Survey on Development of Domain-Specific Knowledge Graph

Dhaval Limdiwala, Amit Patil, and Pushpak Bhattacharyya
Indian Institute of Technology, Bombay
{djlimdi, amitpatil, pb} @cse.iitb.ac.in

Abstract

Knowledge Graph (KG) has become popular for its use in retrieving results for Google search. Google coined the term "Knowledge Graph" in 2012 for the semantic web created by them, which represents the world using nodes and edges. The knowledge graph is capable of deriving intelligence from data and reason the data automatically. Many researchers have attempted to design domain-specific KG to extract hidden insights from the domain. However, the development of the knowledge graph is onerous. In order to avoid the need of domain expertise for designing domain specific knowledge graph, researchers have proposed multiple methods. The paper summarizes the development process of domain specific knowledge graph. This paper will discuss the knowledge graph basics. The paper will also explain the existing knowledge graph and knowledge graph engineering techniques. Moreover, the paper will illustrate the knowledge graph learning techniques from a text corpus of the same domain. Finally, the paper discusses the knowledge graph evaluation techniques.

1 Introduction

In today's world, with the growth of numerous fields and technologies, a large amount of information is present over the web. However, the data is unstructured. To gain insights from the data, it must be converted into structured data. The knowledge graph is an emerging technique through which data can be represented in terms of nodes and edges. Earlier researchers used to refer knowledge graph as ontology. Ontology is an explicit conceptualization of the domain. (Gruber, 1993) defines the ontology and knowledge graph.

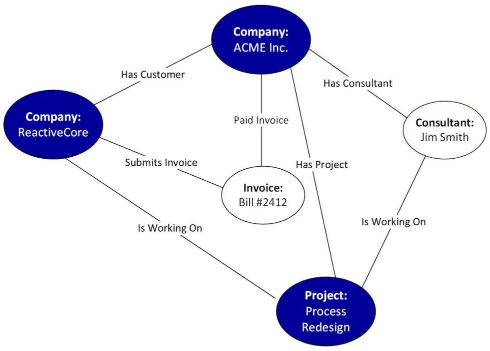

Figure 1: Example of simple KG (Enterprise Knowledge, 2019)

The knowledge graph consists of classes, individuals, object properties, data properties, and rules. The classes represent kinds of things (e.g., Place). The individuals are instances or ground level objects (e.g., Mumbai). An object property is a relation between two individuals (e.g., capitalOf). The relation between an individual and some constant data is specified by data property (e.g., hasPopulation). Moreover, the rules are the assertions which are true in that domain (e.g., A city can belong to only one country). The Figure 1 represents an example of simple KG. It has classes *Company*, *Project* etc. The individuals are ReactiveCore, ProcessRedesign, etc. The relations are isWorkingOn, hasProject, etc.

The remainder of this paper is organized as follows. In section 2, the existing knowledge graphs are explained. In section 3, the knowledge graph engineering software are discussed. In section 4, the knowledge graph learning techniques are illustrated. In section 5, the paper explains the evaluation methods for knowledge graphs. Finally, concluding remarks are provided in section 6.

2 Existing Knowledge Graphs

There are many knowledge graphs developed by various research groups. We will study a few of the existing knowledge graphs briefly in this section.

Miller (1995) designed a lexical database called WordNet. The WordNet groups similar terminologies into a set called synsets. This lexical database is designed at Princeton University. The WordNet has relations such as isA, hyperonymy, and hyponymy. isA relation is used to represent a synonym relationship. The hyponym is the specific kind of hypernym. The bird is hyponym of the animal. The animal is hypernym of the bird. The WordNet is composed of four subnetworks. The smaller networks are of nouns, verbs, adverbs, and adjectives. WordNet aims to store common English words with relationships. It does not focus on any specific domain. The development of WordNet started in 1985. The WordNet has relationships for words with the same senses. Also, it has a semantic relationship between words. The WordNet is used in many contexts. Few of the popular applications of WordNet include word sense disambiguation, text summarization, machine translation, information retrieval, etc.

Speer et al. (2017) ConceptNet is a multilingual knowledge base which stores common-sense relationships between the words The ConceptNet stores synonyms, terms with contexts, related terms, type of relationship, part of relationship, derived terms, used for relationship, capable of relationship etc. Similar to WordNet, ConceptNet is also a generic database not focusing on any specific domain. It also stores the translation of words in multiple languages. It has words from languages such as Arabic, Italian, French, Italic, etc. The ConceptNet is a more advanced version of WordNet. WordNet mostly stores lexical relations but ConeptNet stores real-life relations among things.

Type: Person	
Property	**Expected Type**
additionalName	Text
address	Text
affiliation	Organization
alumniOf	Organization
award	Text
birthDate	Date
birthPlace	Place

Table 1: Example of Person type in Schema.org with few relations

Schema.org Contributors (2019) made an attempt made by a set of companies to create a generic schema for the data present on the internet and represent the data in a structured manner. This structured organization of data helps the search engines to retrieve information from the internet and return relevant results. Schema.org represents schema for commonly used types. Some of the types present in Schema.org include Event, Organization, Person, Place, Product, Review, and Action. Schema.org stores data in various format. The formats supported by Schema.org are RDFa, Microdata, and JSON-LD. Person type has various relationships associated with it. Some relationships are mentioned in Table 1.

Google KG (Singhal, 2012) was launched in 2012. Before 2012, search engines used to return relevant links for the asked queries. However, things changed when Google introduced the Knowledge Graph (KG). This Knowledge Graph is nothing but the knowledge base used by the Google search engine to find results of the queries and answer them. The information retrieved from knowledge graph is shown in the infobox (which is displayed on the right side of the web page besides links). The infobox is also called a knowledge panel. Recently Google has started using Knowledge Graph to answer spoken questions to Google Home. The knowledge graph is developed in English, French, Spanish, German, etc. Googles knowledge graph received criticism. The reasons for criticisms are mentioned below.

- **Absence of source attribution:**

Name	License	Creator
Protege	Open Source and Free	Stanford University
Fluent Editor	Free	Cognitum
PoolParty Thesaurus Server	Commercial	Semantic Web Company
Semaphore Ontology Manager	Commercial	Smartlogic Semaphore Limited

Table 2: KG Engineering Software

Googles KG retrieves information from popular sites such as Wikipedia, Quora but those sites are not authentic, and the information fetched by KG might be wrong. The KG does not provide any source attribution to verify the authenticity of the information shown.

- **Reducing Wikipedia usage:** Before KG, people used to refer Wikipedia for static information, but now KG retrieves the required information from Wikipedia page to the knowledge panel, the readership of Wikipedia is reducing continuously.

3 KG Engineering Software

There are applications to design an ontology or knowledge graph. Using these applications, it is easy to create entities and relations. Not only this, these tools have sophisticated techniques to define the hierarchy of a large number of classes, merge two related knowledge graphs, populate instances, visualize the ontology. Also, these tools have reasons and inference engines to infer more information from present information. Table 2 shows some of the KG engineering software.

Musen (2015) designed software called Protege. It is the most popular software among KG researchers. Protege supports the W3C standard languages such as OWL2 and RDF. Protege has a highly active community, so it is easy to get advice from experts when stuck somewhere. Protege has many supporting software/plugins which make the knowledge graph development easy. Protege software is kept up to date by the Stanford University. The latest version of Protege was released on 14th March 2019. Protege is used by (Zhao et al., 2018), (Ast et al., 2014), (Wu et al., 2014), and (Carvalho et al., 2005) for designing domain-specific knowledge graph.

Fluent Editor is another software to design knowledge graph. It is designed by Cognitum (Seganti et al., 2015). The fluent editor uses the controlled natural language as an interface for editing the knowledge graph. It can process the complex controlled English language queries to generate the knowledge graph. The features of Fluent editor are autocompletion - suggests the right keywords to the user, the editor also has reasoner support, a user can import and export the already developed knowledge graph in Fluent editor. Moreover, the Fluent Editor also provides the visualization of the created knowledge graphs showing the entities and the relationships. The Fluent Editor has more than 2000 users. There are some commercial software as well to design KG such as PoolParty Thesaurus Server created by Semantic Web Company and Semaphore Ontology Manager created by Smartlogic Semaphore Limited.

4 KG Learning Techniques

In this section, we will study the techniques applied to learn the structure or the schema of the knowledge graph. Defining the classes and relations of the knowledge graph is a difficult task. The classes should be unambiguous and must represent a specific set of objects. One of the ways to decide the structure of the ontology for any domain is to ask domain experts to make it. The problem with this methodology is that manual procedure is time taking and hiring domain experts is expensive. So researchers have invented algorithms using which the ontology of a specific domain can be learned. All the techniques discussed in this chapter are unsupervised techniques. All the techniques are summarized in paper (Asim et al., 2018).

4.1 Entity Extraction

Entities are special words which belong to some predefined categories such as person, place, organization, etc. These entities become classes and individuals in the knowledge graph. We applied multiple techniques to extract entities from the corpus. The techniques are classified as statistical techniques and linguistics techniques.

4.1.1 Statistical Techniques

Statistical methods use the statistics of the text corpus to extract entities. These methods do not consider the semantic meaning of the words to give the result. These methods are mostly based on frequencies and probabilities.

Karoui et al. (2007) describes a way to discover the classes and individuals using unsupervised clustering methods. For ext corpus, the clustering method can be applied at a sentence level, and the resulting sentence clusters can be analyzed to identify the classes. The cluster quality metrics can be used to identify the clusters with good quality. The metrics are mentioned below.

- **Davies Bouldin Index (DBI):**
  In simple terms, it is the ratio of the sum of the average distance $(d(X_i)\ and\ d(X_j))$ of all the points from the centre of the clusters to the distance between the centres of the cluster $(d(c_i, c_j))$. The DBI should be as less as possible.

$$DBI = \frac{1}{c}\sum_{i=1}^{c} Max_{i\neq j}\left\{\frac{d(X_i)+d(X_j)}{d(c_i,c_j)}\right\}$$

- **Silhouette Index (SI):**
  It is a measure to check how close the intra-cluster points are and how far the clusters are situated. The silhouette ranges from 1 to +1. The high value of SI indicates good clustering. The average of $s(i)$ will give us the overall SI value.

$$a(i) = \frac{1}{|C_i|-1}\sum_{j\in C_i, j\neq i} d(i,j)$$

$$b(i) = min_{i\neq j}\frac{1}{|C_j|}\sum_{j\in C_j} d(i,j)$$

$$s(i) = \frac{b(i)-a(i)}{Max(a(i),b(i))}$$

Salton and Buckley (1988) states that the TF-IDF (Term Frequency - Inverse Document Frequency) can be used for extracting entities from text corpus. TF-IDF is a statistical method to identify how much important a term is with respect to the document given a set of documents. There are various formulae for calculating TF-IDF. Let t be the term and D be the set of all documents. Let there be N such documents and n_t be the number of documents in which the term t occurs. Let d be any sample document (i.e. $d \in D$). The formulae used by us is mentioned below.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t'\in d} f_{t',d}}$$

$$idf(t,D) = \frac{N}{n_t}$$

$$tfidf(t,d,D) = tf(t,d) * idf(t,D)$$

The TF-IDF value of a term for entire corpus is decided by maximum of all the TF-IDF values for that term with all the documents.

Frantzi et al. (2000) defined automatic multi-word extraction technique. The drawback of TF-IDF is that we cannot get multi-word terms. To overcome this drawback, *C-value* technique can be used.

$$C{-}value(a) = \begin{cases} log_2|a|.(f(a)- \\ \frac{1}{P(T_a)}\sum_{b\in T_a} f(b)), & nested \\ log_2|a|.f(a), & else \end{cases}$$

In the above mentioned formula,
a is the candidate term, for which C-value needs to be calculated
$|a|$ represents the number of words in the term
$f(a)$ is the frequency of term in the corpus
T_a is the set of candidate terms that contain a
$P(T_a)$ is the number of these candidate terms

4.1.2 Linguistics Techniques

The linguistics techniques are essentially a manual analysis of corpus at a deeper level. The base of these techniques is the characteristics of languages and sentence structures etc. There are no objective metrics present for linguistics techniques. The correctness is solely based on the ontologists and domain experts opinion.

Hippisley et al. (2005) defines a method to identify the entities using the syntactic analysis of the text. In the syntactic analysis, the annotated corpus is analyzed based on some heuristics. The corpus is annotated with part-of-speech tags. All the sentences with multiple nouns are extracted, and related entities (part of, synonym of) are identified.

(Hwang, 1999) describes developing an ontology using seed words. Seed words are the domain-specific words which represent high-level concepts. The authors extracted the phrases from the corpus containing the seed words and then learned the hierarchical ontology. The author gave an example of ontology related to *image*. The author gave *display* as the seed word; then he got the following ontology:

```
display
    field emission display
    flat panel display
display panel
display panel substrate
```

4.2 Relation Extraction

Similar to entity extraction, relation extraction techniques are also divided into two categories - statistical and linguistics techniques. There are two kinds of relationships in the ontology - taxonomic relationships and non-taxonomic relationships. The taxonomic relationships are the hierarchical relations (e.g. *subclassOf*) and the non-taxonomic relations are non-hierarchical relations (e.g. *causeOf*).

4.2.1 Statistical Techniques

In this section, we will study two methods. The hierarchical clustering will be used to identify the taxonomic relations, and the Association Rule Mining will be used to discover non-taxonomic relations (Asim et al., 2018).

Dhillon et al. (2003) used the divisive clustering to identify the classifying the text and design the taxonomic relations. In this technique, a large cluster is sub-divided into small clusters.

Agrawal et al. (1996) uses apriori algorithm to find relations. This technique is called Association Rule Mining (ARM). ARM is an approach to identify closely associated terms from the corpus. It is commonly used to discover non-taxonomic relationships. This algorithm takes a set of transactions as input. The transaction can be a sentence or subset of words present in the sentence after removing stopwords and returns the pair of terms which are related with the score. This algorithm is illustrated in the following two steps.

1. Generate frequently occurring item sets X and Y together based on *suppport* greater than *threshold support*. The *support* of an item set fraction of times the item set X and Y occur together $(X \cup Y)$.

$$support = \frac{X \cup Y}{Total\ no.\ of\ transactions}$$

2. Filter the above results on the bases of *confidence*. And, then name the relations between the item sets manually.

$$confidence = \frac{X \cup Y}{X}$$

4.2.2 Linguistics Techniques

In this section, we will discuss the linguistics techniques applied to learn the relations - Dependency Analysis and Open Information Extraction (OpenIE).

(Fundel et al., 2006) uses a dependency analysis of sentences to extract relations. This technique is based on an observation that most of the relations are verbs. To apply this technique, the text corpus is annotated using part-of-speech tags. Then all the sentences should be parsed using dependency parser. After parsing all the sentences, the verbs and associated nouns are observed to discover the relations.

(Mausam, 2016) states that Open Information Extraction (OpenIE) can be used to identify the relations present in sentences using OpenIE. The technique of Open Information Extraction can be used when there is no training data. The implementation of OpenIE is available online.

4.3 Rule Learning

Rules or axioms in the knowledge graph gives definitions of the classes and properties to constrain the scope. Rules play a crucial role in automated reasoning and inferring new information.

Bühmann et al. (2016) designed software called DL-Learner, which can generate rules for the knowledge graph. DL-Learner is a software developed in Java. It is a machine learning framework designed to identify the rules of the domain automatically. The research to improve these algorithms is still going on. DL-Learner can suggest rules regarding equivalent class expression, subclass expression, the domain of the property and the range of the property. The software observes the OWL knowledge base to suggest rules. For example, consider a simple knowledge graph of people having class as Father. After observing all the individuals populated in the Father class, the DL-Learner will identify that all instances have a child and also belong to class Male. So the software will suggest the following rule in the equivalent section.

```
(male and hasChild some Thing)
```

The software is freely available on the internet. It is also compatible with Protege.

5 Knowledge Graph Evaluation

Brank et al. (2005) describes the four methods to evaluate the KG. The methods are - gold standard based, data-driven, application based, and human evaluation. All the methods are described below.

Gold standard based evaluation:

In this method, the designed knowledge graph is compared with the existing knowledge graph of the same domain. The challenge in this type of evaluation is to find the knowledge graph of the same domain. Two knowledge graphs can be compared based on concepts, hierarchy, and relations. Ponzetto and Strube (2007) compared the taxonomy extracted from Wikipedia with the gold standard ontology.

Application based evaluation:

The application based evaluation is done by exploiting the designed knowledge graph for some use case. The KG is analysed based on coverage of competency questions. Competency questions are the queries which are expected to be answered by a knowledge graph. The competency questions are generated from the knowledge graph requirement specification document. The percentage of queries which can be answered by the knowledge graph is the coverage of the knowledge graph. Brank et al. (2005) explains the past application based evaluation of the knowledge graph.

Data-driven evaluation:

In data-driven evaluation, the designed knowledge graph is compared to the corpus or standard of the domain. This type of evaluation is only possible when there exists some existing standard in the domain. This method cannot verify the structure, architecture, or design of the knowledge graph. It can only verify the concepts, hierarchy, and relations.

Human Evaluation:

This is the most reliable evaluation method among all the knowledge graph evaluation methods. The knowledge graph is evaluated by the two kinds of experts. Initially, the KG is evaluated by the KG expert. KG expert is the person who has built KG in the past (may or may not be of the same domain). The KG expert will comment on the annotations and conventions followed in the designed knowledge graph.

The second expert who evaluates the KG is the domain expert. The domain expert will comment on coverage, ambiguity, structure, and design of knowledge graph. Hiring ex-

Level	**Golden Standard**	**Application based**	**Data driven**	**Assessment by humans**
Lexical, vocabulary, concept and data	x	x	x	x
Hierarchy and taxonomy	x	x	x	x
Other semantic relations	x	x	x	x
Context and application		x		x
Syntactic	x			x
Structure, architecture and design				x

Table 3: Overview of KG evaluation approaches (Brank et al., 2005)

perts for evaluation is expensive, and the evaluation phase takes time.

6 Conclusion

In this paper, we covered the process of developing a domain-specific knowledge graph. Initially, we described the basics of the knowledge graph and knowledge graph components. Then, we studied the existing knowledge graphs such as WordNet, ConceptNet, and Google's Knowledge Graph. We also explored the Knowledge Graph engineering software. We studied Protege and Fluent editor in brief.

We studied the knowledge graph learning techniques. We described the statistical and linguistics techniques for entity and relation extraction. Moreover, we discussed the DL-Learner software to extract rules from the knowledge graphs. Finally, we studied the knowledge graph evaluation techniques. Various researchers are trying automate the procedure of making knowledge graph.

References

Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A Inkeri Verkamo, et al. 1996. Fast discovery of association rules. *Advances in knowledge discovery and data mining*, 12(1):307–328.

Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. 2018. A survey of ontology learning techniques and applications. *Database*, 2018.

Markus Ast, Martin Glas, Tobias Roehm, and VB Luftfahrt. 2014. *Creating an ontology for aircraft design*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV.

Janez Brank, Marko Grobelnik, and Dunja Mladenic. 2005. A survey of ontology evaluation techniques. In *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, pages 166–170. Citeseer Ljubljana, Slovenia.

Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. 2016. Dl-learnera framework for inductive learning on the semantic web. *Journal of Web Semantics*, 39:15–24.

Robert Carvalho, Shawn Wolfe, Dan Berrios, and James Williams. 2005. Ontology development and evolution in the accident investigation domain. In *2005 IEEE Aerospace Conference*, pages 1–8. IEEE.

Inderjit S Dhillon, Subramanyam Mallela, and Rahul Kumar. 2003. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of machine learning research*, 3(Mar):1265–1287.

Enterprise Knowledge. 2019. What is ontology? https://enterprise-knowledge.com/what-is-an-ontology/. [Online; accessed 6-June-2019].

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries*, 3(2):115–130.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2006. Relexrelation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Tom Gruber. 1993. What is an ontology. *WWW Site http://www-ksl. stanford. edu/kst/whatis-an-ontology. html (accessed on 06-06-2019)*.

Andrew Hippisley, David Cheng, and Khurshid Ahmad. 2005. The head-modifier principle and multilingual term extraction. *Natural Language Engineering*, 11(2):129–157.

Chung Hee Hwang. 1999. Incompletely and imprecisely speaking: using dynamic ontologies for representing and retrieving information. In *KRDB*, volume 21, pages 14–20.

Lobna Karoui, Marie-Aude Aufaure, and Nacera Bennacer. 2007. Contextual concept discovery algorithm. In *FLAIRS Conference*, pages 460–465.

Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4074–4077. AAAI Press.

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

Mark A. Musen. 2015. The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *AAAI*, volume 7, pages 1440–1445.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Schema.org Contributors. 2019. Schema.org. https://schema.org/. [Online; accessed 6-June-2019].

Alessandro Seganti, Paweł Kapłański, and Piotr Zarzycki. 2015. Collaborative editing of ontologies using fluent editor and ontorion. In *International Experiences and Directions Workshop on OWL*, pages 45–55. Springer.

Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html. Accessed: 2019-06-05.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Yuchang Wu, Vahid Ebrahimipour, and Soumaya Yacout. 2014. Ontology-based modeling of aircraft to support maintenance management system. In *IIE Annual Conference. Proceedings*, page 1159. Institute of Industrial and Systems Engineers (IISE).

Qian Zhao, Qing Li, and Jingqian Wen. 2018. Construction and application research of knowledge graph in aviation risk field. In *MATEC Web of Conferences*, volume 151, page 05003. EDP Sciences.