A Survey on Automatically Constructed Universal Knowledge Bases

Bayzid Ashik Hossain, Abdus Salam and Rolf Schwitter

Abstract

A universal knowledge base can be defined as a domain-independent ontology containing instances. Ontologies define the concepts and relations among these concepts and are used to represent a domain of interest. These universal knowledge bases are the elementary units for automated reasoning on the Semantic Web. The Semantic Web is an extension of the World Wide Web which facilitates software agents to share content beyond the limitations of applications and websites. This survey focuses on the most prominent automatically constructed universal knowledge bases including KnowItAll, DBpedia, YAGO, NELL, Probase, BabelNet and Knowledge Vault. We take a closer look at how these knowledge bases are built, in particular at the information extraction and taxonomy generation process and investigate how they are used in practical applications. Due to quality concerns, the most successful and widely employed knowledge bases are manually constructed to maintain high quality, but they suffer from low coverage, high assembly and quality assurance cost. On the contrary automatic approaches for building knowledge bases try to overcome these drawbacks. Although it is strenuous to achieve the same level of quality as for manual knowledge bases, we found that the surveyed automatically constructed knowledge bases have shown promising results and are useful for many real-world applications.

Keywords

Semantic Web, Information Extraction, Ontology, Knowledge base, Taxonomy, RDFS

Introduction

The World Wide Web has become the largest source of data due to the continuous information explosion. Most of this information is available in natural language, therefore understandable by humans, because they possess the background knowledge for the relevant concepts and relations. They know how to apply this background knowledge and to situate the information in the right context. The poorly structured information available on the web is difficult t o p rocess a utomatically a nd m akes t he t ask of understanding text on the web a very challenging one for machines.

The web is the largest source of unstructured data whereas data can be defined a s u nrefined fa cts without any added interpretation. Information is acquired through the extraction, analysis and interpretation of this data. Knowledge is then gained through the understanding of the available information. In order to solve a problem by a machine, knowledge must be represented in a suitable form by a knowledge engineer or must be automatically learned using machine learning techniques. Knowledge engineers build knowledge bases to capture the diversity and complexity of human language by formally defining the concepts and relations between these concepts with various granularities. Since most of the time information is conveyed on the web in form of text, it is critical for a machine to know how words and phrases are used to express meaning and how to represent this meaning in a particular knowledge representation formalism. This knowledge representation formalism specifies the form in which the knowledge is stored in a knowledge base and prescribes what form of reasoning can be conducted.

Knowledge bases provide benefits t o m ost natural language processing tasks, such as text summarization Nastase (2008), named entity recognition Bunescu (2006), question answering Harabagiu et al. (2000), sentiment analysis Taboada et al. (2011), plagiarism detection Barrón-Cedeno et al. (2010), text categorization Wang and Domeniconi (2008); Navigli et al. (2011), co-reference resolution* Ponzetto and Strube (2007); Rahman and Ng (2011), and word sense disambiguation Cuadros and Rigau (2006, 2008); Navigli (2009).

The Semantic Web is an extension of the traditional web that allows software agents to exchange and share information†. The vision of the Semantic Web is to make the information on the web available in machine-readable form Berners-Lee et al. (2001); Shadbolt et al. (2006). Ontologies are – besides linked data – the building blocks of the Semantic Web. The most commonly used ontologies are man-made because of quality issues. While manual approaches for building ontologies suffer from low coverage, high assembly and quality assurance cost, automatic approaches suffer from lack of accuracy. Semi-automatic approaches Kamel et al. (2013); Maltese and Hossain (2012); Wang et al. (2006) were introduced to get the benefits from both manual and automatic approaches.

*https://nlp.stanford.edu/projects/coref.shtml

†http://semanticweb.org/wiki/Main_Page

Although a good number of manually constructed ontologies exist for specific domains (e.g., Ashburner et al. (2000); Bard et al. (2005); Giunchiglia et al.(2010); Caracciolo et al. (2013); Islam et al. (2017)), only a few domain-independent/universal ontologies are available. These universal ontologies share the concept space limitations as most of them were constructed manually (e.g., WordNet*, Freebase†, Wikidata‡, and ResearchCyc§). Some of them (e.g., Freebase and Wikidata) relied on community efforts to increase their coverage.

The automatic knowledge base construction process reduces cost and improves productivity but at the same time is never perfect as this process can introduce errors and inconsistencies during construction. Recently, a number of automatically constructed knowledge bases such as KnowItAll Etzioni et al. (2004), DBpedia Auer et al. (2007), YAGO Suchanek et al. (2007), NELL Carlson et al. (2010), Probase Wu et al. (2012), BabelNet Navigli and Ponzetto (2010) and Knowledge Vault Dong et al. (2014) became popular, but they still have limitations in terms of coverage, concept space and quality. In this paper, we survey these automatically constructed knowledge bases in detail following a chronological order based on their latest release. The goal of this survey is to provide an informative overview about the state-of-the-art of automatically constructed knowledge bases for researchers in academia and developers in industry. In the following discussion, we use the term "knowledge base system" to denote the system that builds and maintains the corresponding knowledge base.

Survey Criteria

Since this survey only includes automatically constructed domain-independent knowledge bases; manually constructed knowledge bases are not covered. Furthermore, we based this survey on the following key criteria in order to make the knowledge bases easily comparable:

Information Extraction. How the information is extracted from unstructured text for a particular knowledge base. This criterion includes both the information extraction scale (extracting information from the whole web or based on a particular site like Wikipedia) and the methodology that has been used.

Taxonomy Generation. How the taxonomy is generated for the knowledge base and whether any external vocabulary is used for this purpose or not.

Characteristics and Features. The specific characteristics and features that distinguish each knowledge base, including:

- *Data Availability:* Whether the datasets that constitute the knowledge base are freely available for public use or not.
- *Data Representation Formalism:* In which format the datasets are available.
- *Temporal and Spatial Aspects:* Does the knowledge base cover temporal and spatial dimensions for the stored information?
- *Statistics:* This includes the total number of concepts, relation types, relations and facts.
- *Usage/Applications:* The kind of applications that are currently using the knowledge base or might use the knowledge base in the future.
- *Canonicalization:* Whether the concepts and relations are stored in canonicalised form or not. This means whether or not the knowledge base contains any redundant information.
- *Multilinguality:* Whether the knowledge base is available in multiple languages or not. This also includes the total number of languages supported by the knowledge base.

KnowItAll

KnowItAll Etzioni et al. (2004) is a large-scale information extraction system that extracts information from the web in an automated fashion. The primary result of this extraction process is a domain-independent knowledge base. KnowItAll was inspired by the WebKB project Craven et al. (1998, 2000) that aimed to build a probabilistic, symbolic knowledge base which stores the content of the World Wide Web¶. KnowItAll was built to address the problem of accumulating large collections of facts and to ease the cumbersome manual searching process over large volumes of information on the web. For example, searching a list of footballers who won the *ballon d'or* might involve a laborious search over many documents unless we find the right document.

The KnowItAll system associates a probability with every fact it extracts from the web with the help of a number of search engines in order to trade recall for precision. It assesses the extracted information using statistics which is computed by treating the web as a large text corpus. Unlike other bootstrap information extraction systems Brin (1998); Riloff et al. (1999); Agichtein and Gravano (2000) that require a small set of domain specific seed instances as input, KnowItAll uses a bootstrapping system for which manually tagged training sentences are not required. Another important feature of KnowItAll is the use of the pointwise mutual information (PMI)-IR Turney (2001) method to evaluate the probability of extracted facts based on web-scale statistics. KnowItAll employs unsupervised learning methods for fact extraction while using the search engines to collect easily understandable sentences scattered throughout the documents of the web. KnowItAll starts with a domain independent set of universal extraction patterns from which it induces a set of seed instances.
KnowItAll's architecture is composed of four modules that run as a thread and communicate with each other through asynchronous message passing. The four modules are: 1. extractor, 2. search engine interface, 3. assessor, and 4. database. KnowItAll is designed to support scalability and high throughput.

*https://wordnet.princeton.edu/

†https://datahub.io/dataset/freebase

‡https://www.wikidata.org/wiki/Wikidata:Main_Page

§http://www.cyc.com/platform/researchcyc/

¶http://www.cs.cmu.edu/∼webkb/

NP1 {“,”} “such as” NPList2
& head(NP1) = plural(Name(Class1))
& properNoun(head(each(NPList2))) ⇒
instanceOf(Class1,head(each(NPList2)))

Figure 1. Rule template instantiation in KnowItAll

Information Extraction

The KnowItAll system extracts facts consisting of classes and relations from the web. It used 12 search engines including Google Brin and Page (1998), AltaVista Seltzer et al. (1996), and Fast*. KnowItAll uses a perpetual ontology and a small number of universal rule templates from which it creates text extraction rules for each class and relation in its ontology. Whenever a new class or relation is added to the ontology, the extractor uses those generic, domain independent rule templates to generate a set of information extraction rules for that new class or relation. Some of KnowItAll's rule templates are adapted from Marti Hearst's Hearst (1992) hyponym patterns.

Let's consider a rule template "NP1 such as NPList2" which is used by the KnowItAll system. This template specifies that every simple noun phrase in NPList2 is an instance of the class name NP1. This template can be used to find for example countries in sentences like *"We visited countries such as America, England, China and Australia"*. Each rule in KnowItAll has an associated search query composed of the keywords from the rule. For example, if we consider the rule stated above, KnowItAll would issue the query *"countries such as"* to a search engine. Thus, it can automatically formulate queries based on the extraction rule. It caches search engine result pages and avoids querying a search engine when the result is already known. Figure 1 shows an example of KnowItAll's rule template instantiation for a particular class in the ontology to create an extraction rule that searches for instances of that particular class.

The extractor uses the Brill tagger Brill (1992) to assign part-of-speech tags to the text of the web page returned by a search engine. The extractor then identifies noun phrases with the help of regular expressions based on those tags.

Taxonomy Generation

KnowItAll uses a form of pointwise mutual information (PMI) between the words and discriminator phrases that is estimated from the hit counts of the web search engine. For example, if the PMI between *"Trento"* and the *"city of Trento"* is high, then this gives evidence that *"Trento"* is a valid instance of the class *"city"*. The assessor calculates the PMI between every extracted instance and multiple phrases corresponding to their classes. These mutual information statistics are integrated via a *naive Bayes Classifier* to assess the probability of an instance belonging to a class. Previous research Banko and Brill (2001) showed that co-occurrence statistics are highly informative when computed over large corpora. KnowItAll uses search engine hit counts for calculating co-occurrence statistics over billions of web pages.

Characteristics

The information extracted by the KnowItAll system is a domain and language independent knowledge base which uses unsupervised learning methods. Though KnowItAll uses a language independent information extraction process, it does not have multiligual support. The system has a fixed initial ontology and populates this ontology with instances but does not learn new classes or relations. The KnowItAll knowledge base stores knowledge including metadata in a conventional database management system. Facts stored in the knowledge base are not canonicalised and not available online for public use. KnowItAll's knowledge base does not store any spatial or temporal information. In summary: searching a large body of information is the main application and motivation of KnowItAll.

DBpedia

The DBpedia knowledge base Auer et al. (2007) is a community-based project. Its aim is to extract structured and multilingual information from Wikipedia and represent it in a formal notation to answer semantically rich queries. DBpedia makes its information freely available on the web using linked data technologies Berners-Lee (2006); Bizer et al. (2007). It provides a large multidomain RDF data set that can be used in various semantic web applications and a SPARQL endpoint† to query data online.

Currently, DBpedia Lehmann et al. (2015) extracts knowledge from 125 language editions of Wikipedia and it evolves automatically taking changes of Wikipedia into consideration. DBpedia defines a globally unique identifier for each entity thus making it easy to be dereferenced on the web following the linked data principles Bizer et al. (2008). DBpedia is the most famous and accepted knowledge base and has become the central linking hub for the emerging web of data Bizer et al. (2009); Färber et al. (2015).

The DBpedia user community maintains the structure of the knowledge base and they create the mappings between Wikipedia information representation structures and the DBpedia ontology. The DBpedia ontology is responsible for the unification of different template structures within single or multiple Wikipedia language editions. The DBpedia project provides a number of interfaces to access the data via web services‡.

Information Extraction

The DBpedia project uses an information extraction framework to extract structured information from Wikipedia and to convert it into a rich, multilingual knowledge base Lehmann et al. (2015).

A Wikipedia article mainly contains free text. It also uses wiki-markup to identify infobox templates, external page links, category information, geo-coordinates, images, and links between different language editions. The information extraction framework extracts this structured information

*https://news.microsoft.com/2008/01/08/microsoft-announces-offer-to-acquire-fast-search-transfer/

†http://dbpedia.org/sparql

‡http://dbpedia.org/fct/

either from a Wikipedia dump published by Wikimedia foundation* or via the MediaWiki API†. After that the wiki parser transforms each Wikipedia page source code into an Abstract Syntax Tree and sends this tree to the extractors. The extractors extract different types of information (e.g., labels, abstracts, and geographical coordinates) from the Abstract Syntax Tree and generate a set of RDF statements. There exist four types of extractors: 1. mapping-based infobox extractors; 2. raw infobox extractors; 3. feature extractors; and 4. statistical extractors.

Extractors that are used for mapping-based infobox extraction follow manually written mappings‡ which relate Wikipedia infoboxes to the terms in the DBpedia ontology. This information extraction produces high quality data as the mapping rules specify the data types for the infobox properties. On the contrary extractors used in the raw infobox extraction process provide a direct mapping between Wikipedia infoboxes and RDF. Unlike mapping-based infobox extraction, the data quality is poorer in raw infobox extraction because it does not rely on curated knowledge.

During the feature extraction process, a single feature (e.g., label and geographic coordinates) is mainly extracted from a Wikipedia article. The extractors are specialized to extract a particular feature type. During the statistical extraction process some NLP related extractors aggregate data from all Wikipedia pages in order to provide data that is based on statistical measures of the number of page links or word count.

Taxonomy Generation

The generation of DBpedia's taxonomy is based on four existing classification schemes to fulfill different application-specific requirements. These schemes consist of the Wikipedia category system, the YAGO classification scheme, the UMBEL (Upper Mapping and Binding Exchange Layer)§ scheme, and the DBpedia ontology Bizer et al. (2009). DBpedia also uses an unsupervised approach that automatically retains a taxonomy from the Wikipedia category system and designates types to DBpedia entities Fossati et al. (2015).

The main benefit of using the Wikipedia category system for the taxonomy generation is that it is expanded collaboratively and updated regularly by a large number of Wikipedia editors. On the other hand, a disadvantage of the category system is that there exist cycles between the categories that often represent a loose connection between the Wikipedia articles. Thus categories in the Wikipedia category system do not form an absolute topical classification.

DBpedia maintains a Simple Knowledge Organization System (SKOS)¶ Miles and Bechhofer (2009) representation of the Wikipedia category system. The advantage of using the YAGO hierarchy is its depth and that it encodes detailed information in the class name (for example the class *MultinationalCompaniesHeadquarteredInTheNetherlands*). YAGO has few errors and omissions due to the automatic generation of the class hierarchy Bizer et al. (2009). DBpedia uses a script to assign YAGO classes to DBpedia entities which has been jointly developed by DBpedia and YAGO.

On the other hand, the UMBEL scheme is a light weight ontology derived from OpenCyc Matuszek et al. (2006). The main objective of OpenCyc is to interlink web content and data. Since OpenCyc classes are derived from Cyc and Cyc is based on WordNet synsets, a mapping from OpenCyc classes to DBpedia classes can be done using UMBEL Bizer et al. (2009). This is done with the help of the fact that YAGO uses WordNet synsets and DBpedia uses YAGO. The alignment between Wikipedia infoboxes and the DBpedia ontology is done by a community provided mapping process. This mapping process helps to normalize the variation of property and class names. Currently, the DBpedia ontology contains 685 classes in the form of a subsumption hierarchy. These classes are described by 2,795 different properties‖.

Characteristics

DBpedia is a multilingual knowledge base that stores entities and relations among these entities in canonicalised form and provides localised versions for 125 languages. The DBpedia knowledge base has a live synchronization system that works on a continuous stream of updates from Wikipedia and reflects these updates in the knowledge base. DBpedia provides a public SPARQL endpoint for the users to query the knowledge base. It also provides dereferencable URIs following the linked data principles Bizer et al. (2008). The DBpedia datasets** are available online for public use. DBpedia represents its datasets in RDF format and does not store spatio-temporal information.

DBpedia has been used for many applications due to its wide coverage. These applications include natural language processing tasks such as entity classification Dojchinovski and Kliegr (2013), entity disambiguation Mendes et al. (2011); Turian (2013); Kobilarov et al. (2009); Tori and Šolc (2008), question answering (e.g., Ferrucci et al. (2010); Unger et al. (2012); Lopez et al. (2012); Damljanovic et al. (2011); Cabrio et al. (2012)). It is also used in applications like digital libraries and archives, for knowledge exploration such as facet-based browsers Heim et al. (2008), for spatial applications Becker and Bizer (2008), explaining content-based recommendations Musto et al. (2016), for searching and querying Heim et al. (2009), for distributed information retrival Han et al. (2018) and visual question answering Wu et al. (2018).

YAGO

YAGO (Yet Another Great Ontology) Suchanek et al. (2007, 2008) is a universal knowledge base constructed from Wikipedia†† and WordNet Miller (1995). YAGO combines both high coverage and high quality. Rather than using any particular information extraction method, the YAGO system

*https://dumps.wikimedia.org/
†https://www.mediawiki.org/wiki/API:Main_page
‡http://mappings.dbpedia.org/index.php/Mapping_en
§http://umbel.org/
¶https://www.w3.org/TR/skos-reference/
‖http://wiki.dbpedia.org/services-resources/ontology
**http://wiki.dbpedia.org/develop/datasets
††https://en.wikipedia.org/wiki

extracts infobox information and category information from Wikipedia. Category pages are lists of articles that belong to a specific category containing the list of concepts, relations and entities. Each Wikipedia article is a single web page and describes a single topic. The Wikipedia pages are manually assigned to one or more categories. Wikipedia categories maintain a hierarchy, but they are often not useful for ontological purposes. The categorizations of Wikipedia pages and their link structure are available as SQL tables*, thus can be exploited without parsing the actual Wikipedia articles. On the contrary, WordNet provides a clean and carefully assembled hierarchy of thousands of concepts. YAGO links these two sources with the near perfect accuracy of 95% Suchanek et al. (2007). This accuracy was measured using weighted average Wilson score Galárraga et al. (2013) and was evaluated manually.

YAGO is based on a data model of entities and binary relations and introduces a slight extension of RDFS Suchanek et al. (2007). The YAGO data model is able to represent entities, facts, relations between facts, and properties of relations, but it is at the same time simple and decidable. All objects and URLs are represented as entities in the YAGO model. Also numbers, dates, strings and other literals are represented as entities. Classes are entities as well and are arranged in a taxonomic hierarchy expressed by subclass relations. A fact is a triple consisting of an entity name, a relation name and another entity name (e.g., *AlbertEinstein BornInYear 1879*). Each fact has an identifier (e.g., *#1 AlbertEinstein HasWonPrize NoblePrize*) and the YAGO data model considers each identifier as an entity like in RDFS. Entities that are neither facts nor relations are referred to as common entities. Common entities that are not classes are called individuals. The YAGO model also offers a solution to represent n-ary relations.

YAGO2 Hoffart et al. (2011, 2013) is an extension of YAGO which focuses mainly on temporal, spatial and textual dimensions. The system that builds YAGO2 extends the fact extraction approach of YAGO with the help of Wikipedia, WordNet and GeoNames Vatant and Wick.

YAGO3 Mahdisoltani et al. (2014) is a further extension and aims to construct a single coherent knowledge base from Wikipedia in ten different languages. It maps multilingual infobox attributes to recognise relations, merges equivalent entities to canonical entities by using Wikidata Vrandečić (2012); Vrandečić and Krötzsch (2014), cleans the data, and finally arranges all the entities in a single taxonomy. YAGO3 gains one million new entities and seven million new facts over YAGO. YAGO3 uses Wikidata to avoid duplication of entities, because Wikidata maintains its own repository of entities and category identifiers. Wikidata maps these entities to articles in Wikipedia in different languages.

Information Extraction

YAGO extracts facts from infoboxes of Wikipedia. The infoboxes are grouped into templates, which often carry the name of the class of the article entity. YAGO uses an infobox extractor, a term extractor and an attribute mapper for this purpose. It also performs several checks (e.g., type checking and functional clash checking) on its facts. YAGO is designed to be extendable. Since Wikipedia has more individuals than WordNet, the individuals are taken from Wikipedia. Each Wikipedia page is a candidate to become an individual in YAGO. To assign each individual to its class, YAGO exploits the category system of Wikipedia. There are different types of categories (e.g., conceptual categories, administrative categories, and thematic categories). A conceptual category generally indicates the class of an individual. YAGO relies on a form of shallow linguistic parsing using the *Noun Group Parser* Suchanek et al. (2006) to identify the conceptual categories from the thematic categories. If the head of the category name is in plural form, then the category is most likely a conceptual category. The *Pling-Stemmer* Suchanek et al. (2006) is used to identify and stem plural words.

YAGO relies on about 100 manually defined relations which were specified in the source code. YAGO2 re-engineered the source code to make it extensible and re-defined rules in a declarative way within a separate text file. YAGO2 defined factual rules, implication rules, replacement rules and extraction rules. For the temporal dimension, YAGO2 considers four major types of entities: people, groups, artifacts and events. For the geo-spatial dimension, the entities that have permanent spatial extent on earth such as mountain, river, lake, and cities are considered. YAGO2 harvests geo-entities from GeoNames and Wikipedia. YAGO2 has temporal and geo-spatial dimensions for facts too.

In YAGO3, information is extracted from multilingual infoboxes and the information extraction process is similar to the process of English Wikipedia pages. The information extraction process uses Wilson score measures Mahdisoltani et al. (2014) to align relations from multilingual Wikipedia pages.

Taxonomy Generation

The taxonomy of YAGO comes usually from the categories of Wikipedia. YAGO uses a sequence of extractors (e.g., category extractor and subsequent extractor) to complete the taxonomy generation process.

YAGO takes only the leaf categories from Wikipedia and uses WordNet to establish the hierarchy of classes. Each set of synonyms (synset) of WordNet becomes a class in YAGO. YAGO's integration algorithm first determines the head compound, the pre- and post-modifiers of the category name; then stems the head compound of the category name into its singular form and then checks whether there is a WordNet synset or not for the pre-modifier and the head compound. For example, for the Wikipedia category *Portuguese footballers* the algorithm finds the head *footballers* and stems it to the singular form *footballer*. The mapping is non-trivial since one word may refer to different synsets in WordNet. YAGO solves this issue by taking the frequency information relating to each word from WordNet. YAGO obtains a complete hierarchy of classes by getting the upper class from WordNet and the leaves from Wikipedia. In English Wikipedia, the categories of articles are identified by the word category but other languages have different keywords. YAGO3 uses Wikidata to identify categories for this purpose.

*https://www.mediawiki.org/wiki/MediaWiki

Characteristics

YAGO stores entities and relations in canonicalised form and is a multilingual knowledge base. YAGO uses WordNet to deal with unreliable data. Generally, the categories in Wikipedia are arranged as a directed acyclic graph, which in turn provides a hierarchy of categories. This hierarchy just gives the thematic view of the Wikipedia pages and is of little use from an ontological point of view. Hence, the system that generates YAGO takes only the leaf categories of Wikipedia and ignores all other categories. Afterwards, WordNet is used to establish the hierarchy of classes. In its latest version, YAGO covers ten languages. YAGO uses the RDF data model to store its data. The entire YAGO knowledge base is available online* for download. YAGO also provides a SPARQL endpoint† like DBpedia to query data online.

YAGO has been used in many tasks such as data cleaning Chu et al. (2015), semantic searching Bast et al. (2012), entity resolution Pujara et al. (2013), question answering Ferrucci et al. (2010), for aligning other knowledge bases such as multilingual DBpedia Mahdisoltani et al. (2014), understanding and mining unstructured data Huet et al. (2013), and digital humanities Rebele et al. (2017).

NELL

The NELL (Never Ending Language Learning) project Carlson et al. (2010); Mitchell et al. (2015) has been initiated with three main objectives. Firstly, it is a case study in lifelong or never ending learning. Secondly, it is an attempt to advance the state of the art of natural language processing. Thirdly, it is an attempt to develop the world's largest knowledge base that will collect factual information from the web that is useful for many intelligent applications.

NELL was built as an autonomous agent which runs 24 hours per day, 7 days a week, in short forever to better reflect the more ambitious and encompassing type of learning performed by humans. NELL constantly performs two tasks: reading and learning. Reading includes extracting information from a web corpus (Clueweb data set, Callan et al. (2009)) to populate a knowledge base of structured concepts and relations. The learning goal is to read better each day than the previous day. In the following we focus on NELL's knowledge base construction process.

NELL employs a number of knowledge extraction methods, semi-supervised learning methods, and a flexible knowledge representation that allows the integration of the outputs from all the methods. Initially, NELL Carlson et al. (2010) acquired two types of knowledge: knowledge about noun phrases (e.g., which noun phrase refers to which semantic category, such as *countries*, and *sports*) and knowledge about relations among the noun phrases, that is which noun phrase pairs satisfy which specified semantic relation (e.g., *hasOfficeIn(organization, l ocation)*). NELL uses a variety of ways to learn to acquire this knowledge. It learns free form text patterns to extract knowledge from the web; it learns to extract this knowledge from semi-structured web data such as tables or lists; it learns morphological regularities of instances of categories, and probabilistic Horn clauses that enable NELL to infer new instances of relations from learned relation instances.

The original knowledge base of NELL defines an ontology which is a collection of predicates defining c ategories , relations and a handful of seed examples for each predicate‡. Category and relation instances that are added to the knowledge base are partitioned into candidate facts and beliefs. Each belief is a triple (*entity, relation, value*). The subsystem components can look through the knowledge base and confer other external resources (e.g., the web) and then propose new candidate facts. Components supply a probability for each proposed candidate and a summary of the source evidence supporting it. NELL's knowledge integrator (KI) examines the recommended candidate facts and upgrades the strongly supported ones to belief status. NELL works in an iterative way. During each iteration, each subsystem component runs to completion given the current knowledge base and then the KI makes decisions on which newly proposed candidate facts to consider. One can view this approach as an approximation of an Expectation Maximization (EM) algorithm in which the E-step involves iteratively estimating the truth values for a very large set of virtual candidate beliefs in the shared knowledge base to propose updates and the M-step involves retraining the various subsystem components, employing module-specific l earning a lgorithms w ith the refined k nowledge b ase. K I b oth r ecords t hese individual recommendations made in the E-step and makes a final decision about the proposed updates based on the confidence value. As this type of iterative learning system may suffer if labeling errors accumulate, NELL allows humans to interact with the system for 10-15 minutes per day to mitigate this issue.

The NELL system can be seen as a multitask learning system in which different functions are trained together to improve the learning accuracy. In its preliminary version Carlson et al. (2010), NELL contained four subsystem components: Coupled Pattern Learner (CPL), Coupled SEAL (CSEAL), Coupled Morphological Classifier (CMC) and Rule Learner (RL). In its current version Mitchell et al. (2015), more components such as OpenEval Samadi et al. (2013), Neil Chen et al. (2013), Path Ranking Algorithm (PRA) Lao et al. (2011) and OntExt Mohamed et al. (2011) have been added. OpenEval actively searches the web to validate the information, Neil discovers commonsense relationships and similar label instances of a given visual category, PRA infers new beliefs from the existing knowledge base and OntExt extends the ontology.

Information Extraction

For information extraction, NELL uses CPL, CSEAL and RL. CPL is a free-text extractor which learns and uses contextual patterns like *"President of X"* and *"X works for Y"* to extract instances of categories and relations. NELL uses co-occurrence statistics between noun phrases and contextual patterns to learn extraction patterns for each

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

†https://linkeddata1.calcul.u-psud.fr/sparql

‡http://rtw.ml.cmu.edu/rtw/

predicate of interest and uses those patterns to find additional instances of each predicate.

On the other hand, CSEAL is a semi-structured extractor which searches the web with a set of beliefs from each category or relation; then it mines the lists and tables to extract novel instances of the corresponding predicate. It uses mutual exclusion relationships to provide negative examples, which are used to filter out overly general lists and tables.

NELL uses a first-order relational learning algorithm similar to FOIL Quinlan and Cameron-Jones (1993) to infer new relation instances from other relation instances that are already existing in the knowledge base.

Taxonomy Generation

NELL uses CMC to classify noun phrases based on various morphological features. CMC examines candidate facts proposed by other components and classifies new beliefs. NELL invents new relational predicates using the OntExt system. OntExt works in three steps: 1. extracting sentences mentioning known instances of both categories in a category pair; 2. building a context (e.g., the relations between the instances) by using a context co-occurrence matrix from the extracted sentences, then clustering the related context together Mohamed et al. (2011); Barchi and Hruschka (2014); and finally, 3. employing a trained classifier and a final stage manual filtering process to select the appropriate new relation to be added to the ontology.

Characteristics

NELL uses independent subsystem components where an error in one subsystem component does not affect the other components. NELL learns multiple types of inter-related knowledge and use coupled semi-supervised learning strategies to leverage constraints between predicates being learned. NELL uses a uniform knowledge representation to catch possible facts and the promoted beliefs of all types. It uses related inference and learning mechanisms that can operate on this shared representation.

NELL employs a semi-supervised bootstrap learning method, which begins with a small set of labeled data, trains a model and then uses that model to label more data. Bootstrap learning approaches can often suffer from "semantic drift" (where labeling errors in the learning process can accumulate). Semantic drift is the evolution of word usage over time to the point that the orginal meaning is often not accessible anymore. For example, consider the word *"awful"*; it's original meaning was *"inspiring wonder (or fear)"* and was originally used as a short form of *"full of awe"*, in its contemporary usage the word usually has a negative meaning. NELL enforces constrains in the learning process to mitigate this issue. datasets and supplementary files regarding the NELL project are freely available online*. NELL is not multilingual and uses canonicalization only to store relations. It uses RDBMS to store its datasets and has no spatio-temporal information. NELL was mainly designed as an autonomous agent† that continuously learns by extracting structured information from unstructured web pages Mitchell et al. (2018) and was also used in question answering Fader et al. (2014).

Probase

Probase Wu et al. (2012) is a multipurpose, probabilistic knowledge base which was built automatically from a collection of 1.6 billion web pages. It is a project of Microsoft Research Asia with the aim of making machines aware of the mental world of human beings. Since September 2016, Probase is also known as Microsoft Concept Graph‡. Probase's taxonomy is unique in the following aspects: Firstly, it consists of an iterative learning algorithm for extracting *isA* pairs (e.g., *cat isA animal, tree isA plant*) and a taxonomy construction algorithm to make the connection among these pairs for building a concept hierarchy. In 2012, the taxonomy had a reported precision of 92.8% and had the largest scale reported among the existing automatic approaches Wu et al. (2012). Secondly, it is one of the first general purpose taxonomies that takes a probabilistic approach to model the knowledge it holds. Each fact and relation in the knowledge base is associated with a probability. Thirdly, it was the largest general-purpose knowledge base that was constructed automatically from HTML text when it was first published.

The knowledge acquisition process in Probase is done in two stages: 1. information extraction and 2. data cleansing and integration.

Information Extraction

The information extraction process of Probase is iterative and includes both syntactic and semantic iterations. Unlike other information extraction methods, a fixed set of patterns (Hearst patterns Hearst (1992)) is used for each syntactic iteration. For semantic iterations, Probase performs iterative learning at the knowledge level. For example, given the sentence *"athletes other than footballers such as cricketers"*, Probase understands that there are two possibilities here: either *"cricketer isA footballer"* or *"cricketer isA athlete"*. In its first iteration Probase does not have knowledge about athlete, footballer and cricketer, but during its second iteration, the algorithm knows that the frequency of *"cricketer isA athlete"* is higher than the one of *"cricketer isA footballer"*. By combining both methods the algorithm tries to extract information at the semantic level because only applying pattern-based information extraction often prevents deep knowledge acquisition. Syntactic patterns have limited extraction capacity and high quality patterns are rare as well. Sometimes recall is also sacrificed for precision because of the ambiguity of natural language and the low quality of syntactic patterns. Probase searches for *isA* pairs in the text and uses the existing knowledge base to identify the valid pairs.

The Probase information extraction framework focuses on understanding because in many cases semantics is required in addition to syntax for correct extraction. Probase uses the acquired knowledge from each iteration to obtain new knowledge. We can define this information extraction process as an iterative learning task. Let us start from a knowledge base that is initially empty. During each iteration

*http://rtw.ml.cmu.edu/rtw/resources

†http://rtw.ml.cmu.edu/rtw/overview

‡https://concept.research.microsoft.com/Home/Introduction

of extracting *isA* pairs, the knowledge base is updated with the new pair which helps Probase to idenify more *isA* pairs in the subsequent iterations.

The *isA* pair extraction process repeatedly scans the set of sentences extracted from the web. It uses the *SyntacticExtraction* procedure to extract both super-concepts X_s and sub-concepts Y_s. If more than one super-concept candidate exists, it executes the *SuperConceptDetection* function which uses a probabilistic approach to find the most relevant super-concept from the candidate list. For example, consider the sentence *"Countries other than Japan such as China"*. Here the candidates for super-concepts are *"Countries"* and *"Japan"*. Suppose $X_s = \{x_1, ..., x_m\}$; then the function computes the likelihood p($x_k \mid Y_s$) for $x_k \in X_s$. If x_1 and x_2 have the largest likelihood and p($x_1 \mid Y_s$) $\geq$ p($x_2 \mid Y_s$), then it calculates the ratio likelihood r(x_1, x_2) and picks x_1 if the ratio is above a certain threshold.

On the other hand, the procedure *SubConceptDetection* cleans the unwanted sub-concepts in Y_s. Finally, the new pairs are added to the knowledge base. The sub-concept detection function finds the sub-concept from Y_s. Probase does this by extracting features from the sentences. The closer a candidate sub-concept is to the pattern keyword, the higher is the possibility that it is a valid sub-concept. Another observation is that if it is certain that a candidate sub-concept at the k-th position from the pattern keywords (e.g., *"such as"*) is valid, then most likely all the candidate sub-concepts from position 1 to position $k-1$ are also valid. For example, consider the sentence *"Companies such as IBM, Nokia, Procter & Gamble"*. Here the candidates for sub-concepts are *"IBM", "Nokia"* and *"Procter & Gamble"*. In this case, the algorithm finds the largest k such that the likelihood p($y_k \mid x$) is above a certain threshold and y_k is a candidate sub-concept at the k-th position from the pattern keyword. If no y_k has been found, then it assumes $k = 1$. If there exist two candidate sub-concepts in Probase for detection at a certain position *j*, and $y_j \in \{c_1, c_2\}$ where c_1 and c_2 are semantic concepts, then it calculates the ratio likelihood r(c_1, c_2) and picks c_1 over c_2, if the ratio is above a certain threshold.

Taxonomy Generation

The information extraction process in Probase generates a number of *isA* pairs and each pair represents an edge in the taxonomy. Probase models the taxonomy as a directed acyclic graph. A node in the taxonomy is either a concept node (e.g., *Person*) or an instance (e.g., *John*). A concept contains a set of instances and a set of sub-concepts.

For taxonomy construction, Probase identifies some properties of the *isA* pairs and based on those properties introduces two operators that merge nodes belonging to the same sense. For example, consider x^i where *x* represents the node while *i* represents the sense. So, the two nodes x^i and x^j will be equivalent iff *i=j*. Probase reveals a number of properties for taxonomy construction through Hearst patterns. These properties are:

- Let s=$\{(x_1, y_1),...,(x_n, y_n)\}$ be the *isA* pairs derived from a sentence. Here x denotes the super-concept and y denotes the sub-concept. Then, all the x's in s have a unique sense. Thus Probase joins such *isA* pairs derived from a single sentence based on the super-concept node x. This type of taxonomy is called a local taxonomy. Here is an example sentence:

  "Footballers such as strikers, midfielders and defenders ...".
- Let $\{(x^i, y_1), ..., (x^i, y_n)\}$ denote a set of pairs from one sentence and $\{(x^j, z_1), ..., (x^j, z_n)\}$ denote a set of pairs from another sentence. If $\{y_1, ..., y_n\}$ and $\{z_1, ..., z_n\}$ are similar then there is a possibility that x^i and x^j are equivalent and in such a case the construction process performs a horizontal merge (when two concepts are equivalent). Here is an example:

  "Footballers such as strikers, midfielders and defenders ..." and *"Footballers such as strikers, midfielders, wingers ...".*
- Let $\{(x^i, y), (x^i, u_1), ..., (x^i, u_m)\}$ denote pairs obtained from a sentence and $\{(y^k, v_1), ..., (y^k, v_n)\}$ denote pairs obtained from another sentence. If $\{u_1, ..., u_m\}$ and $\{v_1, ..., v_n\}$ are similar, then it is highly likely that y and y^k have the same sense, and then Probase performs a vertical merge operation (when one concept is a sub-concept of another). Here is an example:

  "Athlets such as footballers, cricketers, gymnasts and swimmers ..." and *"Footballers such as strikers, midfielders and defenders ...".*

So the taxonomy construction process for Probase can be divided into three stages: 1. local taxonomy construction, 2. horizontal merging, and 3. vertical merging. The procedure performs horizontal merging before vertical merging but this order is not mandatory. Any sequence of merge operations will give the same result.

Characteristics

Information in 271 languages is available in the Probase knowledge base which is not canonicalised. The idea behind the Probase project is to live with unstructured data and make the utmost possible usage of it. Probase uses a probabilistic framework to manage knowledge. The framework relies on two components: 1. plausibility, the joint probability of an *isA* pair and 2. typicality, the conditional probability between a concept and its instance. Every claim in Probase has a probability of being true. Typicality can be in two directions: 1. instantiation and 2. abstraction. Instantiation indicates the probability of an instance given a concept. On the other hand, abstraction describes the probability of a concept given an instance.

Low quality syntactic patterns obtained from bootstrapping often cause semantic drift. To deal with the semantic drift problem, Probase created sophisticated discriminators by identifying drifting points to remove syntactic patterns of low quality Li et al. (2014). Probase also does web scale taxonomy cleaning Lee et al. (2011) and sparse information extraction using semantic contexts Li et al. (2013) for data cleaning purpose. Facts stored in Probase can be downloaded from the official website of Microsoft Concept Graph*. The

*https://concept.research.microsoft.com/Home/Download

RDFS format has been used in Probase for storing information and it does not store spatio-temporal information.

Applications of Probase include semantic search Wang et al. (2016, 2012b, 2010), understanding web tables Wang et al. (2012a), question answering Fader et al. (2014), short text understanding Wang (2016); Song et al. (2011) and classification Li et al. (2018), topic modelling Tang et al. (2018), open directory based text classification Jun et al. (2018), and learning entity and concept representation Shalaby et al. (2018).

BabelNet

Similar to the YAGO project, the BabelNet project Navigli and Ponzetto (2010, 2012) was motivated by building a wide-coverage multilingual knowledge base. BabelNet was built automatically following a methodology that integrates lexicographic knowledge from WordNet and encyclopedic knowledge from Wikipedia. The integration is performed via an automatic mapping and by filling t he l exical gaps in resource-poor languages with the help of machine translation. BabelNet can be defined a s a n encyclopedic dictionary that provides concepts and instances lexicalized in different languages and is connected with a large amount of semantic relations.

Building such a linguistic resource manually is a very difficult t ask a nd r equires d ozens o f y ears t o b uild from scratch for each new language. Also this type of linguistic resources needs to be interlinked with other resources across languages and domains. EuroWordNet Vossen (1998), MultiWordNet Pianta and Bentivogli (2002), Multilingual Central Repository Atserias et al. (2004) and BalkaNet Stamou et al. (2002) are examples of manual efforts towards building multilingual knowledge bases. These manually built resources often have poorer coverage for non-English languages compared to others which in turn make people bias to conduct research in resource-rich languages.

One major feature of Wikipedia is having the richness of explicit and implicit semantic knowledge mostly about named entities. However, one major limitation of Wikipedia is the lack of lexicographic senses of a given lemma which can be provided by WordNet. The methodology of BabelNet consists of three major steps: 1. combining WordNet and Wikipedia, 2. harvesting multilingual lexicalizations of available concepts by using both human generated translations provided by Wikipedia and a machine translation system to translate occurrences of concepts in sense tagged corpora, and 3. establishing relations between the Babel synsets.

Information Extraction

BabelNet aims to provide an encyclopedic dictionary by harvesting concepts and relations from WordNet and Wikipedia. BabelNet encodes knowledge as a labeled directed graph $G = (V, E)$ where V is the set of nodes (e.g., concepts such as *"play"* and named entities such as *"Shakespeare"*) and $E \subseteq V \times R \times V$ is the set of edges connecting pairs of concepts (e.g., *play is-a dramatic composition*). Each edge is labeled with a semantic relation from R (e.g., *is-a, part-of, instance-of, ...,* ϵ) where ϵ denotes an unspecified semantic relation. Each node $v \in V$ contains a set of lexicalizations of concepts for different languages (e.g., $play_{EN}$, $dramma_{IT}$, $obra_{ES}$). Such multilingually lexicalized concepts are called *Babel synsets.* In order to construct a BabelNet graph, the following information is collected at different stages: 1. all available word senses as concepts and all the lexical and semantic pointers between synsets as relations from WordNet and 2. all encyclopedic entries (e.g., Wikipages) as concepts and semantically unspecified hyperlinked texts.

Since the information found in Wikipedia and WordNet can overlap both in terms of concepts and relations, BabelNet intersects these two knowledge sources in order to provide a unified resource. To enable multilinguality, BabelNet collects the lexical realizations of the available concepts in different languages. Finally, Babel synsets are connected by establishing semantic relations between them.

Taxonomy Generation

The first step towards creating BabelNet is to establish links between WordNet senses and Wikipages. The lemma that corresponds to a particular Wikipage is given by its title (*disaster* for *Disaster*) or the main token of the sense-labeled title (*play* for *Play (activity)*). The taxonomy mapping method in BabelNet works as follows:

- It uses a mapping algorithm that relies on the properties of WordNet and Wikipedia, namely monosemous senses and re-directions; and finds the WordNet sense that increases the probability of the sense providing an adequate corresponding concept for a given Wikipage.
- It considers the mapping process as a disambiguation problem and correlates a disambiguation context (e.g., determines the actual sense of an ambiguous word in a given context) with both WordNet senses and Wikipages. BabelNet uses the same technique as adopted for word sense disambiguation Navigli (2009) to solve this problem.
- Lastly, it leverages two strategies to estimate the conditional probability of a WordNet sense for each Wikipage considering the disambiguation context. These strategies are: 1. bag-of-words Navigli and Ponzetto (2010) and 2. graph-based methods Navigli and Lapata (2010).

Initially, the mapping algorithm assigns each Wikipage w to ϵ. Each Wikipage that has a monosemous lemma both in Wikipedia and WordNet is mapped to its WordNet sense by the algorithm. For each Wikipage w having no previous mapping, the algorithm does the following:

- For each Wikipage d which is a redirection to w, for which a mapping was found and maps to a sense in the synset S and also contains a sense of w, then w is mapped to the matching sense in S.
- If a Wikipage w has not been mapped yet, the most probable sense of w based on the maximization of the conditional probabilities $p(s|w)$ over the senses is assigned.

A Babel synset $S \cup W$ is created after mapping a Wikipage w to a WordNet sense s. Here the sense s belongs

to the WordNet synset S and W includes the Wikipage w, the set of redirections to w, all pages linked via its inter-language links (e.g., translation of the Wikipage into other languages) and the re-directions to the inter-language links found in Wikipedia of the target language. For instance, given the mapping PLAY(THEATRE) = play^1_n, the corresponding Babel synset is *play*$_{en}$, *Bühnenwerk*$_{de}$, and *opera teatrale*$_{it}$.

BabelNet deals with two issues after the mapping is done. The first issue is, a concept might be present in one of the two resources and the second issue is, even if a concept is present in both resources, the Wikipage might not provide any translation for the language of interest. In order to resolve these issues, BabelNet uses a methodology for translating senses in the Babel synset into a missing language using SemCor Miller et al. (1993) (a corpus of more than 200,000 words annotated with WordNet senses). In order to fill the lexical gaps, BabelNet collects sense-annotated data from SemCor and Wikipedia. It then applies the Google Translator* to get the most frequent translations and adds these translations as additional lexicalizations to the knowledge base.

Characteristics

BabelNet is a multilingual knowledge base constructed from Wikipedia and WordNet. It tries to bridge the linguistic gap between the concepts by using a state of the art machine translation system (e.g., Google Translator). BabelNet uses the disambiguation context for the mapping process. The disambiguation context consists of a set of words acquired from corresponding resources for a given concept (e.g., a Wikipage or sense). These word senses are associated with the input concept by means of some semantic relations which provide evidence for a feasible connection for the mapping process. For a Wikipage, the disambiguation contexts are sense labels, links, redirections and categories. For a WordNet sense, synonyms, hypernyms/hyponyms and the glosses are used as disambiguation contexts. In its latest version 3.7, BabelNet covers 271 languages obtained from the automatic integration of ItalWordNet Roventini et al. (2000), Open Multilingual WordNet Bond and Foster (2013), OmegaWiki Meijssen (2009), Wikidata, Wonef Pradet et al. (2014), GeoNames, Wikiquote Stein (2011), VerbNet SCHULER (2005), ImageNet Deng et al. (2009), FrameNet Baker et al. (1998), and WN-Map Daudé et al. (2000) along with Wikipedia and WordNet. datasets of the BabelNet knowledge base can be found online†.

BabelNet is used to perform knowledge-rich, graph-based word sense disambiguation in both monolingual and multilingual settings Lefever and Hoste (2010); Navigli et al. (2013); Elbedweihy et al. (2013); Lesnikova et al. (2015). BabelNet is also used in content-based recommender systems de Gemmis et al. (2015); Narducci et al. (2016), cold-start music recommendation Oramas et al. (2017), cross-language recommendation Narducci et al. (2016) and plagiarism detection Franco-Salvador et al. (2016).

Knowledge Vault

Knowledge Vault (KV, Dong et al. (2014)) is a web scale probabilistic knowledge base that combines information extracted from web content (such as text documents, tabular data, human annotations, and page structure) with prior knowledge derived from existing knowledge repositories. KV is significantly bigger than YAGO, NELL, Probase, and KnowItAll and contains 271 million facts. KV features a probabilistic inference system that computes calibrated probabilities of fact correctness. KV stores information in RDF triple format. For example:
$\langle Barak\ Obama, /people/person/place\_of\_birth, Honolulu \rangle$

Each triple is associated with a confidence score, representing the probability of being true. Entity types and predicates come from a fixed ontology similar to other knowledge base construction systems like YAGO and NELL. KV separates facts about the world from their lexical representation which makes it a language independent structured knowledge repository. Unlike the earlier automatically constructed knowledge bases, KV combines noisy information extracted from the web with prior knowledge derived from Freebase Bollacker et al. (2008). KV is considerably larger than other comparable knowledge bases as it extracts facts from a variety of web data sources (e.g., free text, HTML DOM trees, HTML tables, and human annotations of web pages). KV has 1.6 billion triples among which 324 millions have a confidence of 0.7 or higher and 271 millions have a confidence of 0.9 or higher. KV's architecture is composed of three major components: 1. information extractors, 2. graph-based priors, and 3. a knowledge fusion system.

Information Extraction

The information extractors are the components that extract triples from a large number of web data sources. After extraction each triple is assigned a confidence score by the extractor to represent the uncertainty about the relation and its corresponding arguments. On the other hand, the graph-based prior system of KV learns the probability of each possible triple based on the triple stored in the existing knowledge base. The knowledge fusion system computes the probability of a triple being true based on the agreement between different extractors and priors. Abstractly, KV's knowledge base construction problem can be viewed as constructing a weighted labeled graph, which can be a very sparse $E \times P \times E$ 3d-matrix G, where E represents the number of entities and P represents the number of predicates. $G(s, p, o) = 1$ means that there is a link of type p from s to o, and $G(s, p, o) = 0$ means there exists no link. KV computes the probability $Pr(G(s, p, o) = 1 \mid .)$ for candidate triples (s, p, o) where the probability is conditional on different sources of information. While extracting information, KV conditions on the text features about the triple and while using graph based priors, KV conditions on the known relations in the Freebase graph. Finally, in the knowledge fusion step, KV conditions on both text extraction and prior relations.

For information extraction from text documents, KV runs a suite of standard natural language processing tools over each document. These tools perform named entity

*https://translate.google.com.au/

†http://babelnet.org/download

recognition, part-of-speech tagging, dependency parsing, co-reference resolution and entity linkage (which maps the mentions of proper nouns and their co-references to the corresponding entities in the knowledge base). Next, KV trains the relation extractors using distant supervision Mintz et al. (2009). To be more specific, for each predicate of interest, the information extraction process extracts a seed set of entity pairs containing the predicate, then it finds example sentences containing these entity pairs and extracts patterns from them. For example, if the predicate is *played_for*, the pairs could be *(CristianoRonaldo, RealMadrid)* and *(LionelMessi, Barcelona)*. KV then searches for example sentences that contain pairs such as *(CristianoRonaldo, RealMadrid)* and *(LionelMessi, Barcelona)* in the text documents and extracts patterns from these sentences. In a bootstrapping phase, KV looks for more example sentences that contain these patterns between entity pairs of the correct type and then uses the local closed world assumption available via Freebase to derive labels for the resulting set of extractions. Once the training set is labeled, KV fits a binary classifier for each predicate.

KV extract triples from HTML DOM* trees in a similar way as from text. Fact extraction from HTML tables follows two steps. Firstly, KV performs named entity linkage just like in the text case. Secondly, KV attempts to identify the relation that is expressed in each column header of the table by looking at the entities in the corresponding column. After that it tries to find appropriate predicates by matching these entities to Freebase. For this purpose, KV follows the standard schema matching method Venetis et al. (2011) which is a system that tries to recover the semantics of tables by enriching the table with additional annotations. KV uses manual mappings from schema.org† to Freebase for 14 different predicates (mostly related to people) for fact extraction from human annotated pages. Annotations related to other entities are currently not stored in the KV knowledge base.

Taxonomy Generation

Generally, facts that have been extracted from the web are unreliable and a smart way to solve this issue is to use prior knowledge derived from existing knowledge bases. KV exploits existing triples in Freebase to assign probabilities to possible triples even if there is no corresponding evidence for these facts on the web. This problem is similar to link prediction Huang (2010) in a graph where a number of existing edges are observed and other edges which are likely to exist are predicted. KV uses two approaches to solve this issue: Path Ranking Algorithm (PRA) Lao et al. (2011) and Neural Network Model Mikolov et al. (2013).

PRA is quite similar to distant supervision. It starts with a set of entities that are connected by some predicates. PRA then does a random walk on the graph by starting at all the subject nodes. Paths that reach the object nodes are considered successful. For example: Entity pairs having relations such as *marriedTo(X,Y)* often also have a path of the form: *parentOf(X,Z)* and *parentOf(Y,Z)*. Since if two people share a common child, then they are likely to be married. The paths learned by PRA can be interpreted by rules. Since multiple rules or paths might apply for any given pair of entities, they can be combined by fitting a binary classifier. In PRA the features are the probabilities of reaching a target node from a source node following different types of paths and the labels are derived using the existing prior. An alternative approach for building the prior model is to view the link prediction problem as a matrix completion problem. KV uses a neural network model for this purpose.

Characteristics

In contrast to previous knowledge base architectures, KV fuses together multiple extraction sources with prior knowledge derived from an existing knowledge base. The facts in KV have associated probabilities but KV does not support temporal and spatial facts at present. KV uses RDF triples to represent its knowledge. Graph-based priors are used to deal with unreliable data extracted from the web.

KV fuses extractors and priors which boosts performance. For fusing the extractors, KV uses a feature vector for each extracted triple and then applies a binary classifier to compute the probability. For the sake of simplicity and speed, KV fits a separate classifier for each predicate. datasets of KV are not available online for use.

KV can be used in artificial intelligence applications such as intelligent web services, augmented reality, and virtual assistant use cases‡.

Other Prominent Universal Knowledge Bases

Apart from the automatically constructed knowledge bases discussed in the previous sections, there exist a number of knowledge bases that have mainly been built manually. Freebase and Wikidata are the most prominent manually built collaborative knowledge bases.

Freebase

Freebase Bollacker et al. (2008, 2007) is a domain independent knowledge base containing general human knowledge and has been developed by collaborative data communities like Wikipedia. Freebase merges the diversity of collaborative efforts with the scalability of structured databases to build a stable and practical platform. Knowledge stored in Freebase has been partially collected from Wikipedia and MusicBrainz Swartz (2002).

Freebase is composed of a number of components. The key components are: an expandable tuple store that has some built-in features for query planning and optimization capabilities, an application programming interface which can process both read and write requests using a metaweb query language (MQL Flanagan (2007)), a collaborative typing system, a substantial diverse data set, and a web user interface which is easy to use. The data set is available via a creative common licence. Information in Freebase is stored based on the notion of objects, facts, types and properties. Each object contains an object id and has one or more types. Freebase uses properties from the object types to provide facts Pellissier Tanon et al. (2016).

*https://www.w3.org/TR/DOM-Level-2-Core/introduction.html

†http://schema.org/

‡https://www.engadget.com/2014/08/21/google-knowledge-vault/

The last dump of the Freebase data set can be found on the web*. Freebase was used in building Google's Knowledge Graph.

Wikidata

Wikidata Vrandečić (2012); Erxleben et al. (2014) is a multilingual, community-based knowledge base that uses crowdsourcing Estellés-Arolas and González-Ladrón-De-Guevara (2012) for data acqusition and was introduced by the Wikimedia organization†. The aim of this project is to provide structured, factual information to Wikipedia. Before the introduction of Wikidata, the same information often used to appear in different articles in various languages of Wikipedia with inconsistent values. Wikidata solves this problem by providing clean and consistent data to all the Wikimedia projects in a single location Vrandečić (2012).

Each fact in Wikidata consists of a subject, property and value triple Piscopo et al. (2017). Qualifiers such as the time of validity for a fact are used to provide contextual information. Wikidata also stores the source information (e.g., reference) for each fact. For example to store the population of a particular city, Wikidata also stores the source that provides the population information for that city. As the data acquisition process of Wikidata is crowd-sourced, different sources often provide conflicting data. Wikidata has a mechanism to store and manage such conflicting data by allowing contributors to mark a statement as "preferred". datasets of Wikidata can be accessed through web services in JSON, XML or RDF format. The Wikidata datasets are available online‡ for download.

Freebase was developed by Metaweb Technologies in 2007 and was acquired by Google in 2010. In 2014, these two knowledge bases have been merged together Pellissier Tanon et al. (2016) and now Wikidata contains all the information of Freebase. In 2016, the Freebase project was shut down by Google.

Discussion

Earlier approaches for constructing knowledge bases heavily relied on manual work as well as integration of existing structured knowledge sources. Recently, the spontaneous contributions of humans has declined, therefore the growth of knowledge sources (e.g., Wikipedia) has plateaued Suh et al. (2009). High knowledge base construction time and assembly cost were also a matter of concern. Therefore, new approaches were necessary that do better scale up the knowledge base construction process. This is the motivation behind the creation of several automatically constructed knowledge bases. In this survey we discussed the architectures of different automatically constructed universal knowledge bases. Each of them has a different architecture in terms of information extraction, taxonomy generation, knowledge representation, spatio-temporal dimension, and multilinguality.

Based on our survey, we compare the discussed knowledge bases with respect to the following features:

Concepts, Relation Types and Facts: As we can see in Table 1, BabelNet has the largest number of concepts and facts compared to other knowledge bases and uses a state of the art machine translation system for filling the language gap. Using a machine translation approach to fill the language gap, is BabelNet's unique feature. Among the surveyed knowledge bases, Knowledge Vault has the highest number of relation types.

Canonicalization: DBpedia, YAGO, BabelNet and Knowledge Vault have datasets (both entities and relations) in canonicalised form whereas NELL only has relations in canonicalised form Galárraga et al. (2014).

External Ontology Usage: All the knowledge bases, except Probase and KnowItAll, use an existing ontology for the knowledge base construction process. For example, DBpedia uses DBpedia Ontology; YAGO uses WordNet, GeoNames and Wikidata; Knowledge Vault uses Freebase; NELL uses an initial ontology defining some seed categories and relations; and finally, B abelNet u ses W ordNet and SemCor.

Knowledge Representation Formalism: Among the knowledge bases, YAGO, DBpedia, BabelNet and Knowledge Vault use RDFS as knowledge representation formalism. The others use simple relational database management systems for storing the knowledge.

Temporal and Spatial Dimension: YAGO is the only knowledge base that provides spatio-temporal and contextual information for the stored facts.

Multilinguality: BabelNet, DBpedia and YAGO offer language-specific v ersions o f t heir k nowledge b ases and provide these knowledge bases in 271, 125 and 10 languages, respectively.

Information Extraction Scale and Methodology: KnowItAll, Probase, NELL and Knowledge Vault use web scale information extraction, whereas DBpedia, YAGO and BabelNet use Wikipedia based information extraction. Among the knowledge bases KnowItAll, NELL, Probase and Knowledge Vault use machine learning approaches to extract information for building the knowledge base. KnowItAll uses unsupervised learning, NELL uses semi-supervised learning and Knowledge Vault employs a supervised learning approach. On the contrary, YAGO and BabelNet adopted simple iterative approaches. Only DBpedia uses a mapping-based infobox extraction process.

Application Areas: The discussed universal knowledge bases have been used in many different application areas; for details see Table 2.

Availability: datasets for all knowledge bases are available online for public use, except for KnowItAll and Knowledge Vault. DBpedia, YAGO, BabelNet and Knowledge Vault provide their datasets in the RDFS format. The RDBMS format is used in KnowItAll, Probase and for NELL datasets. SPARQL Endpoint or a Linked Data interface can be used for data querying in DBpedia and BabelNet, while YAGO provides only a SPARQL Endpoint support. DBpedia, Probase and BabelNet have APIs to retrieve their data. On the other hand, YAGO, Probase and BabelNet provide web interfaces to access their data.

Accuracy: It is not straightforward to compare the accuracy of these knowledge bases since the data domain,

*https://developers.google.com/freebase/

†https://www.wikimedia.org/

‡https://www.wikidata.org/wiki/Wikidata:Database_download

Table 1. Knowledge base comparison based on number of concepts, relation types, and facts.

Knowledge Base Systems	No. of Concepts	No. of Relation Types	No. of Facts (in million)
KnowItAll	N/A	N/A	0.05
DBpedia[a]	760	1105	13000
YAGO[b]	352,297	100	120
Probase[c]	2.7m	N/A	4.54
NELL[d]	271	306	50
BabelNet[e]	6.066m	N/A	19717
Knowledge Vault	1100	4469	271

[a] http://wiki.dbpedia.org/datasets/dbpedia-version-2016-10
[b] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/statistics/
[c] https://www.microsoft.com/en-us/research/project/probase
[d] http://rtw.ml.cmu.edu/rtw/
[e] http://babelnet.org/stats

Table 2. Knowledge base comparison based on application areas.

Knowledge Base Systems	Application Areas
KnowItAll	Searching a large body of information.
DBpedia	Question answering, entity classification, entity disambiguation, knowledge exploration, digital libraries and archives, distributed information retrieval, and visual question answering.
YAGO	Data cleaning, entity resolution, aligning knowledge bases, understanding and mining unstructured data, digital humanities, and question answering.
Probase	Semantic search, understanding web tables, short text understanding and classification, topic modelling, open directory based text classification, learning entity and concept representation, and question answering.
NELL	Building autonomous agents and question answering.
BabelNet	Word sense disambiguation, building content-based recommender systems, cross-language recommendations, and cross-language plagiarism detection.
Knowledge Vault	Augmented reality, virtual assistants, and intelligent web services.

the amount of data and the data representations are different. Still this information gives some indications about the quality of the data. YAGO has a manually evaluated accuracy of 95% with respect to Wikipedia. Only Probase comes close to YAGO's accuracy with 92.8%. Knowledge Vault (89.1%), NELL (87%) and BabelNet (84%) have an accuracy that cannot match Probase and YAGO. Among all these knowledge bases KnowItAll has the lowest accuracy with 64% Wang et al. (2012b).

Below we provide our recommendation based on using knowledge bases in different application areas with a focus on data coverage (including spatio-temporal data), multilinguality, data ambiguity, lexicographical support, and access to Wikipedia. Table 3 shows a summary of the recommendation on surveyed knowledge bases. To the best of our knowledge there is no suitable bench-marking application to compare the surveyed knowledge bases. Furthermore, not all surveyed knowledge bases are publicly available and applying a common benchmark is not possible. Therefore, we recommend the knowledge bases based on the statistics we have found in their relevant papers.

Data Coverage: If the application requires a large volume of relational data; for example, a question answering system, then we can use DBpedia, Knowledge Vault or YAGO. Though BabelNet has the highest number of concepts and facts, it would not be suitable for this kind of application, because of the lack of relational information. YAGO is the only choice, if an application requires spatio-temporal data.

Multilinguality: If the application requires language support other than English, then one has to choose among BabelNet, DBpedia or YAGO.

Data Ambiguity: Applications that aim to minimise ambiguity should use a canonicalised knowledge bases such as DBpedia, YAGO, BabelNet or Knowledge Vault.

Lexicographical Support: An application that needs lexicographical knowledge should use BabelNet, since it integrates lexicographic and encyclopedic knowledge from WordNet and Wikipedia, respectively. YAGO can be used for the same purpose but has lower data coverage than BabelNet.

Access to Wikipedia: DBpedia, YAGO and BabelNet can be used if someone wants to access information related

Table 3. Knowledge base recommendation based on data coverage, multilinguality, data ambiguity, lexicographical support, access to Wikipedia and spatio-temporal data.

Knowledge Base System	DC[a]	ML[b]	DA[c]	LS[d]	AW[e]	STD[f]
KnowItAll						
DBpedia	✓	✓	✓		✓	
YAGO	✓	✓	✓	✓	✓	✓
Probase						
NELL						
BabelNet		✓	✓	✓	✓	
Knowledge Vault	✓	✓				

[a] DC = Data Coverage
[b] ML = Multilinguality
[c] DA = Data Ambiguity
[d] LS = Lexicographical Support
[e] AW = Access to Wikipedia
[f] STD = Spatio-temporal Data

to Wikipedia Category. For accessing Wikipedia infobox information, DBpedia and YAGO are the only options.

Future Research

As we have seen, universal knowledge bases play an important role in real-world applications. They have been used to build virtual assistants (e.g., Siri*, Alexa†, Google Now‡, and Cortana§) and provide the foundations for question answering systems Ferrucci et al. (2010), content-based recommender systems de Gemmis et al. (2015), cross-language recommendation Narducci et al. (2016) and plagiarism detection systems Franco-Salvador et al. (2016), big data analytics Suchanek and Weikum (2014), and other natural language based applications.

An important research challenge in this context is multi-modal information extraction Logan IV et al. (2017); Lonij et al. (2017) since a lot of valuable information is available in form of audio, video and images. To be useful, this information needs to be combined with textual information and should use techniques for knowledge base completion Sameer (2017).

Often knowledge bases need to be tailored for a particular application domain (such as a product graph or a goal-oriented dialogue system Ilievski (2018)), this requires novel techniques to extract and refine information from an existing knowledge base and combine this information with additional domain-specific information. Challenges to achieve this are: knowledge integration and cleaning, human-in-the-loop knowledge learning, and graph mining Xin Luna (2017).

Other research challenges are: Using neural network techniques to learn vector representations for entities and relations in a knowledge graph Srinivas and Talukdar (2017), knowledge base alignment (the discovery and refinement of candidate entity pairs among knowledge bases) Chen et al. (2019), and adding commonsense knowledge about the properties of an object to the knowledge bases Weikum et al. (2016). Another important challenge is related to how end users are supported to formulate precise queries when they interact and work with these knowledge bases. For some groups of users, a query language that is based on natural language Franconi et al. (2011); Bast et al. (2016); Hossain and Schwitter (2018) might be more suitable to search and explore these knowledge bases than a formal query language such as SPARQL¶.

Finally, another important issue is that these knowledge bases need to be maintained over their entire lifetime and their quality needs to be continuously monitored and improved Weikum et al. (2016).

Conclusion

This survey presents for the first time a comparative study on the most prominent automatically constructed universal knowledge bases with a particular focus on the information extraction and taxonomy generation process. Information extraction is the process of knowledge acquisition from various sources while taxonomy generation is the process of storing the extracted information in a lexicalized manner. We also discussed what sets these knowledge bases apart and showed how they are currently used in practical settings to help the reader to make an informed choice about their suitability for practical applications. As KnowItAll was the first knowledge base, it creates a kind of benchmark for many successive knowledge base systems. DBpedia is the most famous knowledge base that acts as a central hub in the Linked Open Data cloud. YAGO supports temporal, spatial and multilingual dimensions whereas BabelNet has the largest number of multilingual synsets. Probase provides a probability of correctness for each fact. Nell can be seen as an autonomous agent which continuously learns new facts and provides a probability measure for each of them similar to Probase. Knowledge Vault fuses information from its extractors and priors to improve data quality. The next generation of automatically constructed knowledge bases will benefit from multi-modal information extraction and machine learning techniques such as neural networks to improve their coverage and quality.

*https://www.apple.com/au/ios/siri/
†https://developer.amazon.com/alexa
‡https://www.digitaltrends.com/mobile/what-is-google-now/
§https://www.microsoft.com/en-au/windows/cortana
¶https://www.w3.org/TR/rdf-sparql-query/

References

Agichtein E and Gravano L (2000) Snowball: Extracting relations from large plain-text collections. In: *Proceedings of the Fifth ACM Conference on Digital Libraries*. ACM, pp. 85–94.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT et al. (2000) Gene ontology: tool for the unification of biology. *Nature Genetics* 25(1): 25–29.

Atserias J, Villarejo L, Rigau G, Agirre E, Carroll J, Magnini B and Vossen P (2004) The meaning multilingual central repository, 2004 .

Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R and Ives Z (2007) Dbpedia: A nucleus for a web of open data. *The Semantic Web* : 722–735.

Baker CF, Fillmore CJ and Lowe JB (1998) The berkeley framenet project. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pp. 86–90.

Banko M and Brill E (2001) Scaling to very very large corpora for natural language disambiguation. In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 26–33.

Barchi PH and Hruschka ER (2014) Never-ending ontology extension through machine reading. In: *Hybrid Intelligent Systems (HIS), 2014 14th International Conference on*. IEEE, pp. 266–272.

Bard J, Rhee SY and Ashburner M (2005) An ontology for cell types. *Genome Biology* 6(2): R21.

Barrón-Cedeno A, Rosso P, Agirre E and Labaka G (2010) Plagiarism detection across distant language pairs. In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, pp. 37–45.

Bast H, Bäurle F, Buchhold B and Haussmann E (2012) Broccoli: Semantic full-text search at your fingertips. *arXiv preprint arXiv:1207.2615, 2012* .

Bast H, Buchhold B, Haussmann E et al. (2016) Semantic search on text and knowledge bases. *Foundations and Trends® in Information Retrieval* 10(2-3): 119–271.

Becker C and Bizer C (2008) Dbpedia mobile: A location-enabled linked data browser. In: *Proceedings of the Workshop on Linked Data on the Web*. ACM.

Berners-Lee T (2006) Linked data-design issues `https://www.w3.org/DesignIssues/LinkedData` [Accessed: 2018 09 11].

Berners-Lee T, Hendler J, Lassila O et al. (2001) The semantic web. *Scientific American* 284(5): 28–37.

Bizer C, Cyganiak R and Heath T (2007) How to publish linked data on the web `http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/` [Accessed: 2018 09 11].

Bizer C, Heath T and Berners-Lee T (2008) Linked data: Principles and state of the art. In: *World Wide Web Conference*. pp. 1–40.

Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R and Hellmann S (2009) Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(3): 154–165.

Bollacker K, Cook R and Tufts P (2007) Freebase: A shared database of structured general human knowledge. In: *AAAI*, volume 7. pp. 1962–1963.

Bollacker K, Evans C, Paritosh P, Sturge T and Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, pp. 1247–1250.

Bond F and Foster R (2013) Linking and extending an open multilingual wordnet. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1352–1362.

Brill E (1992) A simple rule-based part of speech tagger. In: *Proceedings of The Workshop on Speech and Natural Language*. Association for Computational Linguistics, pp. 112–116.

Brin S (1998) Extracting patterns and relations from the world wide web. In: *International Workshop on The World Wide Web and Databases*. Springer, pp. 172–183.

Brin S and Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1-7): 107–117.

Bunescu R (2006) Using encyclopedic knowledge for named entity disambiguation. In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*. EACL, pp. 9–16.

Cabrio E, Cojan J, Aprosio AP, Magnini B, Lavelli A and Gandon F (2012) Qakis: an open domain qa system based on relational patterns. In: *International Semantic Web Conference, ISWC 2012*.

Callan J, Hoy M, Yoo C and Zhao L (2009) Clueweb09 data set.

Caracciolo C, Stellato A, Morshed A, Johannsen G, Rajbhandari S, Jaques Y and Keizer J (2013) The agrovoc linked dataset. *The Semantic Web Journal* 4(3): 341–348.

Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka Jr ER and Mitchell TM (2010) Toward an architecture for never-ending language learning. In: *AAAI*, volume 5. p. 3.

Chen L, Gu W, Tian X and Chen G (2019) Ahab: Aligning heterogeneous knowledge bases via iterative blocking. *Information Processing & Management* 56(1): 1–13.

Chen X, Shrivastava A and Gupta A (2013) Neil: Extracting visual knowledge from web data. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1409–1416.

Chu X, Morcos J, Ilyas IF, Ouzzani M, Papotti P, Tang N and Ye Y (2015) Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, pp. 1247–1261.

Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell T, Nigam K and Slattery S (1998) Learning to extract symbolic knowledge from the world wide web. In: *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of artificial intelligence*. AAAI, pp. 509–516.

Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell T, Nigam K and Slattery S (2000) Learning to construct knowledge bases from the world wide web. *Artificial Intelligence* 118(1-2): 69–113.

Cuadros M and Rigau G (2006) Quality assessment of large scale knowledge resources. In: *Proceedings of the 2006 Conference*

on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 534–541.

Cuadros M and Rigau G (2008) Knownet: building a large net of knowledge from the web. In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pp. 161–168.

Damljanovic D, Agatonovic M and Cunningham H (2011) Freya: An interactive way of querying linked data using natural language. In: *Extended Semantic Web Conference*. Springer, pp. 125–138.

Daudé J, Padro L and Rigau G (2000) Mapping wordnets using structural information. In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 504–511.

de Gemmis M, Lops P, Musto C, Narducci F and Semeraro G (2015) Semantics-aware content-based recommender systems. In: *Recommender Systems Handbook*. Springer, pp. 119–159.

Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 248–255.

Dojchinovski M and Kliegr T (2013) Entityclassifier.eu: Real-time classification of entities in text with wikipedia. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 654–658.

Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S and Zhang W (2014) Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 601–610.

Elbedweihy K, Wrigley SN, Ciravegna F and Zhang Z (2013) Using babelnet in bridging the gap between natural language queries and linked data concepts. In: *NLP-DBPEDIA@ ISWC*.

Erxleben F, Günther M, Krötzsch M, Mendez J and Vrandečić D (2014) Introducing wikidata to the linked data web. In: *International Semantic Web Conference*. Springer, pp. 50–65.

Estellés-Arolas E and González-Ladrón-De-Guevara F (2012) Towards an integrated crowdsourcing definition. *Journal of Information science* 38(2): 189–200.

Etzioni O, Cafarella M, Downey D, Kok S, Popescu AM, Shaked T, Soderland S, Weld DS and Yates A (2004) Web-scale information extraction in knowitall: (preliminary results). In: *Proceedings of the 13th International Conference on World Wide Web*. ACM, pp. 100–110.

Fader A, Zettlemoyer L and Etzioni O (2014) Open question answering over curated and extracted knowledge bases. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1156–1165.

Färber M, Ell B, Menne C and Rettinger A (2015) A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *The Semantic Web Journal* 1: 1–5.

Ferrucci D, Brown E, Chu-Carroll J, Fan J, Gondek D, Kalyanpur AA, Lally A, Murdock JW et al. (2010) Building watson: An overview of the deepqa project. *AI Magazine* 31(3): 59–79.

Flanagan D (2007) Developing metaweb-enabled web applications. *Metaweb Technologies, 2007* .

Fossati M, Kontokostas D and Lehmann J (2015) Unsupervised learning of an extensive and usable taxonomy for dbpedia. In: *Proceedings of the 11th International Conference on Semantic Systems*, SEMANTICS '15. New York, NY, USA: ACM. ISBN 978-1-4503-3462-4, pp. 177–184.

Franco-Salvador M, Rosso P and Montes-y Gómez M (2016) A systematic study of knowledge graph analysis for cross-language plagiarism detection. *Information Processing & Management* 52(4): 550–570.

Franconi E, Guagliardo P, Tessaris S and Trevisan M (2011) Quelo: an ontology-driven query interface. *Proceedings of DL 2011* 745: 488–498.

Galárraga L, Heitz G, Murphy K and Suchanek FM (2014) Canonicalizing open knowledge bases. In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, pp. 1679–1688.

Galárraga LA, Teflioudi C, Hose K and Suchanek F (2013) Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: *Proceedings of the 22nd International Conference on World Wide Web*. ACM, pp. 413–422.

Giunchiglia F, Maltese V, Farazi F and Dutta B (2010) Geowordnet: a resource for geo-spatial applications. *The Semantic Web: Research and applications* : 121–136.

Han B, Chen L and Tian X (2018) Knowledge based collection selection for distributed information retrieval. *Information Processing & Management* 54(1): 116–128.

Harabagiu SM, Moldovan DI, Pasca M, Mihalcea R, Surdeanu M, Bunescu RC, Girju R, Rus V and Morarescu P (2000) Falcon: Boosting knowledge for answer engines. In: *TREC*, volume 9. pp. 479–488.

Hearst MA (1992) Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th Conference on Computational Linguistics, Volume 2*. Association for Computational Linguistics, pp. 539–545.

Heim P, Hellmann S, Lehmann J, Lohmann S and Stegemann T (2009) Relfinder: Revealing relationships in rdf knowledge bases. *SAMT* 5887: 182–187.

Heim P, Ziegler J and Lohmann S (2008) gfacet: A browser for the web of data. In: *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW'08)*, volume 417. pp. 49–58.

Hoffart J, Suchanek FM, Berberich K, Lewis-Kelham E, De Melo G and Weikum G (2011) Yago2: exploring and querying world knowledge in time, space, context, and many languages. In: *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, pp. 229–232.

Hoffart J, Suchanek FM, Berberich K and Weikum G (2013) Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence* 194: 28–61.

Hossain BA and Schwitter R (2018) Specifying Conceptual Models Using Restricted Natrual Language. In: *The 16th Annual Workshop of The Australasian Language Technology Association (ALTA 2018), Dunedin, New Zealand*.

Huang Z (2010) Link prediction based on graph topology: The predictive value of generalized clustering coefficient. *Workshop on Link Analysis: Dynamics and Static of Large Networks, 2010* .

Huet T, Biega J and Suchanek FM (2013) Mining history with le monde. In: *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. ACM, pp. 49–54.

Ilievski V (2018) Building advanced dialogue managers for goal-oriented dialogue systems. *arXiv preprint arXiv:1806.00780, 2018* .

Islam MR, Hossain BA, Imteaj MN, Akhter S, Jogesh HS and Mostafa MB (2017) Ontranetbd: A knowledgebase for the travel network in bangladesh. In: *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. pp. 170–174. DOI:10.1109/R10-HTC.2017.8288931.

Jun SY, Aliyeva D, Lee JM and Lee S (2018) Utilizing probase in open directory project-based text classification. *arXiv preprint arXiv:1805.04992, 2018,* abs/1805.04992.

Kamel M, Aussenac-Gilles N, Buscaldi D and Comparot C (2013) A semi-automatic approach for building ontologies from a collection of structured web documents. In: *Proceedings of the Seventh International Conference on Knowledge Capture*. ACM, pp. 139–140.

Kobilarov G, Scott T, Raimond Y, Oliver S, Sizemore C, Smethurst M, Bizer C and Lee R (2009) Media meets semantic web–how the bbc uses dbpedia and linked data to make connections. In: *European Semantic Web Conference*. Springer, pp. 723–737.

Lao N, Mitchell T and Cohen WW (2011) Random walk inference and learning in a large scale knowledge base. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 529–539.

Lee T, Wang Z, Wang H and Hwang Sw (2011) Web scale taxonomy cleansing. *Proceedings of the VLDB Endowment* 4(12): 1295–1306.

Lefever E and Hoste V (2010) Semeval-2010 task 3: Cross-lingual word sense disambiguation. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pp. 15–20.

Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, Hellmann S, Morsey M, Van Kleef P, Auer S et al. (2015) Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *The Semantic Web Journal* 6(2): 167–195.

Lesnikova T, David J and Euzenat J (2015) Interlinking english and chinese rdf data using babelnet. In: *Proceedings of the 2015 ACM Symposium on Document Engineering*. ACM, pp. 39–42.

Li P, He L, Wang H, Hu X, Zhang Y, Li L and Wu X (2018) Learning from short text streams with topic drifts. *IEEE Transactions on Cybernetics* 48(9): 2697–2711. DOI:10.1109/TCYB.2017.2748598.

Li P, Wang H, Li H and Wu X (2013) Assessing sparse information extraction using semantic contexts. In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. ACM, pp. 1709–1714.

Li Z, Li H, Wang H, Yang Y, Zhang X and Zhou X (2014) Overcoming semantic drift in information extraction. In: *EDBT*. pp. 169–180.

Logan IV RL, Humeau S and Singh S (2017) Multimodal attribute extraction. *arXiv preprint arXiv:1711.11118, 2017* .

Lonij VP, Rawat A and Nicolae MI (2017) Extending knowledge bases using images. *6th Workshop on Automated Knowledge Base Construction (AKBC), 2017* .

Lopez V, Fernández M, Motta E and Stieler N (2012) Poweraqua: Supporting users in querying and exploring the semantic web. *The Semantic Web Journal* 3(3): 249–265.

Mahdisoltani F, Biega J and Suchanek F (2014) Yago3: A knowledge base from multilingual wikipedias. In: *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference.

Maltese V and Hossain BA (2012) Sam: A tool for the semi-automatic mapping and enrichment of ontologies. In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, pp. 454–463.

Matuszek C, Cabral J, Witbrock MJ and DeOliveira J (2006) An introduction to the syntax and content of cyc. In: *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*. AAAI, pp. 44–49.

Meijssen G (2009) *The Philosophy behind OmegaWiki and the Visions for the Future*. Peter Lang.

Mendes PN, Jakob M, García-Silva A and Bizer C (2011) Dbpedia spotlight: shedding light on the web of documents. In: *Proceedings of the 7th International Conference on Semantic Systems*. ACM, pp. 1–8.

Mikolov T, Chen K, Corrado G and Dean J (2013) Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781, 2013* .

Miles A and Bechhofer S (2009) SKOS simple knowledge organization system reference, 2009 `http://www.w3.org/TR/skos-reference/`.

Miller GA (1995) Wordnet: a lexical database for english. *Communications of the ACM* 38(11): 39–41.

Miller GA, Leacock C, Tengi R and Bunker RT (1993) A semantic concordance. In: *Proceedings of The Workshop on Human Language Technology*. Association for Computational Linguistics, pp. 303–308.

Mintz M, Bills S, Snow R and Jurafsky D (2009) Distant supervision for relation extraction without labeled data. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*. Association for Computational Linguistics, pp. 1003–1011.

Mitchell T, Cohen W, Hruschka E, Talukdar P, Yang B, Betteridge J, Carlson A, Dalvi B, Gardner M, Kisiel B et al. (2018) Never-ending learning. *Communications of the ACM* 61(5): 103–115.

Mitchell TM, Cohen WW, Hruschka Jr ER, Talukdar PP, Betteridge J, Carlson A, Mishra BD, Gardner M, Kisiel B, Krishnamurthy J et al. (2015) Never ending learning. In: *AAAI*. pp. 2302–2310.

Mohamed TP, Hruschka Jr ER and Mitchell TM (2011) Discovering relations between noun categories. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1447–1455.

Musto C, Narducci F, Lops P, De Gemmis M and Semeraro G (2016) Explod: A framework for explaining recommendations based on the linked open data cloud. In: *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16. New York, NY, USA: ACM. ISBN 978-1-4503-4035-9, pp. 151–154.

Narducci F, Basile P, Musto C, Lops P, Caputo A, de Gemmis M, Iaquinta L and Semeraro G (2016) Concept-based item representations for a cross-lingual content-based recommendation process. *Information Sciences* 374: 15–31.

Nastase V (2008) Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 763–772.

Navigli R (2009) Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)* 41(2): 10.

Navigli R, Faralli S, Soroa A, De Lacalle O and Agirre E (2011) Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, pp. 2317–2320.

Navigli R, Jurgens D and Vannella D (2013) Semeval-2013 task 12: Multilingual word sense disambiguation. In: *Proceedings of the Seventh International Workshop on Semantic Evaluation*, volume 2. pp. 222–231.

Navigli R and Lapata M (2010) An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(4): 678–692.

Navigli R and Ponzetto SP (2010) Babelnet: Building a very large multilingual semantic network. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 216–225.

Navigli R and Ponzetto SP (2012) Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193: 217–250.

Oramas S, Nieto O, Barbieri F and Serra X (2017) Multi-label music genre classification from audio, text, and images using deep features. *arXiv preprint arXiv:1707.04916, 2017* .

Pellissier Tanon T, Vrandečić D, Schaffert S, Steiner T and Pintscher L (2016) From freebase to wikidata: The great migration. In: *Proceedings of The 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 1419–1428.

Pianta E and Bentivogli L (2002) Christian girardi (2002) multiwordnet: developing an aligned multilingual database. In: *Proceedings of the First International Conference on Global WordNet, Mysore, India, January*. pp. 21–25.

Piscopo A, Kaffee LA, Phethean C and Simperl E (2017) Provenance information in a collaborative knowledge graph: an evaluation of wikidata external references. In: *International Semantic Web Conference*. Springer, pp. 542–558.

Ponzetto SP and Strube M (2007) Knowledge derived from wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research* 30: 181–212.

Pradet Q, De Chalendar G and Desormeaux JB (2014) Wonef, an improved, expanded and evaluated automatic french translation of wordnet. *Proceedings of the 7th Global WordNet Conference, Tartu, Estonia* : 32–39.

Pujara J, Miao H, Getoor L and Cohen W (2013) Knowledge graph identification. In: *International Semantic Web Conference*. Springer, pp. 542–557.

Quinlan J and Cameron-Jones R (1993) Foil: A midterm report. In: *Machine Learning: ECML-93*. Springer, pp. 1–20.

Rahman A and Ng V (2011) Narrowing the modeling gap: a cluster-ranking approach to coreference resolution. *Journal of Artificial Intelligence Research* 40: 469–521.

Rebele T, Nekoei A and Suchanek FM (2017) Using yago for the humanities. In: *Workshop on Humanities in the Semantic Web (WHISE) at ISWC*.

Riloff E, Jones R et al. (1999) Learning dictionaries for information extraction by multi-level bootstrapping. In: *AAAI/IAAI*. pp. 474–479.

Roventini A, Alonge A, Calzolari N, Magnini B and Bertagna F (2000) Italwordnet: a large semantic database for italian. In: *LREC*.

Samadi M, Veloso MM and Blum M (2013) Openeval: Web information query evaluation. In: *AAAI*.

Sameer S (2017) Multimodal kbs: Extraction & completion. Remarks by Sameer Singh at 6th Workshop on Automated Knowledge Base Construction (AKBC), 2017, `http://www.akbc.ws/2017/slides/sameer-singh-slides.pdf` [Accessed: 2018 9 12].

SCHULER K (2005) Verbnet: A broad-coverage, comprehensive verb lexicon. *Ph. D. Thesis, University of Pennsylvania, 2005* .

Seltzer R, Ray DS and Ray EJ (1996) *The AltaVista Revolution: How to Find Anything on the Internet*. Osborne/McGraw-Hill.

Shadbolt N, Berners-Lee T and Hall W (2006) The semantic web revisited. *Intelligent Systems* 21(3): 96–101.

Shalaby W, Zadrozny W and Jin H (2018) Beyond word embeddings: learning entity and concept representations from large scale knowledge bases. *Information Retrieval Journal, 2018* DOI:10.1007/s10791-018-9340-3.

Song Y, Wang H, Wang Z, Li H and Chen W (2011) Short text conceptualization using a probabilistic knowledgebase. In: *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Three*. AAAI, pp. 2330–2336.

Srinivas R and Talukdar PP (2017) Revisiting simple neural networks for learning representations of knowledge graphs. *arXiv preprint arXiv:1711.05401, 2017* .

Stamou S, Oflazer K, Pala K, Christoudoulakis D, Cristea D, Tufis D, Koeva S, Totkov G, Dutoit D and Grigoriadou M (2002) Balkanet: A multilingual semantic network for the balkan languages. In: *Proceedings of the International WordNet Conference, Mysore, India*. pp. 21–25.

Stein H (2011) Wikiquote. `https://en.wikiquote.org/wiki/Main_Page` [Accessed: 2018 9 12].

Suchanek FM, Ifrim G and Weikum G (2006) Leila: Learning to extract information by linguistic analysis. In: *COLING ACL*. Citeseer, p. 18.

Suchanek FM, Kasneci G and Weikum G (2007) Yago: a core of semantic knowledge. In: *Proceedings of the 16th International Conference on World Wide Web*. ACM, pp. 697–706.

Suchanek FM, Kasneci G and Weikum G (2008) Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(3): 203–217.

Suchanek FM and Weikum G (2014) Knowledge bases in the age of big data analytics. *Proceedings of the VLDB Endowment* 7(13): 1713–1714.

Suh B, Convertino G, Chi EH and Pirolli P (2009) The singularity is not near: slowing growth of wikipedia. In: *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. ACM, p. 8.

Swartz A (2002) Musicbrainz: A semantic web service. *IEEE Intelligent Systems* 17(1): 76–77.

Taboada M, Brooke J, Tofiloski M, Voll K and Stede M (2011) Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2): 267–307.

Tang YK, Mao XL, Huang H, Shi X and Wen G (2018) Conceptualization topic modeling. *Multimedia Tools and Applications* 77(3): 3455–3471. DOI:10.1007/s11042-017-5145-4.

Tori A and Šolc T (2008) Zemanta service, 2008 `http://developer.zemanta.com/` [Accessed: 2018 02 27].

Turian J (2013) Using alchemyapi for enterprise-grade text analysis. *AlchemyAPI: Denver, CO, USA, 2013* .

Turney P (2001) Mining the web for synonyms: PMI-IR versus LSA on TOEFL. *Machine Learning: ECML 2001* : 491–502.

Unger C, Bühmann L, Lehmann J, Ngonga Ngomo AC, Gerber D and Cimiano P (2012) Template-based question answering over rdf data. In: *Proceedings of the 21st International Conference on World Wide Web*. ACM, pp. 639–648.

Vatant B and Wick M (????) Geonames ontology, 2012. `http://www.geonames.org/ontology/` [Accessed: 2018 02 21].

Venetis P, Halevy A, Madhavan J, Paşca M, Shen W, Wu F, Miao G and Wu C (2011) Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment* 4(9): 528–538.

Vossen P (1998) Introduction to eurowordnet. *Computers and the Humanities* 32(2-3): 73–89.

Vrandečić D (2012) Wikidata: a new platform for collaborative data collection. In: *Proceedings of the 21st International Conference on World Wide Web*. ACM, pp. 1063–1064.

Vrandečić D and Krötzsch M (2014) Wikidata: A free collaborative knowledgebase. *Communications of the ACM* 57(10): 78–85.

Wang H (2016) Understanding short texts. *Web Technologies and Applications* : p. 01.

Wang J, Wang H, Wang Z and Zhu K (2012a) Understanding tables on the web. *Conceptual Modeling* : 141–155.

Wang P and Domeniconi C (2008) Building semantic kernels for text classification using wikipedia. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*. ACM, pp. 713–721.

Wang Y, Li H, Wang H and Zhu KQ (2010) Toward topic search on the web. Technical report, Technical report, Microsoft Research.

Wang Y, Völker J and Haase P (2006) Towards semi-automatic ontology building supported by large-scale knowledge acquisition. In: *AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*, volume 6. p. 06.

Wang Z, Huang J, Li H, Liu B, Shao B, Wang H, Wang J, Wang Y, Wu W, Xiao J et al. (2012b) Probase: a universal knowledge base for semantic search, 2012 .

Wang Z, Wang F, Wang H, Hu Z, Yan J, Li F, Wen JR and Li Z (2016) Unsupervised head–modifier detection in search queries. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11(2): 19.

Weikum G, Hoffart J and Suchanek F (2016) Ten years of knowledge harvesting: Lessons and challenges. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 39(3): 41–50.

Wu Q, Shen C, Wang P, Dick A and van den Hengel A (2018) Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence* 40(6): 1367–1381.

Wu W, Li H, Wang H and Zhu KQ (2012) Probase: A probabilistic taxonomy for text understanding. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, pp. 481–492.

Xin Luna D (2017) Challenges and innovations in building a product knowledge graph. Remarks by Xin Luna Dong at 6th Workshop on Automated Knowledge Base Construction (AKBC), 2017, `http://www.akbc.ws/2017/slides/luna-dong-slides.pptx` [Accessed: 2018 9 12].