YAGO: a Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames

Thomas Rebele[1], Fabian Suchanek[1], Johannes Hoffart[2],
Joanna Biega[2], Erdal Kuzey[2], Gerhard Weikum[2]

[1] Télécom ParisTech,
46 rue Barrault, 75013 Paris, France
http://www.telecom-paristech.fr
[2] Max Planck Institute for Informatics,
Campus E1 4, 66123 Saarbrücken,
http://www.mpi-inf.mpg.de/

Abstract. YAGO is a large knowledge base that is built automatically from Wikipedia, WordNet and GeoNames. The project combines information from Wikipedias in 10 different languages into a coherent whole, thus giving the knowledge a multilingual dimension. It also attaches spatial and temporal information to many facts, and thus allows the user to query the data over space and time. YAGO focuses on extraction quality and achieves a manually evaluated precision of 95%. In this paper, we explain how YAGO is built from its sources, how its quality is evaluated, how a user can access it, and how other projects utilize it.

Keywords: knowledge base, Wikipedia, WordNet, Geonames

1 Introduction

A knowledge base (KB) is a computer-processable collection of knowledge about the world. A KB usually contains *entities* such as Elvis Presley, Stanford University, or the city of Kobe in Japan. It also contains *facts* about these entities, such as the fact that Elvis Presley plays guitar, that Stanford is a university, or that Kobe is located in Japan. KBs find applications in areas such as machine translation, question answering, and semantic search. Early approaches to create such KBs were mostly manual. With the growth of the Web, more and more approaches constructed KBs automatically by extracting information from Web corpora. Some of the more prominent approaches are YAGO, DBpedia, Wikidata, NELL, and Google's Knowledge Vault. Some of these approaches focus on Wikipedia, the free online encyclopedia.

In this paper, we describe one of the earliest approaches in this direction: The YAGO knowledge base [22]. It was the first academic project to build a KB from Wikipedia, closely followed by the DBpedia project [1]. The particular focus in YAGO has been on precision, i.e., on the correctness of the extracted facts. By sending the extracted facts through a sequence of filters, YAGO achieves a

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-46547-0_19

precision of 95%. Today, YAGO is a larger project at the Max Planck Institute for Informatics and Télécom ParisTech University. The KB draws on several sources by now, including WordNet and Geonames, and has grown to 16 million entities and more than 100 million facts. It is part of the Linked Open Data cloud.

This paper is structured as follows. Section 2 gives an overview of YAGO. Section 3 describes the construction of the KB. Section 4 illustrates data formats and tools. Section 5 shows applications of YAGO before Section 6 concludes.

2 The YAGO Knowledge Base

2.1 History

The YAGO project started in 2006 from a simple idea: Wikipedia contains a large number of instances, such as singers, movies, or cities. However, its hierarchy of categories is not directly suitable as a taxonomy. WordNet, on the other hand, has a very elaborate taxonomy, but a rather low recall on instances. It thus seemed promising to combine both resources to get the best of the two worlds.

The first version of YAGO [22] extracted facts mainly from the category names of the English Wikipedia. With the first upgrade of YAGO in 2008 [23], the project started extracting also from the infoboxes. In 2010, we started working on the extraction of temporal and geographical meta-facts, which resulted in YAGO2 [11,12]. The system architecture was completely restructured for YAGO2s [2] in 2013. This helped us for YAGO3 [19], which added extraction from 10 different Wikipedia languages in 2015.

The YAGO project shares its goal with other KB projects, most notably DBpedia [1,17], Wikidata [27], and the Google Knowledge Vault [5]. Unlike the Knowledge Vault, YAGO is publicly available for download. Unlike DBpedia and Wikidata, YAGO is not constructed through crowdsourcing, but through information extraction and merging. The YAGO project puts a particular focus on the quality of its data, which is assessed through regular manual evaluations. It also has a rather elaborate taxonomy in comparison to other projects, which it inherits from WordNet [6]. YAGO also integrates several multilingual sources into a single KB. Finally, YAGO pays particular attention to the anchoring of the facts in time and space.

2.2 Content

YAGO facts follows the RDF model [28], where facts are represented by triples of a subject, a predicate, and an object. An example is

<Barack_Obama> <wasBornOnDate> "1961-08-04"^^xsd:date.

YAGO gives each fact a *fact identifier*. For example, the above fact has the fact identifier <id_1km2mmx_1xk_17y5fnj>. This allows YAGO to state temporal or spatial information, or the origin of facts. We can say, e.g., that the above fact was extracted from the English Wikipedia page about Barack Obama:

<id_1km2mmx_1xk_17y5fnj> <extractionSource>
<http://en.wikipedia.org/wiki/Barack_Obama>.

YAGO covers topics of general interest such as geographical entities, personalities of public life or history, movies, and organizations. For this, YAGO uses a manually predefined set of 76 relations. In total, the KB contains 16 927 153 entities and 1 185 433 982 triples. The triples are partitioned into *themes*, which can be downloaded separately. YAGO has the following groups of themes (number of triples in parentheses):

- Taxonomy-related facts (95m): the class hierarchy (570k), types (16m), their transitive closure (78m), and schema information (486).
- A simplified taxonomy with just three layers (17m). It contains the leaf levels of the WordNet taxonomy, the main YAGO branches (person, organization, building, artifact, abstraction, physical entity, and geographical entity), and the root node owl:Thing.
- The main facts (55m), i.e., relations between entities (5m), facts with dates (3m), facts with other literals (1m), and labels (45m)
- GeoNames facts (39m), mainly types, labels, and coordinates of geo-entities.
- Meta-facts (203m), i.e., facts about the origin of facts (201m), as well as their time and location (2m)
- Labels for classes in various languages from the Universal WordNet (787k)
- Links to other KBs (4m), notably to DBpedia (4m), GeoNames (117k) and WordNet identifiers (156k)
- Raw information from Wikipedia in RDF (296m), which other projects can use to avoid parsing of Wikipedia. We provide infobox attributes of entities (72m), the infobox templates that an entity has on its Wikipedia page (5m), the infobox attributes per template (262k), Wikipedia-links between the entities (63m), and the source facts for all of these.
- Redirect links and hyperlink anchor texts from Wikipedia (471m).

3 Construction of YAGO

3.1 Sources

Wikipedia. Most of the information in YAGO comes from Wikipedia, the community-driven online encyclopedia. Wikipedia contains not just textual material, but also a hierarchical category system and structured data in the form of *infoboxes*. As a rule of thumb, each Wikipedia page becomes an entity in YAGO. Facts about these entities are created mainly from Wikipedia Infoboxes, using a set of manually compiled mappings from Infobox attributes to YAGO relations. Entity types are extracted from the Wikipedia leaf level categories. The upper part of the Wikipedia class hierarchy is discarded.

Temporal Knowledge. YAGO extracts the time span of facts by hand-crafted regular expressions from the Wikipedia infoboxes and categories. For example, from the infobox excerpt from Cristiano Ronaldo's Wikipedia page

| years2 = 2003–2009 |clubs2 = [[Manchester United F.C.]]

YAGO extracts the the start time and end time of the fact <Cristiano_Ronaldo> <playsForTeam> <Manchester United F.C.>. YAGO stores time points as `xsd:date` literals attached to the fact id of the original fact. If a date contains only the year and month, YAGO uses placeholders, as in "2003-12-##".

WordNet. The WordNet KB [6] is a lexical database of the English language [20]. Among other things, it defines a taxonomy of nouns (e.g. ballet dancer is a hyponym of dancer). YAGO takes the leaves of the Wikipedia category hierarchy and links them to WordNet synsets. This yields, e.g.

<wikicat_Norwegian_ballet_dancers>
rdfs:subClassOf ⟶ <wordnet_ballet_dancer_109834699>
rdfs:subClassOf ⟶ <wordnet_dancer_109989502>

YAGO includes WordNet Domains [18], which groups words into 167 thematic domains, and allows, e.g., searching for entities related to "computer science". The Universal WordNet [4] extends WordNet to over 200 languages, and YAGO uses it to add labels in many languages to the WordNet classes in YAGO.

GeoNames. The GeoNames KB[3] contains 7m geographical entities such as villages, cities, and notable buildings. It contains a class hierarchy and facts such as `locatedIn` facts for cities and countries. GeoNames provides links to Wikipedia, which we use to map the entities to YAGO entities. The GeoNames classes are mapped to WordNet classes by a heuristic defined on the token-overlap of their description. The precision of this matching heuristic is 94.1%, with a recall of 86.7% [12].

3.2 Extraction process

Architecture. In YAGO, an *extractor* is a small code module that is responsible for a single, well-defined extraction subtask. An extractor takes certain themes as input, and produces certain themes as output. Therefore, the architecture of the YAGO extraction system can be represented as a bipartite graph of extractors and themes. This architecture allows for parallelization of the extraction process: Each extractor provides a list of input themes and a list of output themes, and each extractor is started by a scheduler as soon as its input becomes available [2].

Filtering. While the initial extractors are responsible for extracting raw facts from the sources, the following extractors are responsible for cleaning these facts. The facts first undergo redirection, a process where entities are replaced by their canonical versions in Wikipedia. They are then de-duplicated, and sent through various syntactic and semantic checks. Most notably, the facts are checked for compliance with the type signatures of the relations [2,11,12,15,23].

The modular architecture proved useful when YAGO was made multilingual [19]. Only 3 major new extractors had to be added for the translation of entities. After that, the translated facts later undergo the same procedures as the facts obtained from the English Wikipedia [19].

[3] `http://www.geonames.org/`

3.3 Evaluation

Every major release of YAGO is evaluated for quality. Since there is no high quality gold standard of comparable size, this evaluation is done manually. Since the large number of facts in YAGO makes a complete manual evaluation infeasible, we evaluate a randomly chosen sample of facts for every relation. We evaluate only facts obtained by information extraction (not, e.g., imported facts). Facts are evaluated with respect to the extraction source (Wikipedia).

We developed a Web tool that presents a fact with the corresponding Wikipedia pages to a human judge. The judge clicks on “correct”, “incorrect” or “ignore”, and procedes to the next fact. As YAGO3 extracts facts from Wikipedias in several languages, we extended the tool so as to show the Wikipedia pages of the corresponding language and of the time of the Wikipedia dump.

The last evaluation of YAGO was made in 2015, and took two months. 15 people participated and evaluated 4 412 facts of 76 relations, which contain 60m facts in total. They judged 98% of the facts in the sample to be correct. To verify the statistical significance of this result, we calculate the Wilson interval [3]. Weighted by the number of facts, the interval has a center of 95% and a width of 4.19%. This means that the true ratio of correct facts in YAGO lies between 91% and 99%, with $\alpha = 95\%$ probability.[4]

4 Infrastructure

Data format. We provide YAGO in two formats, TTL (Terse RDF Triple Language, also called Turtle)[5] and TSV (Tab Separated Values). The TTL format allows using YAGO with standard Semantic Web software such as Apache Jena. Since TTL does not support fact identifiers directly, we store a fact identifier in a comment that precedes the fact. The TSV format allows users to easily import the facts into a database, or to handle the data programmatically. The format also allows storing fact identifiers as an additional column. We provide a script for importing the TSV files into an SQL database.

Users can download YAGO from the Webpage of the Max-Planck Institute for Informatics[6]. We further published the newest version, YAGO3, to Datahub[7]. The Creative Commons Attribution 3.0 License allows everyone to use YAGO, as long as the origin of the data is credited. YAGO follows the FAIR principles (Findable, Accessible, Interoperable, and Re-usable), thanks to the use of the standard TTL format, its copious metadata, and its open license.

YAGO is an active research project, and the teams at the Max-Planck Institute for Informatics and at Télécom ParisTech provide support and maintenance. Since every major revision of YAGO is evaluated manually, YAGO is updated in the rhythm of months or years.

[4] see `https://w3id.org/yago/statistics` for the complete statistics

[5] see `https://www.w3.org/TR/turtle/` for specifications

[6] `https://w3id.org/yago`

[7] `https://datahub.io/dataset/yago`

Tools. We provide several tools to explore the data in YAGO. A graph browser[8] visualizes an entity with its in- and outgoing edges arranged in a star shape. Users can navigate the graph by clicking on an entity. Edges with the same direction and label are grouped together. Flags indicate the origin of the particular fact. The SPOTLX browser (Subject, Predicate, Object, Time, Location, conteXt)[9] allows querying YAGO with spatial and temporal visualizations. Users can ask questions such as "Which politicians born before 1900 were also scientists?". We also provide example queries. The Data Science Center of Paris-Saclay offers a SPARQL endpoint for YAGO[10], together with example SPARQL queries[11].

5 Applications of YAGO

DBPedia. This project [1] is a community effort to extract a KB from Wikipedia. The KB uses two taxonomies in parallel: a hand-crafted one from its contributors, and the YAGO taxonomy. For this purpose, the `type` and `subclassOf` facts from YAGO are imported into a proper namespace in DBpedia.
IBM Watson. The Watson system [7] can answer questions in natural language. It uses several data sources, among them the type hierarchy of YAGO. Watson participated in the TV quizz show *Jeopardy* together with human players, and was awarded the first place.
AIDA. The AIDA system [13] can find names of entities in text documents, and map them to the corresponding YAGO entities. For example, in the sentence "When *Page* played *Kashmir* at *Knebworth*, his *Les Paul* was uniquely tuned.", AIDA recognizes the names in italics using a graph algorithm and entity similarity measures. AIDA can understand that "Page" here refers to *Jimmy Page* of Led Zeppelin fame (and not, e.g., to *Larry Page*), and that "Kashmir" means the song, not the region. YAGO is also used to resolve temporal references such as "the presidency of Obama" or "the second term of Merkel" [16].
Semantic Culturomics. YAGO has been used to annotate articles of the French journal *Le Monde* with entities from the KB [14]. These annotations allow to compute statistics on entities over time, such as: What are the countries where many foreign companies operate (are mentioned)? What is the proportion of women mentioned in Le Monde, and how did it change over time? The combination of structured knowledge (from YAGO) and unstructured knowledge (from Le Monde) illustrates correlations not visible in these resources alone.

6 Conclusions and future work

YAGO is a knowledge base that unifies information from Wikipedia, WordNet and GeoNames into a coherent whole. In this paper, we have described the

[8] `https://w3id.org/yago/svgbrowser`
[9] `https://w3id.org/yago/demo`
[10] `https://w3id.org/yago/sparql`
[11] `https://w3id.org/yago/dataset`

sources, the extraction process, and the applications of YAGO. For future work, we want to extend the knowledge of YAGO along the following dimensions:
Release cycle. Reducing the manual work required for the evaluation could shorten the release cycle. Many relations such as <isLocatedIn> or <wasBornOnDate> will retain almost all facts from the previous version. We could therefore reuse a part of the previously evaluated facts and combined with a manually evaluated proportional sample of the new facts. We will investigate how to assure the validity of the precision estimation.
Textual extension. The textual source of the facts often contains additional subtleties that cannot be captured in triples. We are therefore working on an extended knowledge graph that allows text phrases in the positions of the triples [29]. We are also working on extracting commercial products from the Web [24].
Commonsense knowledge. Properties of everyday objects (e.g. that spiders have eight legs) and general concepts are of importance for text understanding, sentiment analysis, and even object recognition in images and videos. We have started this line of research recently [25,26].
Intensional knowledge. Commonsense knowledge can also take the form of rules. For example, active sports athletes hardly ever hold political positions. We have already developed methods for mining Horn clauses [8,10], but more general forms of rules need to be tackled [9].
NoRDF. For some information (such as complex events, narratives, or larger contexts), the representation as triples is no longer sufficient. We call this the realm of NoRDF knowledge (in analogy to NoSQL databases), which we want to explore in the near future.

Finally, today's KBs may be correct, but they are hardly ever complete [21].

Acknowledgments This research was partially supported by Labex DigiCosme (project ANR-11-LABEX-0045-DIGICOSME) operated by ANR as part of the program "Investissement d'Avenir" Idex Paris-Saclay (ANR-11-IDEX-0003-02).

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: ISWC. Springer (2007)
2. Biega, J., Kuzey, E., Suchanek, F.M.: Inside YAGO2s: A transparent information extraction architecture. In: WWW demo (2013)
3. Brown, L.D., Cai, T.T., DasGupta, A.: Interval estimation for a binomial proportion. Statistical science pp. 101–117 (2001)
4. De Melo, G., Weikum, G.: Towards a universal wordnet by learning from combined evidence. In: CIKM (2009)
5. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: KDD (2014)
6. Fellbaum, C.: WordNet: An Electronic Lexical Database. Language, speech, and communication, MIT Press (1998)

7. Ferrucci, D.A., Brown, E.W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J.M., Schlaefer, N., Welty, C.A.: Building watson: An overview of the deepqa project. AI Magazine 31(3) (2010)
8. Galarraga, L., Symeonidou, D., Moissinac, J.C.: Rule Mining for Semantifying Wikilinks. In: Linked Open Data Workshop at WWW (2015)
9. Galárraga, L., Suchanek, F.M.: Towards a Numerical Rule Mining Language. In: AKBC workshop (2014)
10. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast Rule Mining in Ontological Knowledge Bases with AMIE+ . In: VLDBJ (2015)
11. Hoffart, J., Suchanek, F.M., Berberich, K., Lewis-Kelham, E., De Melo, G., Weikum, G.: YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In: WWW (2011)
12. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence 194, 28–61 (2013)
13. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust Disambiguation of Named Entities in Text. In: EMNLP (2011)
14. Huet, T., Biega, J.A., Suchanek, F.M.: Mining History with Le Monde . In: AKBC workshop (2013)
15. Kasneci, G., Ramanath, M., Suchanek, F., Weikum, G.: The YAGO-NAGA approach to knowledge discovery. ACM SIGMOD Record 37(4), 41–47 (2009)
16. Kuzey, E., Setty, V., Strötgen, J., Weikum, G.: As time goes by: comprehensive tagging of textual phrases with temporal scopes. In: WWW (2016)
17. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal 6(2) (2015)
18. Magnini, B., Cavaglia, G.: Integrating Subject Field Codes into WordNet. In: LREC (2000)
19. Mahdisoltani, F., Biega, J., Suchanek, F.: YAGO3: A knowledge base from multilingual Wikipedias. In: CIDR (2015)
20. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM 38(11), 39–41 (1995)
21. Razniewski, S., Suchanek, F.M., Nutt, W.: But What Do We Actually Know? . In: AKBC workshop (2016)
22. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW (2007)
23. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Web Semantics 6(3) (2008)
24. Talaika, A., Biega, J.A., Amarilli, A., Suchanek, F.M.: IBEX: Harvesting Entities from the Web Using Unique Identifiers . In: WebDB workshop (2015)
25. Tandon, N., de Melo, G., De, A., Weikum, G.: Knowlywood: Mining Activity Knowledge From Hollywood Narratives. In: CIKM (2015)
26. Tandon, N., de Melo, G., Suchanek, F., Weikum, G.: WebChild: harvesting and organizing commonsense knowledge from the web. In: WSDM (2014)
27. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledge base. Communications of the ACM 57 (2014)
28. W3C: RDF 1.1 Concepts and Abstract Syntax (2014)
29. Yahya, M., Barbosa, D., Berberich, K., Wang, Q., Weikum, G.: Relationship Queries on Extended Knowledge Graphs. In: WSDM (2016)