Knowledge Base Suitable for Answering Questions in Natural Language

TOMASZ BOIŃSKI
Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics
Narutowicza Street 11/12
80-233 Gdańsk
POLAND
tobo@eti.pg.gda.pl

ADRIAN AMBROŻEWICZ
Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics
Narutowicza Street 11/12
80-233 Gdańsk
POLAND
poczta@adiq.pl

JULIAN SZYMAŃSKI
Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics
Narutowicza Street 11/12
80-233 Gdańsk
POLAND
julian.szymanski@eti.pg.gda.pl

Abstract: This paper presents three knowledge bases widely used by researchers coping with natural language processing: OpenCyc, DBpedia and YAGO. They are characterized from the point of view of questions answering system. In this paper a short description of the aforementioned system implementation is also presented.

Key–Words: Knowledge base, Natural language processing

1 Introduction

Growing amount of data available across the Internet is a viable source of knowledge. The problem lies in the form of knowledge representation and ways to access it. Recently many techniques became available to structuralize this knowledge [16, 12, 15]. A few Knowledge Bases also emerges. Among the most often used is OpenCyc [4], DBpedia [10] and YAGO [14]. All those resources create new possibilities enabling creation of different types natural language processing based solutions [6, 13].

In this paper we describe and evaluate aforementioned knowledge bases for use in natural language based question answering system. The paper also presents a short characteristic of the aforementioned system and shortly describes simple implementation used for testing DBpedia knowledge base.

The structure of this paper is as follows. Section 2 presents the aforementioned knowledge bases. Section 3 shortly describes our simple implementation of the question answering system and in Section 4 some conclusions and observations are pointed out.

2 Knowledge Bases

2.1 OpenCyc

Cyc [8] is a rule based expert system with common sense knowledge base. It contains knowledge consisting of very high number of concepts and rules defined around those concepts that describe everyday real life situations, like shopping, relations between people, time, space, abstract concepts etc. The database is also extended by knowledge about grammar and lexis which allows natural language processing.

Cyc knowledge base is utilized in a complex reasoner [4], which allows performing knowledge processing similar to that of a human.

The Cyc System is available in three flavours:

- full, commercial – Cyc,
- free, but limited – OpenCyc,
- intended for research projects – ResearchCyc.

OpenCyc differs from Cyc by a limited knowledge base. It has over 300 000 concepts connected with 3 million assertions. Those assertions are mainly of taxonomical type. Full Cyc knowledge base is filled with other types of assertions and is extended with knowledge from more specific domains. OpenCyc 2.0 System contains:

- Cyc ontology containing all concepts and assertions,
- Reasoning module – Cyc Inference Engine,
- Knowledge Base browser – OpenCyc Browser.
- Links between Cyc concepts and WordNet [7] synsets,
- Links between Cyc and FOAF [3] concepts,
- Links between Cyc concepts and Wikipedia [19] and DBpedia [10] articles,

- English names in canonical and conjugated form,
- Documentation,
- CycL [11] language specification,
- Cyc API.

2.1.1 Cyc Knowledge Base

OpenCyc is shipped with a knowledge base containing over 300 000 concepts that are connected with 15 000 relations using 3 million of taxonomical assertions. New assertions are constantly added to the database, mainly as a result of knowledge inference. All concepts in knowledge base are treated as keywords of CycL [11] language which is a formal representation of Cyc knowledge.

The knowledge in Cyc is organized as a pyramid. The top concept is *Thing*, which is a top concept of Cyc ontology. Directly below *Thing* abstract concepts and knowledge about possible relations is located. The concepts become more and more specific towards the base of the pyramid. At the bottom lies specific knowledge in form of facts and data (like names) which are connected with concepts in the knowledge base.

The knowledge base is divided into multiple micro-theories that share common assumptions. Every micro-theory is focused on given knowledge domain, level of detail or time frame. Such micro-theory mechanism allow keeping in knowledge base seemingly contradictory assertions – each micro-theory have to has consistent assertions but the whole knowledge base can have contradictory ones. Such organisation of knowledge speeds up reasoning, increases scalability of the solution and allows usage of different reasoners for different knowledge types.

2.1.2 Reasoning Engine

OpenCyc is shipped with integrated reasoning engine that allows users to create new assertions and add data to the database. The database recognizes two types of basic assertions:

- facts – Tom is a student.
- rules – If a person X is a student than that person has an index.

New facts can be generated by the knowledge base using resolution. When a new fact is added the reasoning engine infers new facts derived from the new one and adds them to the database. New facts can also be added as a by-product of a query.

2.1.3 CycL language

CycL is an LISP based language used to represent knowledge in Cyc. First order logic was extended by elements of Second order logic, skolemization and nonmonotonic reasoning. The CycL dictionary is composed of terms, that can be divided to:

- constants,
- variables,
- non-atomic terms,
- micro-theories.

The terms are connected into sentences that form assertions that are stored in the knowledge base. Sentences in CycL are also used to form queries to Cyc reasoning engine.

2.1.4 NLP Subsystem

This is currently the most actively developed part of Cyc. It is used to interpret facts and questions formed using English language. The NLP System is composed of a lexicon, syntactic parser and semantic interpreter. The lexicon contains information about syntax and semantics of English language. The semantic parser is used to create sentences out of symbols within the lexicon. Such sentences are transformed into CycL syntax and verified by the semantic interpreter.

2.1.5 Wikipedia Links

OpenCyc 2.0 was extended to provide links to Wikipedia articles. Some of the constants are part of wikipediaArticleName and wikipediaArticleURL predicates which connects them with proper article. Currently there are 19103 links defined to English language Wikipedia. Those links, however limited in number, allow extracting non trivial links between Wikipedia articles using Cyc knowledge base and reasoning capabilities.

2.1.6 Summary

OpenCyc is an extensive and formalized knowledge base. Unfortunately in the free version it mainly provides taxonomical relations which is not enough. Being a lite version of a commercial product it provides powerful system with somewhat limited knowledge base that hinders potential uses of the System. It also provides limited interoperability with widely used solutions like WordNet and Wikipedia what limits its interoperability.

2.2 YAGO

YAGO [14] is a knowledge base developed by Max-Planck-Institut für Informatik in Saarbrücken. It is developed strictly by a dedicated team which takes great care in assuring the quality of stored information. Currently it holds 450 million facts about 10 million entities.

In 2012 an extended version, called YAGO2, was released. It introduced full compatibility with RDF/OWL by providing the knowledge base in Turtle format, the resources were combined into a subsets like Core, GeoNames etc. and the entities were extended by knowledge about domains.

2.2.1 Data Sources

YAGO Knowledge Base is automatically generated based on the following data sources:

- WordNet
  - YAGO class hierarchy is directly derived from WordNet synset hierarchy,
  - Most of the entities stored in YAGO are correlated with WordNet synsets, those correlations are also visible in construction of YAGO ontology.
- GeoNames [18]
  - Entities with identical geographical names are grouped,
  - Wikipedia entries describing location that are described in GeoNames are directly transform into YAGO entities,
  - GeoNames classes are mapped into their WordNet counterparts.
- Wikipedia
  - Information from Wikipedia info boxes are mapped into YAGO facts,
  - Facts are generated based on Wikipedia categories:
    - Aggregation categories ("born in 1990") are parsed and interpreted in YAGO,
    - Leaves in Wikipedia categories structure are prioritized,
    - The articles and categories are mapped into WordNet synsets.

In general YAGO knowledge base is thus an extension of WordNet structure with knowledge from Wikipedia.

2.2.2 Quality

The quality of YAGO knowledge base is verified constantly using questionnaires regarding selected samples of data [9]. The measured accuracy is near 95%. Furthermore tests shown that 87% of GeoNames entries were mapped to WordNet synsets with 94% accuracy.

2.3 DBpedia

DBpedia aims at transforming Wikipedia articles into an RDF compatible database with specified ontology. DBpedia entries are created by automatic conversion of Wikipedia articles into triples in RDF format. Currently transformation includes:

- Title,
- Abstract,
- Geo coordinates,
- Categories,
- Pictures,
- Links,
- Info-boxes.

The most information is gathered from info-boxes that are directly converted into triples.

2.3.1 DBpedia Ontology

Early versions of DBpedia had no ontology. Currently it's structure is extended by such. DBpedia ontology has 529 classes which form a subsumption hierarchy and are described by 2,333 different properties [5]. Unfortunately data extracted from Wikipedia not always is mapped into the ontology. Especially that extracted using old algorithms need further verification and correction.

We performed a preliminary check of the most popular concepts in terms of type and homogeneity of available information. The came to the following conclusions:

- The highest homogeneity is characteristic to entries describing well defined and unchangeable terms like city, country, music record, language. Information stored in records related to each other are consistent and coherent,
- In other cases heterogeneity is observed that is dependent on the domain of the concept. Persons were described in different way dependent on

their profession. Each category has its own separate ontology incompatible with other ontologies. Currently there is work done that should eliminate the simple property to ontology mapping and standardize the description of a concept but most of the entities are still not manually transformed to this new dbpedia-owl ontology.

- Low number of relations between entries. In most cases records are connected with each other only by links to common categories.

2.4 Differences between YAGO and DBpedia

Both YAGO and DBpedia are based on similar assumptions and realize similar goals. The main difference lies in the way of knowledge acquisition. In general DBpedia is based on community effort of mapping knowledge directly from Wikipedia and YAGO is based on stable heuristics. YAGO is developed by a formal group of developers dedicated to the project whereas DBpedia is developed by a rather loose community.

The main differences between those systems are:

- DBpedia has its own ontology containing 529 classes while YAGOs ontology is based on WordNet taxonomy and has 350 000 classes,
- YAGO provides basic information through relations defined in yagoSchema, DBpedia has no such mechanism,
- DBpedia mappings of info-boxes are performed quite loosely compared to YAGO, attributes are mapped “as is”,
- YAGO’s quality is controlled by manual quality tests on preselected samples of data,
- DBpedia defines concrete classes like “Writer” or “Musician”, whereas YAGO states only a fact that somebody “is a creator”. The concrete information about form of creation can be derived indirectly from type of the work related to given person,
- YAGO distinguishes incomplete dates and allows their comparison with full dates whereas DBpedia does not,
- YAGO does not contain cycles in relations whereas DBpedia does,
- Information stored in DBpedia is more detailed than in YAGO but is domain dependent and inconsistent across different topics.

Both YAGO and DBpedia interchange knowledge. Data from YAGO is being integrated into DBpedia, whereas DBpedia serves as YAGO’s access point to world of Linked Data [1, 2].

YAGO and DBpedia knowledge base can be accessed using online SPARQL endpoints. They are located at `http://lod2.openlinksw.com/sparql` and `http://dbpedia.org/sparql` respectively. Unfortunately both of them are unreliable thus requiring local installments of both systems.

3 Simple Question Answering System

We decided to implement a simple question answering system based on question templates that uses DBpedia as a knowledge base. Its current version allows asking questions to the DBPediia database in two ways. The first of these involves the creation of queries in the form of triples of type ¡entity, property, property value, [¡property, property value¿, ...]. This approach allows the generation of simple queries with a single entity and describing it set the properties of the selected values. The second way is to generate simple queries in natural language formulated according to predefined template. This way a user can ask a question in form of ”which, entity, has, property, property value [and property, the value of property, [and ...]]. Sample question is ”Which country has Australian_Dollar currency and language English”. In both cases it is possible to define a single entity and its properties.

In both cases the system worked surprisingly well allowing prefetching property values in the background based on user input which simplifies query construction. The main difficulty is to transform natural language formed question into a SPARQL query. The consistency of knowledge available in DBpedia is unfortunately quite low. Replies for the same queries but with different subjects were not comparable as there is inconsistency in terms of knowledge available. E.g. different persons are characterized by different properties depending on their occupation, achievements etc. DBpedia would require some kind of standardization in this matter.

The DBpedia SPARQL endpoint worked well when it was available, but unfortunately it provided a lot of downtime proving it unusable for production or even extensive testing. Local installment using Apache Jena [17] was thus necessary.

In future we would like to expand the possibilities of the system by introducing other types of queries that can be forwarded to the knowledge base. The first task is thus the extension of ways to ask questions in natural language by a better analysis of the wording of the question that should go beyond the pre-defined templates. It is therefore necessary to allow other forms of questioning (which, where, whose, when, etc.) and other forms of queries (e.g. ”In Which country you can pay with Australian_Dollar and speak English” or omission explicitly expressed entity ”Where you can pay using Australian_Dollar and speak English”). The next task is to add the ability to use the property values that are not labelled directly in DBpedia resource (e.g. Australian Dollar rather than Australian_Dollar). In the longer term the students should try to introduce complex queries in form of ”Which country has currency that was Introduced on February 14th 1966 and language English”.

4 Conclusion

All three presented systems are widely used by researchers coping with natural language processing. OpenCyc is the most formalized and mature of the solutions but in free version provides limited knowledge in terms of relations between entities. YAGO and DBpedia interlinks with other widely used solutions like WordNet and Wikipedia allowing a formalized gateway to use that resources and create and interoperable system. Currently YAGO is more formalized than DBpedia but provides far less information and currently it seems to be a better solution. DBpedia efforts on standardizing and formalizing its structure can however change this situation and due to larger amount of data available, when backed by formal and consistent structure, can provide a better solution.

In our tests we focused on DBpedia, in spite of its constantly changing and inconsistent knowledge base, due to larger amount of data available. Unfortunately it proved to be unreliable mainly due to lack of formalization and consistency. In the near future we plan on migrating our simple implementation of question answering system to utilize YAGO database and through its formalized structure link with DBpedia. Those two systems interlink with each other so it might seem a proper solution that would allow combine YAGO formalisms with DBpedia extensive database. That however needs a verification.

References:

[1] T. Berners-Lee, Linked data-design issues (2006), *http://www.w3.org/DesignIssues/LinkedData.html*, [Online; accessed 10-March-2014].

[2] C. Bizer, T. Heath, and T. Berners-Lee, Linked data-the story so far, *International journal on semantic web and information systems*, 5(3), pp. 1–22, 2009.

[3] D. Brickley and L. Miller, The Friend of a Friend (FOAF) project, 2000.

[4] J. Curtis, G. Matthews and D. Baxter, On the effective use of Cyc in a question answering system, In *Proc Workshop on Knowledge and Reasoning for Answering Questions*, pp. 61–70, 2005.

[5] DBpedia, The DBpedia Ontology (3.9), 2014, *http://wiki.dbpedia.org/Ontology*, [Online; accessed 10-March-2014].

[6] A. Diosteanu, L. A. Cotfas, A. Smeureanu and S. D. Dumitrescu, Natural language processing applied in itinerary recommender systems. In *Proceedings of the 10th WSEAS international conference on Applied computer and applied computational science*, pp. 260–265, World Scientific and Engineering Academy and Society (WSEAS), 2011.

[7] C. Fellbaum, WordNet: An electronic lexical database, 1998.

[8] D. Foxvog, Cyc, In *Theory and Applications of Ontology: Computer Applications*, pp. 259–278, Springer, 2010.

[9] J. Hoffart, F. M. Suchanek, K. Berberich and G. Weikum, YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia, *Artificial Intelligence*, 194, pp. 28–61, 2013.

[10] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia, *Semantic Web Journal*, 2014.

[11] B. D. Lenat and R. V. Guha, The evolution of CycL, the Cyc representation language, *ACM SIGART Bulletin*, 2(3), pp. 84–87, 1991.

[12] J. Rzeniewicz, J. Szymański and W. Duch, Adaptive algorithm for interactive question-based search, In *Intelligent Information Processing VI*, pp. 186–195, Springer, 2012.

[13] M. D. Seddiqui and A. Masaki, Ontology instance matching by considering semantic link cloud, *9th WSEAS Int. Conf. on Applications of Computer Engineering*, 2010.

[14] F. M. Suchanek, G. Kasneci and G. Weikum, Yago: a core of semantic knowledge, In *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, ACM, 2007.

[15] J. Szymański and W. Duch, Annotating words using wordnet semantic glosses, In *Neural Information Processing*, pp. 180–187, Springer, 2012.

[16] J. Szymański and W. Duch, Information retrieval with semantic memory model, *Cognitive Systems Research*, 14(1), pp. 84–100, 2012.

[17] The Apache Software Foundation, Apache Jena, 2014, *http://jena.apache.org/*, [Online; accessed 10-March-2014].

[18] M. Wick and B. Vatant, The geonames geographical database, *http://geonames.org*, [Online; accessed 10-March-2014].

[19] Wikipedia, About Wikipedia – Wikipedia, The Free Encyclopedia, 2013, *http://en.wikipedia.org/w/index.php?title=About_Wikipedia&oldid=582531310*, [Online; accessed 10-March-2014].