetric Approach to Interactive Shape Editing</td>
</tr>
<tr>
<td>MPI-I-2007-4-003</td>
<td>R. Bargmann, V. Blanz, H. Seidel</td>
<td>A Nonlinear Viseme Model for Triphone-Based Speech Synthesis</td>
</tr>
<tr>
<td>MPI-I-2007-4-002</td>
<td>T. Langer, H. Seidel</td>
<td>Construction of Smooth Maps with Mean Value Coordinates</td>
</tr>
<tr>
<td>MPI-I-2007-4-001</td>
<td>J. Gall, B. Rosenhahn, H. Seidel</td>
<td>Clustered Stochastic Optimization for Object Recognition and Pose Estimation</td>
</tr>
<tr>
<td>MPI-I-2007-2-001</td>
<td>A. Podelski, S. Wagner</td>
<td>A Method and a Tool for Automatic Verification of Region Stability for Hybrid Systems</td>
</tr>
<tr>
<td>MPI-I-2007-1-002</td>
<td>E. Althaus, S. Canzar</td>
<td>A Lagrangian relaxation approach for the multiple sequence alignment problem</td>
</tr>
<tr>
<td>MPI-I-2007-1-001</td>
<td>E. Berberich, L. Kettner</td>
<td>Linear-Time Reordering in a Sweep-line Algorithm for Algebraic Curves Intersecting in a Common Point</td>
</tr>
<tr>
<td>MPI-I-2006-5-005</td>
<td>R. Angelova, S. Siersdorfer</td>
<td>A Neighborhood-Based Approach for Clustering of Linked Document Collections</td>
</tr>
<tr>
<td>MPI-I-2006-5-004</td>
<td>F. Suchanek, G. Ifrim, G. Weikum</td>
<td>Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents</td>
</tr>
<tr>
<td>Document ID</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>MPI-I-2006-5-003</td>
<td>V. Scholz, M. Magnor</td>
<td>Garment Texture Editing in Monocular Video Sequences based on Color-Coded Printing Patterns</td>
</tr>
<tr>
<td>MPI-I-2006-5-002</td>
<td>H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum</td>
<td>IO-Top-k: Index-access Optimized Top-k Query Processing</td>
</tr>
<tr>
<td>MPI-I-2006-5-001</td>
<td>M. Bender, S. Michel, G. Weikum, P. Triantafillou</td>
<td>Overlap-Aware Global df Estimation in Distributed Information Retrieval Systems</td>
</tr>
<tr>
<td>MPI-I-2006-4-010</td>
<td>A. Belyaev, T. Langer, H. Seidel</td>
<td>Mean Value Coordinates for Arbitrary Spherical Polygons and Polyhedra in $R^3$</td>
</tr>
<tr>
<td>MPI-I-2006-4-009</td>
<td>J. Gall, J. Potthoff, B. Rosenthal, C. Schnorr, H. Seidel</td>
<td>Interacting and Annealing Particle Filters: Mathematics and a Recipe for Applications</td>
</tr>
<tr>
<td>MPI-I-2006-4-008</td>
<td>I. Albrecht, M. Kipp, M. Neff, H. Seidel</td>
<td>Gesture Modeling and Animation by Imitation</td>
</tr>
<tr>
<td>MPI-I-2006-4-007</td>
<td>O. Schall, A. Belyaev, H. Seidel</td>
<td>Feature-preserving Non-local Denoising of Static and Time-varying Range Data</td>
</tr>
<tr>
<td>MPI-I-2006-4-005</td>
<td>A. Belyaev, H. Seidel, S. Yoshizawa</td>
<td>Skeleton-driven Laplacian Mesh Deformations</td>
</tr>
<tr>
<td>MPI-I-2006-4-004</td>
<td>V. Havran, V. Herzog, H. Seidel</td>
<td>On Fast Construction of Spatial Hierarchies for Ray Tracing</td>
</tr>
<tr>
<td>MPI-I-2006-4-003</td>
<td>E. de Aguiar, R. Zayer, C. Theobalt, M. Magnor, H. Seidel</td>
<td>A Framework for Natural Animation of Digitized Models</td>
</tr>
<tr>
<td>MPI-I-2006-4-002</td>
<td>G. Ziegler, A. Tevs, C. Theobalt, H. Seidel</td>
<td>GPU Point List Generation through Histogram Pyramids</td>
</tr>
<tr>
<td>MPI-I-2006-2-001</td>
<td>J. Hoffmann, C. Gomes, B. Selman</td>
<td>Bottleneck Behavior in CNF Formulas</td>
</tr>
<tr>
<td>MPI-I-2005-4-006</td>
<td>C. Fuchs, M. Goesele, T. Chen, H. Seidel</td>
<td>An Empirical Model for Heterogeneous Translucent Objects</td>
</tr>
<tr>
<td>MPI-I-2005-4-005</td>
<td>G. Krawczyk, M. Goesele, H. Seidel</td>
<td>Photometric Calibration of High Dynamic Range Cameras</td>
</tr>
<tr>
<td>MPI-I-2005-4-002</td>
<td>O. Schall, A. Belyaev, H. Seidel</td>
<td>Sparse Meshing of Uncertain and Noisy Surface Scattered Data</td>
</tr>
<tr>
<td>MPI-I-2005-4-001</td>
<td>M. Fuchs, V. Blanz, H. Lensch, H. Seidel</td>
<td>Reflectance from Images: A Model-Based Approach for Human Faces</td>
</tr>
<tr>
<td>MPI-I-2005-2-004</td>
<td>Y. Kazakov</td>
<td>A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures</td>
</tr>
<tr>
<td>MPI-I-2005-2-003</td>
<td>H.d. Nivelle</td>
<td>Using Resolution as a Decision Procedure</td>
</tr>
<tr>
<td>MPI-I-2005-2-002</td>
<td>P. Maier, W. Charatonik, L. Georgiev</td>
<td>Bounded Model Checking of Pointer Programs</td>
</tr>
<tr>
<td>MPI-I-2005-2-001</td>
<td>J. Hoffmann, C. Gomes, B. Selman</td>
<td>Bottleneck Behavior in CNF Formulas</td>
</tr>
<tr>
<td>MPI-I-2005-1-007</td>
<td>I. Katriel, M. Kutz</td>
<td>A Faster Algorithm for Computing a Longest Common Increasing Subsequence</td>
</tr>
</tbody>
</table>
MPI-I-2005-1-003  S. Baswana, K. Telikepalli  Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs

MPI-I-2005-1-002  I. Katriel, M. Kutz, M. Skutella  Reachability Substitutes for Planar Digraphs