YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia

Fabian MS, Gjergji K, Gerhard WE

WWW, 2007

Mengdi Huang (mengdih@seas.upenn.edu)

02/11/2019

Penn Engineering

Problem

- **The Wikipedia categories' hierarchy is barely useful for ontological purposes.**
  - eg: Zidane is in the super-category named "Football in France", but Zidane is a football player and not a football
- **Despite its clean taxonomy of concepts, WordNet lacks knowledge about individuals like persons, organizations, products, etc. with their semantic relationships; its number of facts is also limited.**

- **Disadvantages of current Ontologies**
  - Not scalable to Non-Canonical Facts (TextRunner)
  - Non-defined Relations, and Domains (DBPedia)
  - Evaluation Results Missing (SemanticWikipedia)

Motivation

- Given the differences between existing ontologies and their disadvantages, would it help to combine them?
- Yes! Applications that utilize ontology can boost their performance if a high coverage and quality ontology with knowledge from several sources is available.
- Can we build a new ontology from multiple sources that profit, on one hand, from the vast amount of individuals known to Wikipedia, while exploiting, on the other hand, the clean taxonomy of concepts from WordNet?

Motivation

- YAGO Facts:
  - Input: Wikipedia infoboxes, WordNet (and GeoNames in later versions)
  - Output: KG with 1 million entities and 5 million facts (2007)

    → KG with 350K entity types, 10M entities, 120M facts (2016)

    Thomas Rebele, Fabian M. Suchanek et. al, ISWC 2016

  - Temporal and spatial information
- Challenges
  - The results of automatic information extraction approach contain many false positives, e.g., IsA(Aachen Cathedral, City), thus requires heavy quality control
  - Man-made ontologies suffer from low coverage, high cost for assembly and quality assurance, and fast aging. No human-made ontology knows the most recent Windows version or the latest soccer stars.

Contents:

- Problem & Motivation
- Review: Ontology
- Previous Approaches (SoTA)
- Contributions
- Results & Analysis
- Conclusions
- Shortcomings & Future Work
- Other work that cited YAGO

Review: Ontology

- Knowledge represented as a set of concepts within a domain, and a relationship between those concepts.

  http://en.wikipedia.org/wiki/Ontology (information science)

- In general it has the following components:
  - Entities
  - Relations
  - Domains
  - Rules
  - Axioms, etc.

Review: Ontology - Usage

- **Used in numerous fields of Semantic Web, and other fields like:**
  - Machine translation: WordNet's synsets to resolve pattern disambiguity for sentence translation (Chatterjee et al., 2005)
  - Word Sense Disambiguation: Wikipedia's categories, hyperlinks, and disambiguous articles, used to create a dataset of named entity (Bunescu R., and Pasca M., 2006).
  - Query Expansion: WordNet's synsets used to create different synonyms, hyphonyms of a query term (Liu et al., 2004).
  - Document Classification: WordNet's synsets used as concepts to create a link between word-concept to concept-document, in order to create a conditional probability distribution used later to classify documents (Ifrim G., and Weikum G., 2006).
  - Question Answering, Information Retrieval, Record Linkage, Data Cleaning...

Previous approaches (SoTA)

- **Approach 1: Manual Assembling**
  - eg: WordNet, Cyc or OpenCyc, SUMO -- Man-made ontologies suffer from low coverage, high cost for assembly and quality assurance, and fast aging. No human-made ontology knows the most recent Windows version or the latest soccer stars.
- **Approach 2: Automatic Information Extraction**
  - eg: KnowItAll -- results contain many false positives, e.g., IsA(Aachen Cathedral, City), thus requires heavy quality control
- **Disadvantages of current Ontologies**
  - Non-Canonical Facts (TextRunner)
  - Non-defined Relations, and Domains (DBPedia)
  - Evaluation Results Missing (SemanticWikipedia)

Contributions of YAGO

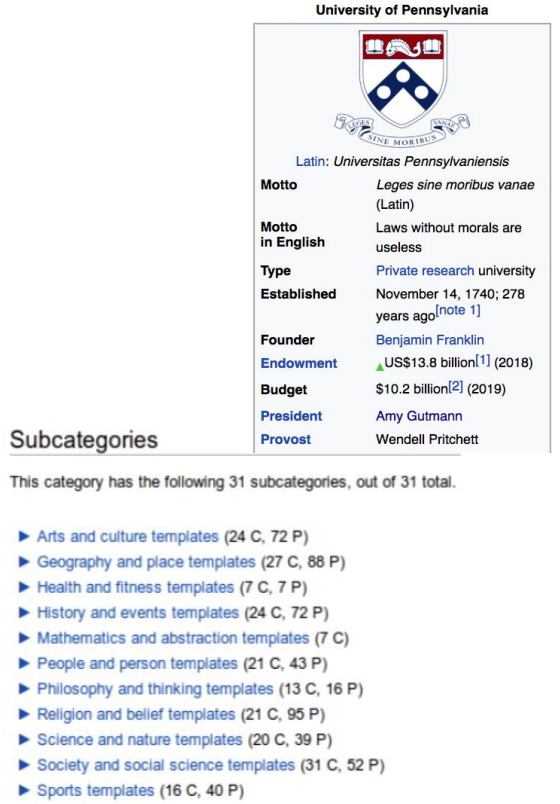
- Key contributions:
  - Rich Ontology: Linking Wikipedia categories to WordNet
  - High Quality: High precision extractions (~95%)

- Other contributions:
  - YAGO is decidable, extensible, and compatible with RDFS

Contributions of YAGO

YAGO emerged for the need of creating a bigger ontology using current existing ontologies, its main aims were:

- Unification of Wikipedia and WordNet.
- Make use of rich structures and information, such as: Infoboxes, Category Pages, etc.
- Ensure plausibility of facts via type checking.



Model Structures

- Representation Model Structure in YAGO:
  - Data Model: extension to RDFS, includes acyclic transitivity (atr).
  - Entities: abstract ontological objects, with the following properties:
    - Each entity is part of at least one class,
    - Classes arranged in taxonomic hierarchy,
    - Relations are entities (express transitivity of relations - atr),
    - Facts are the triple: entity, relation, entity,
    - Each fact has an identifier.

YAGO Entities (n-ary) relation

fact:1 #1 Elvis Presley BornInYear 1935

fact:2 #2 #1 FoundIn Wikipedia

Elvis Presley BornInYear 1935 FoundIn Wikipedia

N-ary (n entities involved in the relationship)

It is based on the assumption that for each n-ary relation,
a primary pair of its arguments can be identified.
The primary pair can be represented as a binary fact with a fact identifier.
All other arguments can be represented as relations that hold between the primary pair and the other argument.

Semantics

- Semantics & YAGO's description through Reification Graphs:
  - Reification Graphs
    - finite set of common entities C.
    - finite set of fact identifiers I.
    - finite set of relation names R.

$$y : \mathcal{I} \rightarrow (\mathcal{I} \cup \mathcal{C} \cup \mathcal{R}) \times \mathcal{R} \times (\mathcal{I} \cup \mathcal{C} \cup \mathcal{R})$$

1. Minimal set of common entities:
   $\mathcal{C} = \{\text{entity, class, relation, atr}\}$
2. Minimal set of relation names:
   $\mathcal{R} = \{\text{type, subClassOf, domain, range, subRelationOf}\}$

1. **Equivalency**: two ontologies $\mathbf{y_1}, \mathbf{y_2}$ are equivalent if the fact identifiers in $\mathbf{y_2}$ can be renamed by a bijective substitution.
2. **Consistency**: an ontology **y** is called consistent **iff** there exists a model for it.
3. **Query language**: a pattern for a reification graph $G_{N,I,L}$ over a set of variables V, $V \cap (N \cup I \cup L) = \emptyset$ is a reification graph over set of nodes $N \cup V$, the set of identifiers $I \cup V$, and the set of labels $L \cup V$. A matching of a pattern P for a graph G is a substitution $\sigma : V \rightarrow N \cup I \cup L$, such that $\sigma(P) \subset G$, is called a match.

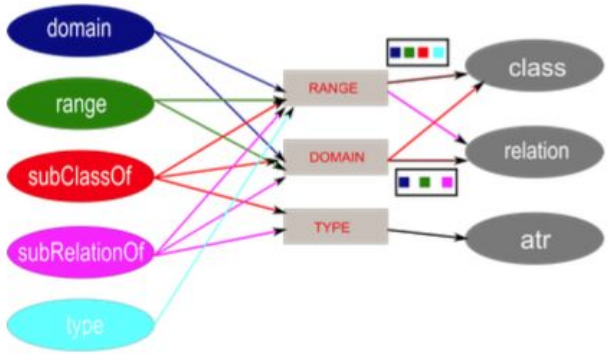
YAGO Model

1. *all facts of y are true in structure*
2. *if* $\Psi(x, TYPE, string)$ *for some x, then* $\mathcal{D}(x) = x$
3. *if* $\Psi(r, TYPE, atr)$ *for some r, then* $\nexists x : s.t.\ \Psi(x, r, x)$

Semantics

Semantics and operations over the YAGO ontology, are defined as follows:

1. facts: $\mathcal{F} : \mathcal{I} \rightarrow (\mathcal{I} \cup \mathcal{C} \cup \mathcal{R}) \times \mathcal{R} \times (\mathcal{I} \cup \mathcal{C} \cup \mathcal{R})$
2. rewrite system: $F \cup \{f_1, \ldots, f_n\} \rightarrow F \cup \{f_1, \ldots, f_n\} \cup f, \forall F \subseteq \mathcal{F}$
3. axiomatic rules for the rewrite system:



$\forall r, r_1, r_2 \in \mathcal{R}, x, y, c, c_1, c_2 \in \mathcal{I} \cup \mathcal{C} \cup \mathcal{R}, r_1 \neq TYPE,$
$r_2 \neq SubRelationOf, r \neq SubRelationOf, r \neq TYPE$
$c \neq atr, c_2 \neq atr$
(1) $\{(r_1, SubRelationOf, r_2), (x, r_1, y)\} \hookrightarrow (x, r_2, y)$
(2) $\{(r, TYPE, atr), (x, r, y), (y, r, z)\} \hookrightarrow (x, r, z)$
(3) $\{(r, DOMAIN, c), (x, r, y)\} \hookrightarrow (x, TYPE, c)$
(4) $\{(r, RANGE, c), (x, r, y)\} \hookrightarrow (y, TYPE, c)$
(5) $\{(x, TYPE, c_1), (c_1, SubClassOf, c_2)\} \hookrightarrow (x, TYPE, c_2)$

Information Extraction

The extracted classes are parsed into pre/post(-modifiers) and head compound, using *Noun Group Parser* (Suchanek F. et al., 2006), and the procedure is as follows:

Function wiki2wordnet(c)
Input: Wikipedia category name c
Output: WordNet synset

```
1   head =headCompound(c)
2   pre =preModifier(c)
3   post =postModifier(c)
4   head =stem(head)
5   If there is a WordNet synset s for pre + head
6      return s
7   If there are WordNet synsets s_1, ...s_n for head
8                 (ordered by their frequency for head)
9      return s_1
10  fail
```

- $\exists$ synset for pre-modifier, and head compound, then Wikipedia class becomes a subclass of the WordNet class.
- $\nexists$ synset, map head compound to the WordNet synset with the highest frequency.

Information Extraction

For example, a category name like Naturalized citizens of the United Statesis broken into a pre-modifier (Naturalized), a head (citizens) and a post-modifier (of the United States). Heuristically, we found that if the head of the category name is a plural word, the category is most likely a conceptual category. We used the Pling-Stemmer from [26] to reliably identify and stem plural words. This gives us a (possibly empty) set of conceptual categories for each Wikipedia page.

Other Heuristics - Information Extraction

1. **Synsets:** exploit the meaning of different classes from wikipedia, i.e. *metropolis*, and *urban center* both are equivalent to the synset *city*.
2. **Redirects:** using Wikipedia redirects to add valuable information to the relation MEANS, i.e. Einstein, Albert MEANS Albert Einstein.
3. **Person Names:** extract name components to add information to relations such as: GivenNameOf, FamilyNameOf.
4. **Relational Categories:** it adds valuable information from the Wikipedia Categories, i.e Rivers in Germany, can be extracted information such as LocatedIn, etc.
5. **Language Categories:** for some Wikipedia categories, exist the equivalent category name in different languages, thus is extracted information for relations such as: IsCalled, and InLanguage, i.e. London IsCalled 'Londres' InLanguage French.

Quality Control

The purpose of quality control was to deliver high quality ontology, and overcome the drawbacks of previous ontologies:

- **Canonicalization**: each fact and entity has a unique reference
  - **Redirect Resolution**: incorrect typed candidate facts, are resolved into correct ones, using Wikipedia redirects.
  - **Duplicate facts**: removes duplicate facts, and only the most precise are kept, i.e, birthday 1935-01-08 is favored instead of 1935.
- **Type Checking**: checks the plausibility of generated facts:
  - **Reductive**: for a candidate fact that there couldn't be determined a class, is removed. While for facts that their class is not in the expected domain those are removed too.
  - **Inductive**: for entities with a birthdate in most cases those are persons, thus if they don't have a class, for those is assigned *person* as a class rather than eliminating them.

Manual Evaluation - Accuracy

Table 1: Accuracy of YAGO

Relation	# evaluated facts	Accuracy	
subClassOf	298	97.70%	± 1.59%
type	343	94.54%	± 2.36%
familyNameOf	221	97.81%	± 1.75%
givenNameOf	161	97.62%	± 2.08%
establishedIn	170	90.84%	± 4.28%
bornInYear	170	93.14%	± 3.71%
diedInYear	147	98.72%	± 1.30%
locatedIn	180	98.41%	± 1.52%
politicianOf	176	92.43%	± 3.93%
writtenInYear	172	94.35%	± 3.33%
hasWonPrize	122	98.47%	± 1.57%

Note that not everybody may agree on the definition of synsets in WordNet (e.g., a palace is in the same synset as a castlein WordNet). These cases of disputability are inherent even to human-made ontologies.

Manual Evaluation - Size

Table 2: Size of YAGO (facts)

Relation	Domain	Range	# Facts
SUBCLASSOF	class	class	143,210
TYPE	entity	class	1,901,130
CONTEXT	entity	entity	~40,000,000
DESCRIBES	word	entity	986,628
BORNINYEAR	person	year	188,128
DIEDINYEAR	person	year	92,607
ESTABLISHEDIN	entity	year	13,619
LOCATEDIN	object	region	59,716
WRITTENINYEAR	book	year	9,670
POLITICIANOF	organization	person	3,599
HASWONPRIZE	person	prize	1,016
MEANS	word	entity	1,598,684
FAMILYNAMEOF	word	person	223,194
GIVENNAMEOF	word	person	217,132

Table 3 shows the number of entities in YAGO.

Table 3: Size of YAGO (entities)

Relations	14
Classes	149,162
Individuals (without words)	907,462

Sample Usage

Table 6: Sample queries on YAGO

Query	Result
When was "Mostly Harmless" written? (Mostly_Harmless, WRITTENINYEAR, $y)	$y=1992
Which humanists were born in 1879? ($h, TYPE SUBCLASSOF*, humanist) ($h, BORNINYEAR, 1879)	$h=Albert_Einstein and 2 more
Which locations in Texas and Illinois bear the same name? ($t, LOCATEDIN, Texas) ($n, MEANS, $t) ($n, MEANS, $k) ($k, LOCATEDIN, Illinois)	$n="Farmersville" and 121 more

Conclusions

Main features of YAGO, and its contributions:

1. High coverage and high quality ontology.
2. Integration of two largest ontologies Wikipedia, and WordNet.
3. Usage of structured information such as *Infoboxes*, *Wikipedia Categories*, *WordNet Synsets*.
4. Introduction of a new data model, *YAGO Model*.
5. Expression of acyclic transitive relations.
6. Type checking, ensuring that only plausible facts are contained.
7. Canonical facts.
8. Query Engine, etc.

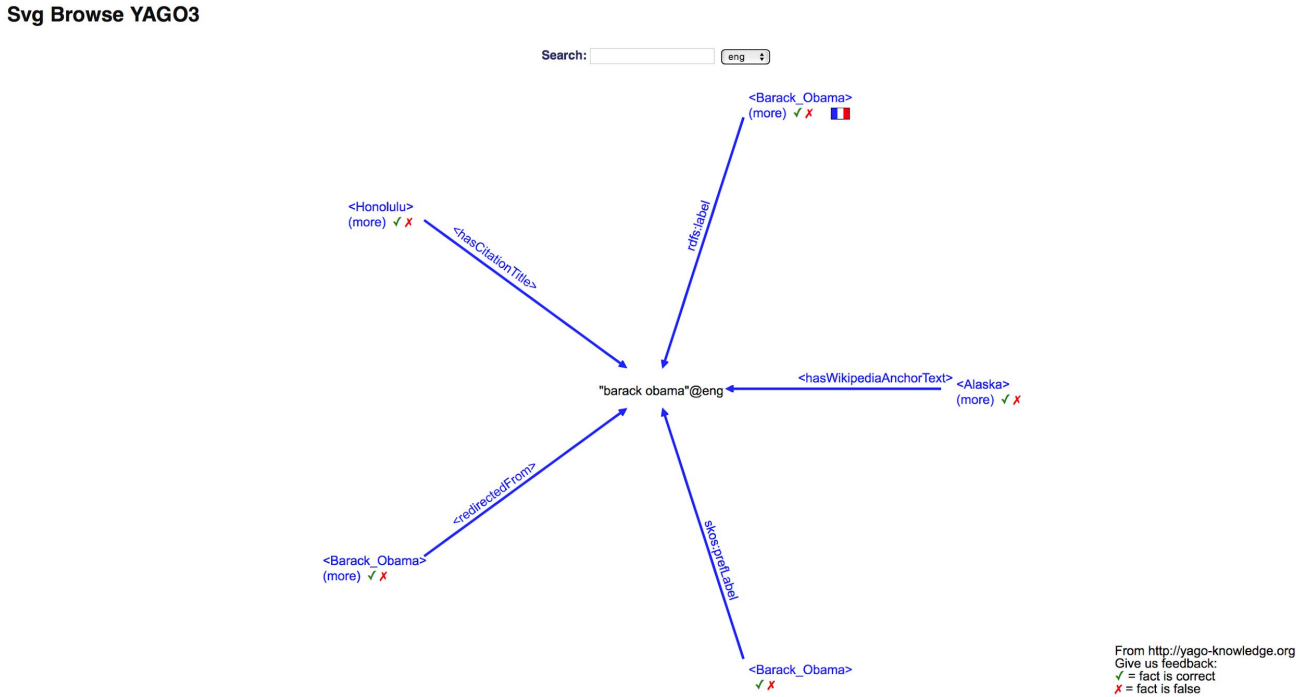
Conclusions

Figure: YAGO and the relation to other Ontologies.

Shortcomings, Possible Solutions and Future Work (YAGO2)

1. New types of rules: **Factual rules, Implication rules, Replacement rules, Extraction rules**
2. Extracting information from different point of views:
   1. Temporal Dimension: Assign begin and/or end of time spans to all entries, facts, events, etc.
   2. Geo-Spatial Dimension: assign location in space to all entities having a permanent location.
   3. Textual Dimension: extract information from Wikipedia, for relation such as hasWikipediaAnchorText, hasCitationTitle, etc, and also multi-lingual is considered in this dimension.
3. 80 million facts in YAGO2 with near-human quality.

Shortcomings, Possible Solutions and Future Work (YAGO2)



https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/demo/

Thank you!