Probabilistic Topic Modeling Based Framework for Exploration of RDF Data

by

Seyedamin Pouriyeh

(Under the Direction of Krzysztof J. Kochut)

Abstract

During the past few decades, the amount of Web content has grown exponentially. Recently, a vast amount of datasets in a variety of domains has been published as part of the Linked Open Data (LOD) project. As a result, exploring and exploiting these massive heterogeneous datasets, which are typically represented using the Resource Description Framework (RDF), has gained a considerable attention. With this work, we aim to address exploring and exploiting of such datasets within two broad categories: RDF dataset summarization and RDF dataset profiling. With respect to RDF dataset summarization, we focus on entity summarization, which aims to produce an abridged, but sufficient descriptions of all entities in the dataset. In other words, entity summarization is a way to absorb and distill descriptive knowledge from RDF datasets. We propose a probabilistic topic model using Latent Dirichlet Allocation (LDA) for the entity summarization task called *ES-LDA* and its extension, *ES-LDA*$_{ext}$, which combines prior knowledge

with statistical learning techniques within a single framework, in order to create more reliable and representative summaries of entities. We demonstrate the effectiveness of our approach by conducting extensive experiments and show that our models outperform state-of-the-art techniques and enhance the quality of the entity summaries. RDF dataset profiling is a task that involves generating a proper profile for RDF datasets on the Web so that they can be discovered more easily. Basically, RDF dataset profiles are expected to facilitate data discovery, consumption, and integration with statistics and useful metadata about the content of the RDF datasets. We propose topic-wise RDF dataset profiling, called *R-LDA*, using LDA technique. In our model, we identify a number of topics that can represent an RDF dataset and assign a set of Wikipedia categories to the obtained topics that are semantically relevant, understandable, and cover the discovered topics well. The union of the assigned categories serves as a profile of the dataset, in a sense that it provides an overall characterization of the datasets content.

INDEX WORDS: Ontology Summarization, Entity Summarization, Semantic Web, RDF/S Exploring, Topic Modeling, RDF Profiling.

Probabilistic Topic Modeling Based Framework for Exploration of RDF Data

by

Seyedamin Pouriyeh

M.Sc., Shiraz University, 2009

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2018

Probabilistic Topic Modeling Based Framework for Exploration of RDF Data

by

Seyedamin Pouriyeh

Approved:

Major Professors:	Krzysztof J. Kochut
Committee:	Hamid Reza Arabnia
	Juan B. Gutierrez
	Mehdi Allahyari

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2018

Probabilistic Topic Modeling Based Framework for Exploration of RDF Data

Seyedamin Pouriyeh

July 2018

To my family: Soheyla and Amir

Acknowledgments

This dissertation would not have been possible without the continuous support, friendly guidance, and expert advice of my major advisor, Professor Krys Kochut. I would like to acknowledge my indebtedness and extend my warmest thanks to him.

I would like to express my sincere thanks to Professor Hamid Reza Arabnia for generously sharing his time and supporting my career goals. I greatly appreciate his excellent assistance and spiritual support of my family and me during my PhD study.

The dissertation has also benefited from valuable comments and suggestions made by Professor Juan Gutierrez. I take this opportunity to thank him. I am especially indebted to Professor Mehdi Allahyari for extended discussions and valuable suggestions that have contributed greatly to the improvement of my dissertation.

I would like to thank my parents, whose love, support, and guidance are with me in whatever I pursue. Most importantly, I wish to thank my loving and supportive wife, Soheyla, who provides unending inspiration. Without their support and encouragement, I might never have completed this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the past few decades, the amount of Web content and in particular Linked Open Data (LOD)[1], enabled by widespread use of the Internet, has been constantly growing. As a result, massive heterogeneous datasets which are typically represented as Knowledge Bases (KBs) have become freely available to research communities.

Different general purpose knowledge bases, often built on encyclopedic knowledge, are publicly available and contain large amounts of knowledge for human and machine consumption. Prominent KB examples include DBpedia [8], Wikidata [93], YAGO [85], and Freebase [20]. Freebase, which is considered a collaborative KB, was created based on Wikipedia data. Later, with the shut down of Freebase, its content was transferred to Wikidata, another KB proposed by Wikimedia Foundation [93, 69]. YAGO, which was developed in the Max Planck Institute for

[1] http://lod-cloud.net/

Computer Science [2] is an automatically extracted knowledge base from Wikipedia, GeoNames [3], and WordNet [4]. The most popular knowledge base, which is also considered as a central interlinking hub within the LOD, is DBpedia. It is often known as the structured version of Wikipedia for mainly machine consumption.

In general, KBs provide information about entities that are mostly represented in the form of ontologies. An ontology is typically defined by two different layers, including the schema layer and data the layer. The two layers make up a framework to represent knowledge bases, including classes, entities, and relationships among them.

Entities within KBs that are uniquely identifiable things or objects refer to real world and abstract things that can be described through their relations (properties) with other things (entities) such as persons, organizations, and places [11]. These relations in a knowledge base are often stored in RDF datasets and represented in RDF triples format, <`subject, predicate, object`> where the *subject* is an entity, the *predicate* is a relation, and the *object* is another entity (or literal).

With the exponential growth of LOD in recent years, an abundance of knowledge via RDF datasets has become available on the Web. These datasets vary with respect to their topics, domain coverage, size, complexity, and so forth. Given this scale of these RDF datasets, their heterogeneity, often their inconsistency, and the lack of meta-data, finding the resources that can be linked, queried, or reused in different applications has become an interesting area of research in the Semantic

[2] https://www.mpi-inf.mpg.de/home/
[3] http://www.geonames.org
[4] https://wordnet.princeton.edu/

Web communities. RDF dataset profiling techniques, which aim to facilitate data consumption and data integration with statistics and useful meta-data about the content of the RDF datasets, are considered as an effective approach to this challenge. Additionally, as KBs are publicly available to change and grow in size, the number of entities and their representations (via their triples) have increased. As a result, entity descriptions often come with a large volume of statements (triples) which make it difficult to comprehend the data, unless we can use an approach that allows us to select the most relevant facts about that entity. Among the proposed models to address this challenge, ontology summarization and in particular entity summarization techniques have attracted a particular attention in recent years.

1.1 RDF Dataset Profiling

Linked Open Data (LOD) has seen an exponential growth via publishing huge volume of RDF datasets on the Web. In order to identify suitable RDF datasets for different applications and enterprises, W3C[5] recommends that potential data publishers provide recapitulative information on their datasets available on the Web. This information, which functions as meta-data, will facilitate access to the datasets to be discovered, queried, and interlinked more easily. Because this information is not always available, we face with a large number of datasets without a proper profile, leading to a high demand for different data profiling techniques. In general, there is no comprehensive definition for data profiling and the related

[5]https://www.w3c.org/

tasks; however, RDF dataset profiling typically covers several tasks including the following [7]:

- *Statistical profiling:* this type of data profiling mainly focuses on statistical information about the number of entities as well as the distribution of properties and RDF triples, which are often related to data types and patterns in the dataset.

- *Metadata profiling:* dataset profile with respect to the metadata should cover the main informative categories, including the general information (dataset description, release and update dates), practical information (access points, data dumps), and legal information (license information, openness).

- *Topical profiling:* the representative knowledge on the content and structure of the dataset in the form of tags, keywords, categories, informative subgraphs, etc.

In this dissertation, we focus on topical profiling techniques in order to generate a data profile for a given RDF dataset.

1.2 Ontology Summarization

As the size and the complexity of ontologies increase, there is a need to facilitate ontology comprehension, exploration, and exploitation and to help users take advantage of an ontology quickly. There are numerous ontology management techniques, including ontology partitioning, ontology segmentation, ontology summarization, and others, that attempt to provide efficient and effective models to

address these challenges. In ontology partitioning, an ontology is divided into subsets, called partitions, to alleviate certain challenges of large ontologies, such as scalability, complexity, and maintenance. Ontology segmentation, on the other hand, tries to find extractable parts of an ontology that can be reused outside its original context. Usually, a relevant segmented knowledge is acquired from the whole ontology for the purpose of increasing tractability for both humans and machines. In ontology summarization, which is usually defined as a "the process of distilling knowledge from an ontology in order to produce an abridged version for different tasks" [101], the more important and representative ontology entities and relationships are selected. Ontology summarization is also applicable with respect to its different layers. In data layer, an entity description often comes with a large volume of statements which makes it difficult to comprehend the data unless we can use an approach that allows us to select the most relevant facts about that entity. Among the proposed models to address this challenge, ontology summarization, and in particular ontology summarization at the data layer, which is typically called entity summarization, have gained a more attention in recent years.

1.3 Topic Modeling

Recently, the topic modeling approach has become a popular method for uncovering the hidden themes from data such as text corpora, images, and so forth. This model has been widely used for various text mining tasks, such as machine translation [84], word embedding [12, 32], automatic topic labeling [95, 4, 6], and others

[5]. In the topic modeling approach, each document is considered as a mixture of topics, where a topic is a probability distribution over words. When the topic proportions of documents are estimated, they can be used as the themes (high-level semantics) of the documents. In this dissertation, we propose a probabilistic topic model that combines prior knowledge with statistical learning techniques within a single framework to create more reliable and representative summaries for entities. Additionally, we use topic modeling for RDF dataset profiling using Wikipedia categories.

1.4 Contributions

The main research theme governing this dissertation is the integration of prior knowledge with the topic modeling to explore RDF datasets for summarization and profiling tasks. We propose different novel topic models which focus mainly on summarization. We develop a probabilistic topic model called ES-LDA and the extended version of that model, ES-LDA$_{ext}$, which combines prior knowledge with statistical learning techniques within a single framework to create more reliable and representative summaries for entities. We demonstrate the effectiveness of our approach by conducting extensive experiments and show that our model outperforms the state-of-the-art techniques and enhances the quality of the entity summaries. Additionally, we present a new topic-wise RDF dataset profiling, called R-LDA, utilizing topic modeling technique and Wikipedia categories. Note that the application of the proposed models are not limited to summarization and

profiling and can be generalized to other problems such as ontology partitioning, tagging, and so on.

The main contributions of this research can be summarized as follows:

- We propose an entity summarization technique based on probabilistic topic model, ES-LDA, and show the benefits of our method over traditional ones, and demonstrate its effectiveness through comprehensive evaluation.

- In the extended version of ES-LDA, called ES-LDA$_{ext}$, we take advantages of the Word2Vec technique in order to enhance the quality of the summary. The results also confirm the effectiveness of our model.

- We have a comprehensive survey on ontology summarization techniques with respect to terminological definitions in ontologies (schema layer). We mainly sort, review, and compare various *graph-based methods* for ontology summarization.

- We present a novel topic-wise RDF dataset profiling model, called R-LDA, using Wikipedia categories. The proposed model aims to find number of topics that represent a given RDF dataset and assign the representative categories from Wikipedia which are semantically relevant, understandable for humans and highly cover the discovered topics.

1.5 Dissertation Overview

The reminder of this dissertation is organized as follows:

- In Chapter 2, we primarily focus on the preliminary definitions that we need throughout this dissertation. We describe the Semantic Web and explain a couple of primary concepts and standards associated with it. In addition, we explore the Latent Dirichlet Allocation (LDA) topic model and inference algorithms for topic models

- Related work with respect to ontology summarization and RDF dataset profiling are the topics that we mainly discuss in Chapter 3.

- Chapter 4, describes existing ontology summarization techniques and measures (at the schema layer) corresponding to each technique to identify the most important elements of an ontology. We mainly sort, review, and compare various *graph-based methods* for ontology summarization.

- Chapter 5, is dedicated to our model, ES-LDA, which utilizes topic modeling technique within a single framework to create more reliable and representative summaries for entities. We demonstrate the effectiveness of our approach by conducting extensive experiments and show that our model outperforms the state-of-the-art techniques and enhances the quality of the entity summaries.

- Chapter 6, which considers the extended version of ES-LDA model, ES-LDA$_{ext}$, focuses primarily on new augmentation techniques in order to enhance the entity summarization results.

- Chapter 7, mainly discusses R-LDA model for RDF dataset profiling using topic modeling.

- Chapter 8, concludes the dissertation, summarizing the contributions and describing directions for further research building on the foundations established in this work.

Chapter 2

Background

In this chapter, we present the Semantic Web concept and its associated components. Understanding the nature, purpose and principles of the Semantic Web are the key points before the challenges of the RDF exploring. We begin with a big picture of the World Wide Web and the Semantic Web and a short background description related to ontology, Resource Description Framework (RDF) and RDF Schema (RDFS). Finally, we continue with describing probabilistic topic modeling technique, and in particular Latent Dirichlet Allocation (LDA), a technique we use to explore RDF data.

2.1 World Wide Web and Semantic Web

The World Wide Web (WWW) was invented by Tim Berners-Lee [15] in 1989 with the idea of accessing documents using different machines through the Web via the Internet. His idea was based on combining three technologies includ-

ing Uniform Document Identifier (UDI), later called Uniform Resource Identifier (URI), designed to uniquely identify a document; Hyper Text Markup Language (HTML), used to publish documents; and Hypertext Transfer Protocol (HTTP), employed to enable communications between machines. Over the years, with the explosion of both the quantity and range of data over the Web, finding, sharing, and exchanging data on the Web have become increasingly difficult. The Semantic Web [14, 13], as an extension of the WWW, aims to describe the meaning of web content using semantic annotations in order to enable data to be found, shared, and reused among different applications and enterprises. In 2001, Tim Berners-Lee [13] proposed the main idea of evolving Web content into the Semantic Web where Semantic Web aims to structures the Web and to allow information sharing/exchanging across applications using a framework that makes the data not only human-readable but also represented it in a form that is machine-processable [14]. The key point in the Semantic Web is defining a common model that enables easy communication among different platforms. Ontology term, which was highlighted more in 2001 by Tim Berners-Lee [13], functions as a framework that allows for implementation of sharable and exchangeable environment for Web content.

Tim Berners-Lee in [13] describes the Semantic Web as *"...The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*

A few technologies which are usually associated with Semantic Web including Ontology, RDF/S, and Linked Data will be discussed later in forthcoming sections.

2.1.1 Ontology

Ontologies, which are considered as basic building blocks of the Semantic Web, have been designed and used as a way to represent knowledge in different domains. In general, an ontology proposes a common vocabulary to enable exchange, sharing or reuse of domain knowledge. Concepts in the domain and the relations among the concepts are two key components which play an important role in each ontology. An ontology is typically defined by two different layers. The first layer, called the schema layer, the semantic layer, or an ontology's TBox, functions as meta-data and describes the fundamental aspects of the data layer. The other layer, called the data layer or ontology's ABox, stores the actual data according to the defined schema layer. The two layers make up a framework that represents knowledge bases, including classes, entities, and relationships among them.

DBpedia [8], Wikidata [93], YAGO [85] and Freebase [20] are among the most prominent knowledge bases, that are freely available to research communities. They contain large amounts of knowledge for human and machine consumption. The aforementioned knowledge bases consist of millions of entities and billions of edges (relations) that connect those entities together through proper relations and make a large-scale knowledge graph. Entities in a knowledge based data graphs refer to real world and abstract things that can be described through their relations (properties) with other things (entities). Resource Description Framework (RDF), RDF Schema, and Ontology Web Language (OWL) are common recommended languages by W3C to represent ontologies [59]. In the next section we explain RDF in details.

2.1.2 RDF and RDF Schema

RDF is an XML-based language recommended by W3C for describing resources on the Web. RDF relies heavily on the Web structure, utilizing many of its features and extending them in order to define distributed network of data. RDF uses URIs [1] for identifying resources (location, person, web page, organization, etc) on the Web and describes them in statement form with named properties and values. Statements in RDF are represented in the form of *triples* including *Subject*, *Predicate*, and *Object*, <*Subject, Predicate, Object*>, where *Subject* denotes a given resource that has property value (*Object*) for property *Predicate*. In an RDF statement, a *Subject* is typically a URI reference, *Object* can be a URI reference or Literal [2] and *Predicate* is a URI reference. RDF statements with the same subjects can form a graph-shaped representation of RDF statements which is called an RDF graph (Figure 2.1).

In such an RDF graph, all the resources, including subjects and objects (literals), are represented as vertices, resources are represented within, and predicates are described as the labels of directed edges from subjects to corresponding objects (literals). An example of nine RDF triples and corresponding RDF graph from DBpedia are depicted in Table 2.1[3] and Figure 2.1 respectively (by convention oval and rectangle are drawn around resources and literals respectively). The first statement in Table 2.1 conveys the information that `J.C.Penney is`

[1] Uniform Resource Identifier.

[2] Literal is a string often comes with URI datatype, which utilizes lexical format to identify values.

[3] For simplification, terms of a statement are represented in an abbreviation form in table and figure.

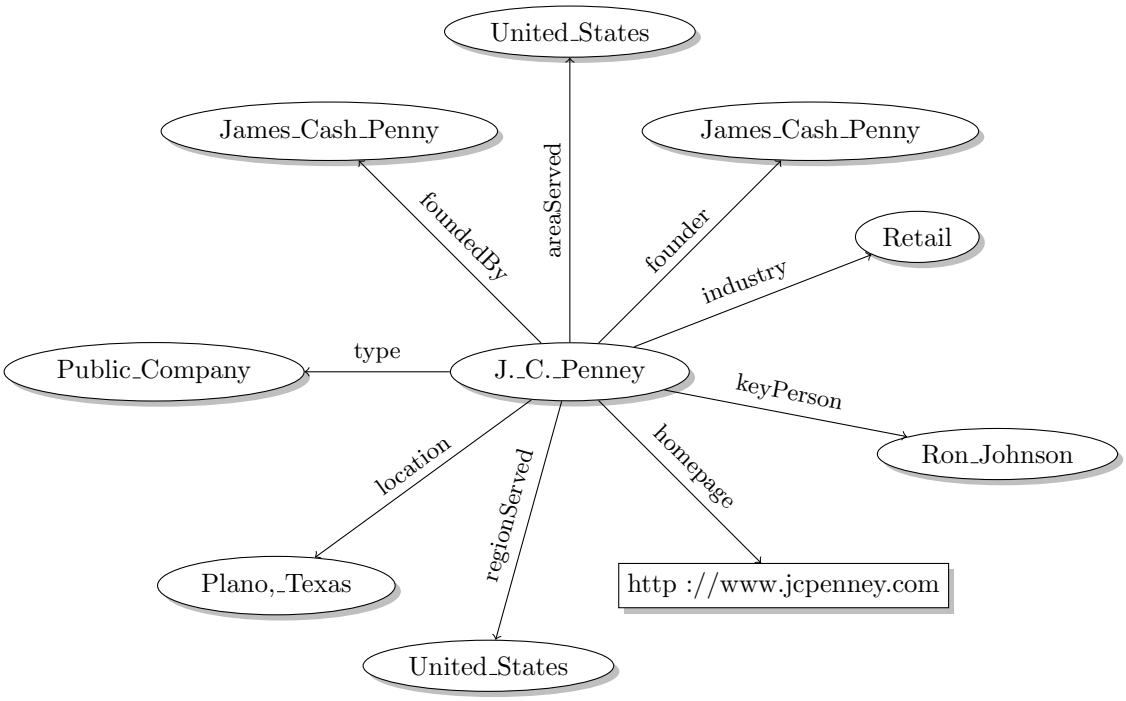


Figure 2.1: Sample entity description of the entity J.C.Penney in RDF graph format.

`a type of Public Company`. Similarly, the remaining statements describe other characteristics of the `J.C.Penney` entity.

An RDF graph typically comes with RDF Schema (RDFS), which describes vocabularies used in RDF statements. RDFS is proposed by W3C and functions as RDFs vocabulary description language. In other words, RDFS describes properties, classes of resources, and the relation among them in another layer on top of RDF [22]. RDFS has some predefined semantic terminology such as *Class* and *subClassOf* where it makes the hierarchy of classes using *subClassOf* property.

Table 2.1: Example of RDF triples for J.C.Penney entity

Triples
<dbpedia[1]:J._C._Penney><rdf[2]:type><dbpedia-owl[3]:Public_Company>
<dbpedia:J._C._Penney><dbpprop[4]:foundedBy><dbpedia:James_Cash_Penney>
<dbpedia:J._C._Penney><dbpprop:areaServed><dbpedia:United_States>
<dbpedia:J._C._Penney><dbpprop:founders><dbpedia:James_Cash_Penney>
<dbpedia:J._C._Penney><dbpedia-owl:industry><dbpedia:Retail>
<dbpedia:J._C._Penney><dbpedia-owl:keyPerson><dbpedia:Ron_Johnson>
<dbpedia:J._C._Penney><dbpprop:homepage><http://www.jcpenney.com>
<dbpedia:J._C._Penney><dbpprop:regionServed><dbpedia:United_States>
<dbpedia:J._C._Penney><dbpprop:location><dbpedia:Plano,_Texas>

[1] http://dbpedia.org/resource/ [2] http://www.w3.org/1999/02/22-rdf-syntax-ns#
[3] http://dbpedia.org/ontology/ [4] http://dbpedia.org/property/

As an example, consider that "Person" is defined as a *class* and "Grad Student" is defined as a *subClassOf* of the "Person". Therefore, if "Amin" is represented as a "Grad Student", we can easily infer that "Amin" is a type of "Person" due to the semantics of the RDFS.

2.1.3 Web Ontology Language (OWL)

The Web Ontology Language (OWL) [68, 33] is the W3C recommendation to make Web resources more processable for applications and enterprises by adding information about the resources. OWL, which is considered as one of the most expressive standardized Semantic Web languages, is placed on top of RDF/S. OWL facilitates the representation of the meanings of terms that are utilized in vocabularies and also relationships between those terms. Applying additional

vocabularies with formal semantics, makes OWL as a powerful language for expressing meaning and semantics in comparison with XML, RDF, and RDFS for expressing meaning and semantics. In other words, OWL provides more abilities for describing machine readable and interpretable contents [60]. OWL is represented through three different sublanguages including OWL-Lite, OWL-DL and OWL-Full which have been designed for the use of specific communities and users. OWL-Lite is the simplest version of OWL language and corresponds to description logic, OWL-DL aims to support maximum expressiveness with computational completeness and decidability, finally, OWL-Full comes with maximum expressiveness and the syntactic freedom of RDF [38, 60].

2.2 Linked Data

Linked data aims to provide large scale integration of, and reasoning on data on the Web [94]. In other words, linked data is a method of publishing semi-structured data in such a way that it is interlinked with other data sources in order to facilitate the discovery of new knowledge. Linked Data, which is based on the standard Web technologies such as HTTP, RDF, and URI, was invented by Tim Berners-Lee. He provided the following set of rules for publishing linked data on the web [94]:

1. Use URIs as names for things.

2. Use HTTP URIs, so that people can look up those names.

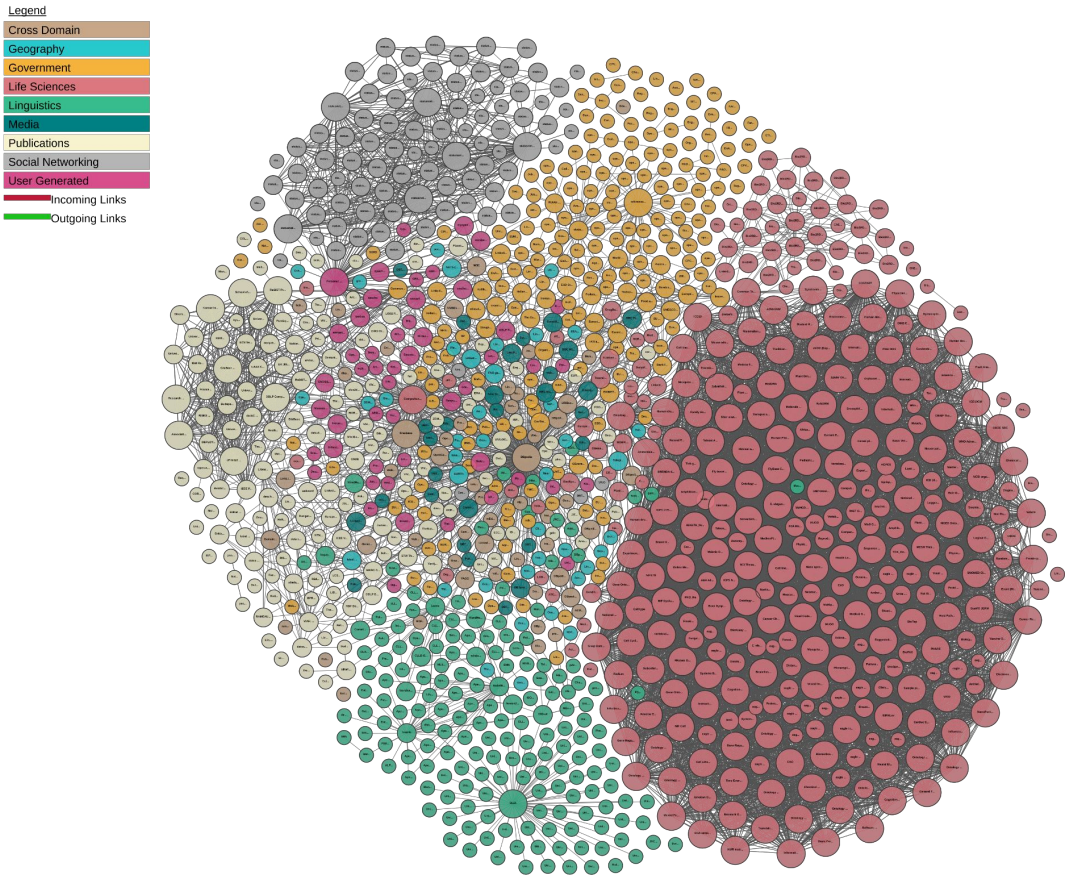


Figure 2.2: LOD cloud diagram[4]

3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).

4. Include links to other URIs, so that they can discover more things.

Following the aforementioned rules, Linked Data has grown exponentially via publishing datasets on the Web. As a result, it includes a vast number of inter-

connected datasets that can be classified as domain specific, domain independent, or even encyclopedic in nature. These datasets can be used in a multitude of applications. The most recent Linked Open Data (LOD) cloud depicted in Figure 2.2 provides an overview of the linked data sets that are available on the Web. DBpedia is known as one of the most important and primary datasets of LOD [8] and is essentially an ontology containing structured information extracted from the Wikipedia documents. It consists of description logic, known as schema layer, and data repository, called data layer, resides on the Web in the LOD cloud. The most recent DBpedia's ontology[5] contains 685 classes, which form a subsumption hierarchy and are described by 2,795 different properties. As of April 2016, the English version of the DBpedia[6] knowledge base, describes 6 million entities, of which 4.6 million have abstracts, 1.53 million have geo coordinates and 1.6 million depictions.

Recently, enhancing the intelligence of Web, enterprise search, and information integration using DBpedia knowledge bases are attracting more attention in different communities for variety of applications. Additionally, as the datasets on DBpedia continue to grow in size and complexity [10], there is great demand for new management techniques such as compression [51], summarization [87, 89], partitioning [54, 97], and profiling [1] because these techniques allow useful information to be extracted and a quick snapshot to be provided.

[5]https://wiki.dbpedia.org/services-resources/ontology
[6]https://wiki.dbpedia.org/dbpedia-version-2016-04

2.3 Probabilistic Topic Models

Probabilistic topic models are built on the assumption that there is a hidden thematic structure behind each observation in a dataset. In the case of a corpus of documents, the expected assumption is that there is a hidden topic behind each word. In topic models, documents are considered as a mixture of topics, where a topic is a probability distribution over words. There are two main topic models including Probabilistic Latent Semantic Analysis (pLSA) proposed by Hofmann (1999) [48] and Latent Dirichlet Allocation (LDA) coined by Blei [19]. pLSA mainly focuses on document modeling and does not provide any probabilistic model at the document level which makes it difficult to generalize it to model new unseen documents while LDA, which is considered as an extended version of pLSA, by utilizing a Dirichlet prior on mixture weights of topics per documents.

The ultimate goal of Latent Dirichlet Allocation (LDA) [19] is discovering the thematic structure of a collection of documents and annotate each document based on the extracted thematic. The basic assumption in LDA is that each document in a corpus of documents is exhibited various topics, where each topic is a probability distribution over vocabularies.

Based on this assumption, consider $\mathcal{D} = \{d_1, d_2, \ldots, d_D\}$ as a corpus of document and $\mathcal{V} = \{w_1, w_2, \ldots, w_V\}$ is the vocabulary of the corpus. A topic $z_j, 1 \leq j \leq K$ is described as a multinomial probability distribution over the $|\mathcal{V}|$ words, $p(w_i|z_j)$ while $\sum_i^{|\mathcal{V}|} p(w_i|z_j) = 1$. Figure 2.3 depicts the graphical model of LDA. Each node in this model is a random variable and its role in the model

defines its label accordingly. The latent nodes are unshaded while observed ones are shaded. The rectangles are called plate notion and refer to replication.

Based on the graphical model of LDA in Figure 2.3, the generative process for the corpus $\mathcal{D}$ is as follows:

1. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \text{Dir}(\beta)$
2. For each document $d \in \{1, 2, \ldots, \mathcal{D}\}$,
   (a) Sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
   (b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
      i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
      ii. Sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

The ultimate goal of LDA is computing conditional distribution of hidden variables $(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}})$ given the observed ones $(w_{1:\mathcal{D}})$. The aforementioned conditional distribution the topic structure given the observed documents, known as *posterior probability*, is calculated as follow:

$$P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}} | w_{1:\mathcal{D}}) = \frac{P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, w_{1:\mathcal{D}})}{P(w_{1:\mathcal{D}})} \tag{2.1}$$

The numerator, the joint distribution of all the random variables, can be calculated for any configuration of hidden variables based on the equation 2.2.

$$P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, w_{1:\mathcal{D}}) = \prod_{j=1}^{K} P(\phi_j|\beta) \prod_{d=1}^{|\mathcal{D}|} P(\theta_d|\alpha) \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:K}, z_{d,n}) \right) \tag{2.2}$$

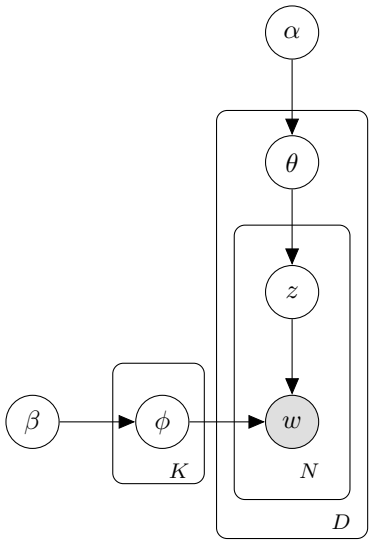


Figure 2.3: The LDA Graphical Model

The denominator, called *marginal probability*, is intractable [19] and the exact value of posterior distribution can not be computed; however, there are wide variety of efficient methods such as sampling or variational techniques available for approximating it. *Gibbs Sampling* which is most commonly used sampling algorithm tries to collect samples from the posterior distribution in order to approximate it with an empirical distribution.

Gibbs sampling, which is an iterative approach, initially starts with randomly assigned topics to all words, then the algorithm iterates over all the words and in each iteration, it samples a new topic assignment for each word using the conditional distribution of that word given all other current word-topic assignments

based on equation 2.3. The word-topic probability distributions will be available when the algorithm reaches a steady state after the final iteration.

$$P(z_i = k|w_i = w, \mathbf{z}_{-i}, \mathbf{w}_{-i}, \alpha, \beta) = \frac{n^{(d)}_{k,-i} + \alpha}{\sum_{k'=1}^{K} n^{(d)}_{k',-i} + K\alpha} \times \frac{n^{(k)}_{w,-i} + \beta}{\sum_{w'=1}^{W} n^{(k)}_{w',-i} + W\beta} \tag{2.3}$$

where $z_i = k$ is the topic assignment of word i to topic k, z_{-i} refers to the topic assignments of all other words. $n^{(k)}_{w,-i}$ is the number of times word w assigned to topic k excluding the current assignment. Similarly, $n^{(d)}_{k,-i}$ is the number of times topic k is assigned to any words in document d excluding the current assignment. For a theoretical overview on Gibbs sampling see [25, 45].

2.4 Word Embedding

In Natural Language Processing (NLP), vector space models [81] are usually utilized to represent words in a vector space [2]. This space contains all words, while semantically similar words are introduced in a such way that they can be close to each other [91]. Mikolov et al. [63] proposed Word2Vec as an efficient technique to create these vector spaces.

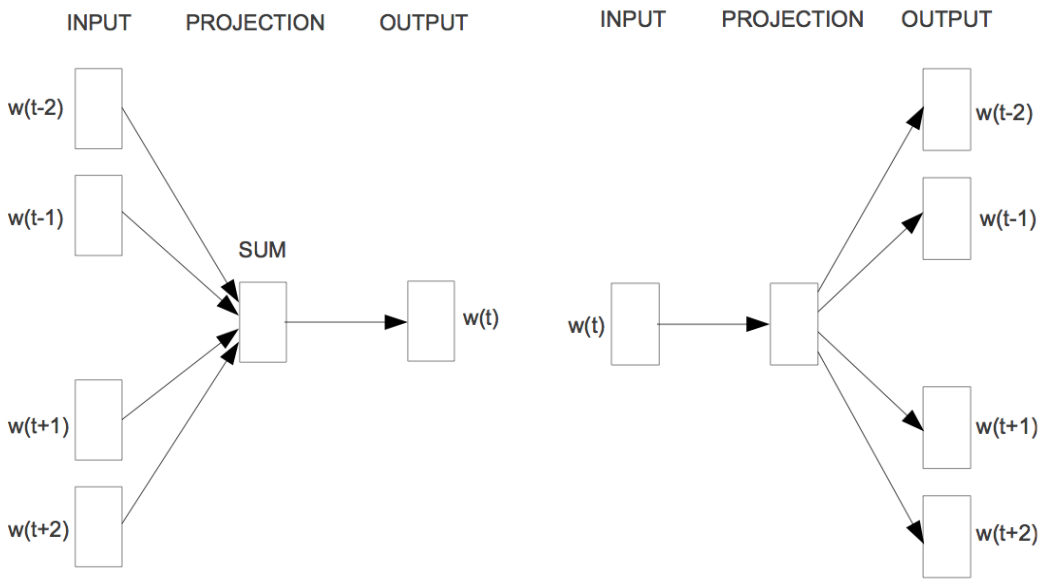
Word2Vec, which is considered as a semantic learning framework, uses a shallow neural network model to learn the representations of words/phrases in a particular text document. The interesting point about Word2Vec is that this model applies a neural network consisting of an input layer, a projection layer, and an output layer to understand the semantic meaning behind terms.

2.4.1 Word2Vec

The Word2Vec model, which is recognized as one of the most popular and extensively used word embedding techniques, has recently attracted significant attentions from different communities, including machine learning and the Semantic Web. It offers a computationally efficient way, based on a neural network model to learn word embedding from raw text [79]. The intuition behind the Word2Vec model focuses on training a network to predict neighboring words. Additionally, the most interesting property of Word2Vec is its ability to carry semantic meanings, which are beneficial in a wide range of data applications, ranging from semantic data integration to NLP. There are two architectures available based on the Word2Vec model (Figure 2.4), including *Continuous Bag of Words (CBOW)* and *Skip-Gram*. In the *CBOW* architecture, the Word2Vec model predicts a word given surrounding words while *Skip-Gram* receives a word as an input and estimates the surrounding words of that word (we utilize the skip-gram technique in Chapter 6).

Skip-Gram model: Given a sequence of training words w_1,w_2, w_3,..., w_T, and a context window, c, The skip-gram model attempts to predict the surrounding words or similar words to an input word. This process is completed through maximizing the following average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leqslant j\leqslant c, c\neq 0} logp(w_{t+j}|w_t) \tag{2.4}$$



(a) CBOW Architecture.

(b) Skip-Gram Architecture.

Figure 2.4: Word2Vec Architecture [63, 62]

where:

$$p(w_{t+c}|w_t) = \frac{exp(v\prime_{w_{t+c}}^T v_{w_t})}{\sum_{v=1}^{V} exp(v\prime_{w_v}^T v_{w_t})} \tag{2.5}$$

where V is the complete vocabulary of words, v'_w is the output vector of the word w, and $\bar{v}$ is the averaged input vector of all the context words calculated as follow:

$$\bar{v} = \frac{1}{2c} \sum_{-c \leqslant j \leqslant c, c \neq 0} p(w_t|w_{j-c}, \cdots, w_{t+c}) \tag{2.6}$$

2.5 Named Entity Recognition

Named Entities are usually defined as nouns or noun phrases that refer to particular types of entities, such as persons, organizations, locations, and so on. Named Entity Recognition (NER), which is one of the core components in Natural Language Processing, aims to identify, extract and classify named entities in a text. Spotting named-entities in text is an important task in different areas, such as information retrieval, summarization, question answering, and machine translation. We utilize this technique to retrieve entities in literals, which is described in Section 6.5.1.

Chapter 3

Related Work

This chapter focuses primarily on the most important existing RDF/S exploration techniques with respect to ontology summarization and RDF dataset profiling. Considering ontology summarization, various techniques have been developed to distill knowledge from a given ontology in order to produce an abridged version of that [101]. The proposed methods utilize mainly the common measures available in graph theory which aim to identify the most important part(s) of a graph [73, 76]. In general, different approaches highlight different aspects of ontology summarization such as diversity, centrality, and coverage in order to come up with a new model for ontology summarization. However, they attempt to generate a concise and coherent summary to convey enough information for an adequate understanding and provide and extensive coverage respectively.

RDF datasets profiling approaches, on the other hand, aim to facilitate data integration and consumption with statistics and meta-data about the content of RDF datasets. In the literatures, RDF dataset profiling is considered as the task

of providing insights through the data such as statistics about value distribution and of finding and extracting information patterns in the data. RDF dataset profiles are usually expected to represent the importance of datasets without any extra needs for detailed inspection of the raw data.

3.1 Ontology Summarization

In the literature, ontology summarization is referred to as an extractive summarization approach, in which the important terminological concepts (at the schema layer, or the TBox) and the important entities (at the data layer, or the ABox) are extracted to represent a summary of an ontology. Based on different measures, various methods of ontology summarization have been proposed, which utilize various criteria of generating a summary for a given ontology. In this chapter, we focus on the cutting-edge techniques of ontology summarization at the data layer, often referred to as the *Entity Summarization*. Advancements of ontology summarization at the schema layer will be discussed in Chapter 4.

3.1.1 Entity Summarization

Summarization, in general, is considered as one of the main approaches to making the information more readily available. Researchers in different communities have taken a strong interest in this task and, accordingly, have proposed various methods for a wide variety of summarization techniques in multiple areas. Document summarization [66], database summarization [23], and graph summarization [65] are just a few examples of techniques that have been studied by different commu-

nities. RDF data summarization, and in particular *Entity summarization*, as a task of producing a abridged, but still sufficient entity descriptions, has attracted considerable attentions as a way to absorb and distill descriptive knowledge from RDF datasets. Many tasks such as semantic data integration [30] and natural language processing such as entity disambiguation [31], and many others can benefit from entity summarization.

RELIN, proposed by Cheng et al. [29], utilizes relatedness and informativeness-based centrality to weight features that are expressed by predicate-object pairs of entities. The PageRank algorithm is used to rank individual features and ultimately extract representative triples, called representative features, for RDF graph entities. RELIN highlights the most similar and central triples, while in summarization, keeping the diversity of summarized triples is the key point. The SUMMARUM model [87], which also uses the PageRank algorithm to rank triples according to the popularity, utilizes the Wikipedia pages for a better navigation within Linked Data through the ranking of triples. The two aforementioned approaches could not meet the diversity requirement in the summarization process.

FACES [43], on the other hand, aims to incorporate the diversity in the selected triples for each entity. Partitioning the feature set and ranking the partitioned features are two primary steps in the FACES model. The main idea behind this model is to generate semantically diverse clusters, called "facets", from a given entity using an adaptation of the COBWEB algorithm [36]. The triples within each clusters is ranked using tf-idf-related popularity measure on the subject.

LinkSUM [86], the recent version of SUMMARUM, focused primarily on the objects instead of the diversity of properties for entities and showed a better result

on the same dataset, in comparison with FACES. Both SUMMARUM and FACES also discount literals in entity summarization. The extended version of FACES, FACES-E [42], and RELIN consider literals as good candidates for use in entity summarization. FACES-E gleans types for literals in RDF triples and uses this technique to be able to employ literals in entity summarization, while RELIN computes summaries for object and datatype properties (literals). The modified version of RELIN, called RELINM, also uses literals in the entity summarization [42].

In addition to the aforementioned models dedicated to entity summarization, variety of ranking models and tools, including TripleRank [37] and TRank [88] that aim to rank triples and concepts, respectively, incorporating ranking algorithms. However, Cheng et al. [29] indicated that these methods are not appropriate for the entity summarization problem, which needs ranking of feature sets based on their importance to identify the underlying entity.

3.2 RDF Dataset Profiling

As LOD datasets, and in particular RDF datasets, vary with respect to different features, such as statistics, quality, dynamics, etc., discovering reliable information with related to these features is essential in most applications. There exist wide variety of approaches and tools which aim to automatically extract statistics and descriptive information from RDF datasets. They focus mainly on different aspects such as statistical, topical and so on, in order to generate a profile to describe and understand an RDF dataset.

Assaf et al.[7] proposed *Roomba* as a framework to automatically generate, validate, and enrich descriptive dataset profiles in four main categories including general, access, ownership or provenance.

RDFStats [53] proposed by Langegger et al. aims to generate statistics such as entity counts (per class) and histograms (per class, property, value type) for RDF datasets.

The *ExpLOD* [52] tool utilizes the metadata about the structure of an RDF dataset (set of used RDF classes or properties) in order to summaries a dataset. The metadata is augmented with other information such as number of instances per class or the number of used properties.

Mäkelä et. al proposed *Aether* [1] [58] as a web application which is utilized to generate, visualize, and compare extended VOID statistical descriptions of RDF datasets. The generated statistical descriptions includes triples, entities, and statistics that are related to both triples and entities.

LODStats [9] is a statement-stream-based tool which can be used to gathering 32 comprehensive different statistical criteria for datasets. These statistics mainly represent the dataset in both data (instance) and schema layers using Vocabulary of Interlinked Datasets (VOID)[2] [3] and Data Cube Vocabulary[3]. It primarily covers triple frequencies, triples with blank nodes, average length of literals, labeled subjects, class and property usage, class hierarchy depth, cardinalities, and others.

[1] http://demo.seco.tkk.fi/aether/
[2] https://www.w3.org/TR/void/
[3] https://www.w3.org/TR/vocab-data-cube/

Abedjan et al. proposed *ProLOD* [1] as a web-based tool which analyzes and visualizes datasets in order to generate statistics such as data type and patterns distribution upon them. Mining and cleansing datasets are two other available options in the extended version of *ProLOD*, called *ProLOD++*[4], which enables it to generate a profile based on key analysis components such as frequencies, distribution of subjects, predicates, and objects.

Although the existing works are primarily focused on different aspects of RDF datasets at the schema and data layers, none of them have provided topic-wise RDF dataset profiling using knowledge based topic modeling techniques.

[4]https://www.hpi.uni-potsdam.de/naumann/sites/prolod++/

Chapter 4

Graph-based Ontology Summarization: A Survey [1]

[1] Seyedamin Pouriyeh, Mehdi Allahyari, Qingxia Liu, Gong Cheng, Hamid Reza Arabnia, Maurizio Atzori, Krys Kochut, "Graph-based Ontology Summarization: A Survey". Submitted to IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE 2018).

Abstract

Ontologies have been widely used in numerous and varied applications, e.g., to support data modeling, information integration, and knowledge management. With the increasing size of ontologies, ontology understanding, which is playing an important role in different tasks, is becoming more difficult. Consequently, ontology summarization, as a way to distill key information from an ontology and generate an abridged version to facilitate a better understanding, is getting growing attention. In this survey paper, we review existing ontology summarization techniques and focus mainly on graph-based methods, which represent an ontology as a graph and apply centrality-based and other measures to identify the most important elements of an ontology as its summary. After analyzing their strengths and weaknesses, we highlight a few potential directions for future research.

4.1 Introduction

An *ontology* provides an explicit specification of a vocabulary for a shared domain [41]. *Terms* in that vocabulary are mainly classes and properties denoting concepts and their relationships in the domain, respectively, forming a conceptualization of the world that we wish to represent for some purpose. In an ontology, the interpretation and use of terms are constrained by formal *axioms*. As ontologies can help people and organizations reach consensus on conceptualizations, they have found wide application in knowledge management, information integration, data access, etc. In particular, they play an important role in the recent explosive growth of Semantic Web deployment, where an ontology is frequently used as the schema of a knowledge base.

With the dramatic growth in both size and complexity of ontologies, their comprehension, exploration, and exploitation are becoming increasingly difficult. Summarization, in order to generate an overview or a preview of an ontology, is one possible solution that has received increasing research attention, recently. *Ontology summarization* is defined as a technique of distilling key information from an ontology in order to produce an abridged version for different tasks [101]. The output is a compact ontology summary, for a better and quicker understanding of an ontology, which can facilitate and reduce the cost of the next tasks in various applications such as ontology evaluation [21], matching [82], and search.

Compared with an early literature review [57], we have witnessed the emergence of many ontology summarization techniques, in recent years. In this survey paper, rather than providing a comprehensive bibliography, we mainly sort, re-

view, and compare various *graph-based methods* for ontology summarization. An ontology can be transformed into different graph models to represent the relations between terms and/or axioms. A broad range of measures have been presented to assess the importance of each node, which can be a term or an axiom. A subset of top-ranked nodes form an ontology summary, so the output of an ontology summarization approach is usually *a list of ranked terms or axioms*. Some approaches further choose paths to connect selected nodes and return *a subgraph*.

Table 4.1 summarizes the methods that will be reviewed in this paper. We will first compare different graph models, and then discuss measures for assessing node importance including centrality-based, coverage-based, and others. Finally, we conclude the paper with future directions. Note that our survey focuses on the summarization of terminological definitions in ontologies (i.e., TBox). Methods for summarizing instance data in knowledge bases (i.e., ABox), e.g., [34], will not be addressed.

4.2 Graph Models

An ontology provides definitions (i.e., axioms) for a set of terms. To represent the relations between terms and/or axioms, various graph models have been developed. In this section we review, illustrate, and compare those models.

4.2.1 RDF Graph

An ontology encoded in RDFS or OWL, which are languages recommended by W3C, can be transformed into an *RDF graph* as illustrated in Fig. 4.1. Each

Table 4.1: Ontology Summarization Methods

	Output	**Graph Model**	**Centrality**	**Other Measures**
[102]	ranked terms	vocabulary dependency graph	EC	TC
[92]	ranked terms	class graph	DC, BC, EC	-
[101]	ranked axioms	RDF sentence graph	DC, BC, EC	Di
[70]	ranked axioms	RDF sentence graph	DC	QR
[71]	ranked terms	class graph	DC, PC	Co, NS, Po
[98]	ranked terms	class graph	EC	-
[100]	ranked axioms	term-sentence graph	EC	Di, Po
[26]	ranked terms	class graph	EC	-
[72]	subgraph	class graph	DC	FC
[27]	ranked axioms	term-sentence graph	EC	QR, Ch
[55]	ranked terms	class graph	DC	-
[39]	subgraph	vocabulary dependency graph	-	QR
[77]	subgraph	class graph	DC, CC	-
[89]	subgraph	class graph	RC	-
[24]	ranked terms	class graph	EC	QR
[67]	subgraph	class graph	DC, BC, EgC BrC, HC, Ra	-
[90]	subgraph	class graph	RC	-

node-edge-node triple in the graph is called an RDF triple. In this example ontology, three classes and two properties are described by five axioms which are distinguished by different line styles in the figure.

RDFS is an extension of RDF; it is straightforward to represent an RDFS ontology as a graph. In such a graph, all the terms defined in an ontology are represented by nodes. Nodes are connected by directed arcs representing relations between two classes (e.g., `rdfs:subClassOf`), between two properties (e.g., `rdfs:subPropertyOf`), or between a property and a class (e.g., `rdfs:domain`, `rdfs:range`).

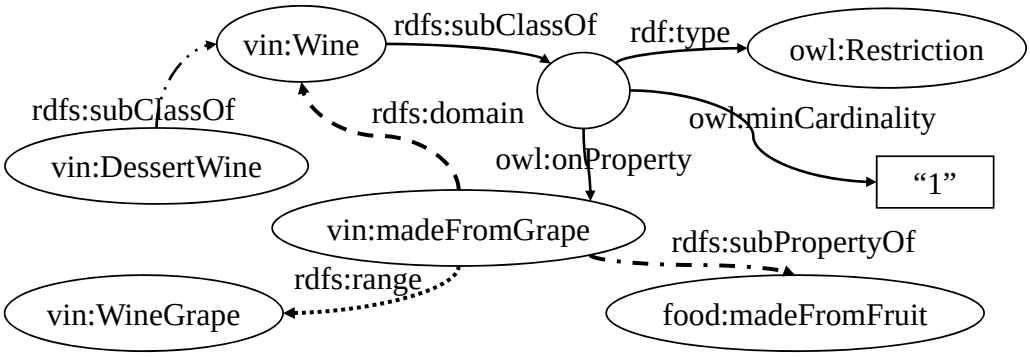


Figure 4.1: An example RDF Graph.

For OWL, W3C provides a document (as part of the OWL language) that defines the mapping of OWL ontologies into RDF graphs. OWL is more expressive than RDFS, and allows complex term definitions. Some axioms, e.g., `owl:Restriction`, which involves multiple terms, are transformed into multiple RDF triples connected by blank nodes.

Comments. As a “standard” graph representation of ontology, RDF graphs have rich tool support. They can be easily processed, stored, queried, and exchanged. However, in many cases an RDF graph representation of an ontology appears unnatural from the semantics point of view.

4.2.2 Class Graph

In order to directly represent semantic relations between classes, Wu *et al.* [98] presented a graph model where nodes represent classes and directed arcs represent binary relations between classes, which we call a *class graph*. Figure 4.2

illustrates a class graph for the ontology in Fig. 4.1. Note that some axioms (e.g., `owl:Restriction`) are not covered by this graph representation.

As to the relations between classes, if we only allow `rdfs:subClassOf`, the resulting graph will be a class hierarchy representing subsumption relations, as considered in [71]. More generally, a relation can also be a property defined in the ontology, connecting from its domain (which is a class) to its range (also a class).

Comments. Class graphs are close to human cognition. As classes are first-class citizens, class graphs are particularly suitable for approaches to ranking classes. However, the expressivity of class graph is limited. It well supports binary relations between classes but not more complex axioms involving multiple classes, e.g., `owl:unionOf`.

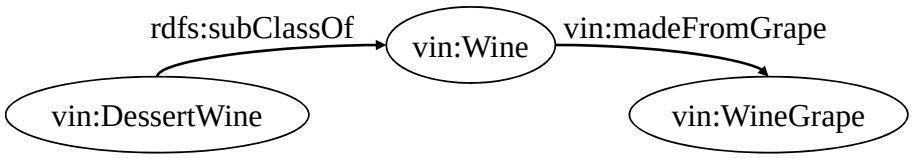


Figure 4.2: An example class graph.

4.2.3 RDF Sentence Graph

Zhang *et al.* [101] proposed an *RDF sentence graph*. An RDF sentence is a subset of RDF triples, and a set of RDF sentences form the finest partition of the triples in an RDF graph such that each blank node only appears in one block. In many cases, an RDF sentence corresponds to an axiom in OWL, since when mapping OWL ontologies into RDF graphs, blank nodes are introduced when an axiom is transformed into multiple RDF triples.

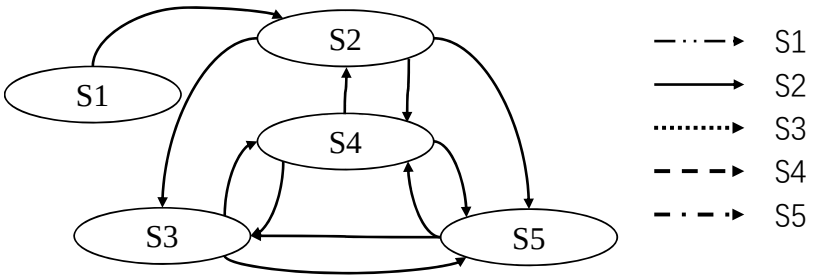


Figure 4.3: An example RDF sentence graph derived from Fig. 4.1, where each RDF sentence corresponds to a subset of the RDF triples in Fig. 4.1 that have a particular line style.

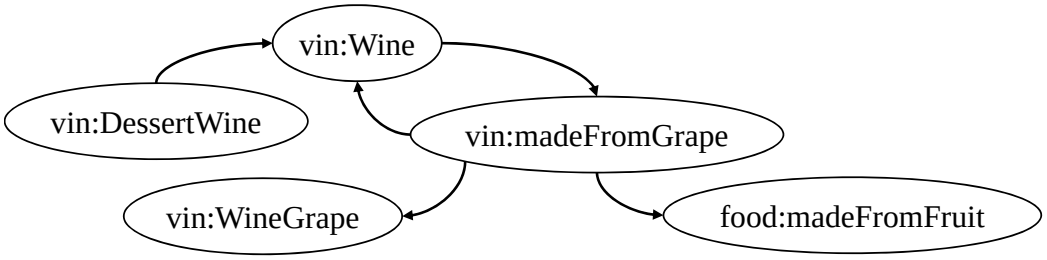


Figure 4.4: An example vocabulary dependency graph.

In an RDF sentence graph, nodes represent RDF sentences, which are adjacent if the terms they describe overlap. Figure 4.3 illustrates an RDF sentence graph for the ontology in Fig. 4.1; the five RDF sentences exactly correspond to five axioms. Zhang *et al.* [101] differentiate between two types of arcs, depending on the structural role of the shared terms, which we will not elaborate. Penin *et al.* [70] further cluster textually similar RDF sentences into topic nodes.

Comments. Compared with RDF triples, there is a better correspondence between RDF sentences and OWL axioms. In an RDF sentence graph, RDF sentences (or roughly speaking, axioms) are first-class citizens, making this model

particularly suitable for ranking triples/axioms. However, terms are not explicitly represented in this model, which may limit its application.

4.2.4 Vocabulary Dependency Graph

Based on RDF sentences, Zhang *et al.* [102] propose *vocabulary dependency graph*, where nodes represent terms, and edges connect terms that co-occur in an RDF sentence. Co-occurrence in an RDF sentence indicates dependency between terms. Figure 4.4 illustrates a vocabulary dependence graph for the ontology in Fig. 4.1, derived from Fig. 4.3. Compared with the class graph in Fig. 4.2, this new graph covers more terms (e.g., properties), though the edges are unlabeled. Essentially, in a vocabulary dependence graph, each axiom (represented by an RDF sentence) as a complex relation over multiple terms is decomposed into multiple binary relations.

Comments. Compared with the a sentence graph, a vocabulary dependence graph explicitly represents terms in the model, thereby being suitable for ranking terms. Compared with a class graph, a vocabulary dependence graph has both classes and properties as nodes, being suitable for ranking both of them. However, the meaning of an edge in a vocabulary dependence graph is not as explicit as in a class graph.

4.2.5 Term-Sentence Graph

Zhang *et al.* [100] present a bipartite graph model, where terms and RDF sentences are both represented by nodes, which we call a *term-sentence graph*. A directed

arc connects an RDF sentence to a term if the term is described in that RDF sentence. Figure 4.5 illustrates a term-sentence graph for the ontology in Fig. 4.1, derived from Fig. 4.3. Zhang *et al.* [100] differentiate between three types of arcs, depending on the structural role of term in RDF sentence, which we will not elaborate. The model is simplified in [27], where edges are undirected and unlabeled.

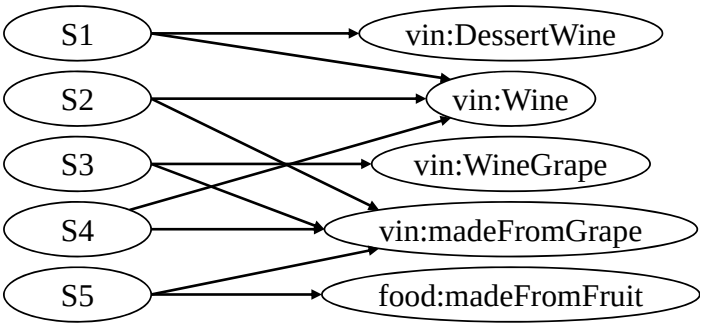


Figure 4.5: An example term-sentence graph.

Comments. A term-sentence graph is more complex than all the above-mentioned models. One advantage is that, compared with an RDF sentence graph and a vocabulary dependence graph, it explicitly represents both the terms and RDF sentences in the model, thereby expanding its potential application.

4.3 Assessment Measures

In a graph model, a broad range of node importance meanings in the context of ontology summarization has led to many different algorithms. In this section we primarily review popular centrality-based measures. We also discuss coverage-based, application-specific, and other measures.

4.3.1 Centrality-based Measures

Centrality-based measures are used to find topologically important nodes in a graph representation of an ontology. In general, centrality-based measures are defined via the available structure of the elements of a graph including nodes and edges. These measures primarily focus on the quantitative properties of graph structure such as number of edges and position of nodes, to assess the importance of a node. Some measures take edge types into consideration. As different centrality measures highlight different topological properties of a graph, their outputs are usually not consistent.

Degree Centrality (DC)

As one of the simplest centrality measures, *degree centrality* calculates the number of edges incident to a node v:

$$\mathrm{DC}(v) = |\text{Number of edges incident to } v| \,. \tag{4.1}$$

Pappas *et al.* [67] use this measure on a class graph to assess the local centrality of each class as its importance. The degree of a class indicates the richness of its description. Nodes with higher degree centrality are more important.

For a directed graph, degree centrality is divided into two categories: *in-degree centrality* and *out-degree centrality*, used in [101, 77]. The former counts the number of incoming arcs, and the latter counts the number of outgoing arcs.

Instead of considering all the edges incident to v, we may also count only those of specific types. More generally, different types of edges can be assigned different

weights, to measure *weighted degree*. For example, Peroni *et al.* [71] define the *density* of a class v as the weighted sum of its number of subclasses, properties, and instances:

$$\begin{aligned} \text{Density}(v) = w_S * &\text{ Number of subclasses of } v \\ + w_P * &\text{ Number of properties of } v \\ + w_I * &\text{ Number of instances of } v\,, \end{aligned} \tag{4.2}$$

where w_S, w_P, w_I are weights. Similar methods have been used in [92, 55]. Pirez *et al.* [72] and Queiroz-Sousa *et al.* [77] divide edges by their types into standard (e.g., is-a, part-of, same-as) and user-defined, which are weighted separately.

Relative Cardinality (RC)

Whereas in the above approaches weights are empirically configured, we highlight *relative cardinality* [89, 90], which is a way of automatically weighting edges for calculating weighted degree. In a class graph, the *cardinality* of an edge which represents a property connecting two classes is the number of the corresponding instances of the classes connected with that specific type of property. Therefore, classes and properties having more instances in a knowledge base are considered more important.

Comments on Degree Centrality Degree centrality and its variants (e.g., relative cardinality) can be efficiently computed in linear time, which is important when an ontology is very large. However, to assess the importance of a node, these measures mainly use its local information, i.e., the subgraph surrounding

that node. Without exploiting the global graph structure, the effectiveness of these measures is limited.

Path-based Centrality (PC)

Path-based centrality calculates the number of paths that pass through a particular node. For example, Peroni *et al.* [71] count the number of root-leaf paths in a class hierarchy that pass through each class v as its importance:

$$\mathrm{PC}(v) = |\text{Number of root-leaf paths passing through } v| \,. \tag{4.3}$$

A class in the middle of many root-leaf paths is central.

Betweenness Centrality (BC)

As a special case of path-based centrality, it makes sense to only consider *shortest paths*. Specifically, *betweenness centrality* is defined as the number of shortest paths from all nodes in a graph to all other nodes that pass through that node. Tzitzikas *et al.* [92] use the following implementation of betweenness to assess the importance of each node v in a class graph:

$$\mathrm{BC}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \,, \tag{4.4}$$

where σ_{st} is the total number of shortest paths from node s to node t in the graph, and $\sigma_{st}(v)$ is the total number of those paths passing through node v. The same

as degree centrality, a node with a higher betweenness value is considered more important. Betweenness has also been used on RDF sentence graph [101].

Ego Centrality (EgC)

Alternatively, for each node v, let G_v be the subgraph induced by v and its neighbors, which contains all the edges between them. Pappas *et al.* [67] calculate the betweenness centrality of v within G_v, which is called *ego centrality*:

$$\mathrm{EgC}(v) = \mathrm{BC}(v) \text{ calculated within } G_v\,. \tag{4.5}$$

Bridging Centrality (BrC)

As an improvement to betweenness, Pappas *et al.* [67] presented *bridging centrality*. A node with a high bridging centrality is one that connects densely connected components in a graph. To measure that, the bridging centrality of a node v is defined as the product of v's betweenness centrality (BC) and v's bridging coefficient (Br):

$$\begin{aligned} \mathrm{BrC}(v) &= \mathrm{BC}(v) \cdot \mathrm{Br}(v) \\ \text{where } \mathrm{Br}(v) &= \frac{\mathrm{DC}(v)^{-1}}{\sum_{u \in N(v)} \mathrm{DC}(u)^{-1}}\,, \end{aligned} \tag{4.6}$$

where $\mathrm{DC}(v)$ is the degree of node v and $N(v)$ is the set of v's neighbors. Betweenness centrality and bridging coefficient characterize global and local features of a node, respectively.

Comments on Path-based Centrality Compared with degree centrality, path-based centrality and its variants (e.g., betweenness centrality, bridging centrality) exploit the global graph structure, going beyond the neighborhood of a node. However, it is computationally expensive to calculate betweenness, which involves calculating the shortest paths between all pairs of nodes in a graph.

Closeness Centrality (CC)

Similar to betweenness, *closeness centrality* is another measure for determining the importance of nodes on a global scale within a graph. A node is usually considered as a key node if it can quickly interact with all the other nodes in a graph, not only with its immediate neighbors. The closeness of a node v is originally defined as the average length of the shortest paths between v and all other nodes in a graph:

$$\mathrm{CC}(v) = \frac{n-1}{\sum_{u \neq v} d(v,u)}, \tag{4.7}$$

where $d(v,u)$ is the distance between v and u, i.e., the number of edges in the shortest path between them, and n is the number of nodes in the graph.

Closeness centrality is used in [77], where an improved implementation for assessing the importance of each class v in a class graph is proposed:

$$\mathrm{CC}(v) = \frac{\sum_{u \neq v} \frac{score(u)}{d(v,u)}}{\sum_{u \neq v} \frac{1}{d(v,u)}}, \tag{4.8}$$

where $score(u)$ is the importance score of node u determined by some other measure. This new implementation gives emphasis on the classes that are close to other important classes.

Harmonic Centrality (HC)

We have seen several minor modifications made to the definition of closeness. Pappas *et al.* [67] present *harmonic centrality*, in which the average distance is replaced by the harmonic mean of all distances:

$$\mathrm{HC}(v) = \frac{1}{\sum_{u \neq v} d(v,u)}\,. \tag{4.9}$$

Radiality (Ra)

Pappas *et al.* [67] also present *radiality*, which takes the diameter of a graph into account:

$$\mathrm{Ra}(v) = \frac{1}{\sum_{u \neq v} \left(D - d(v,u)^{-1}\right)}\,, \tag{4.10}$$

where D is the diameter of the graph, namely the greatest distance between any pair of nodes in the graph.

Comments on Closeness Centrality Closeness centrality and its variants (e.g., harmonic centrality, radiality) are similar to betweenness, also involving calculating the shortest paths between all pairs of nodes in a graph. One difference is that, a node with a high closeness value is usually located at the center of the graph (in terms of distance), but such a node may not have a high betweenness

value because it may not be a bridging node that resides in many shortest paths connecting other nodes.

Eigenvector Centrality (EC)

A widely adopted principle is that a node is important if it is connected with important nodes. For example, in a class graph, a class is important if the classes it connects with are important. This gives rise to *eigenvector centrality* which iteratively calculates the importance of each node v in a graph:

$$\mathrm{EC}(v) = \frac{1}{\lambda} \sum_{u \in N(v)} \mathrm{EC}(u)\,, \tag{4.11}$$

where $N(v)$ is the set of v's neighbors, and λ is a constant factor for normalization. The eigenvector centrality of a node is the sum of the eigenvector centrality of its neighbors. The computation iterates over all the nodes in the graph, one round after another until convergence.

Whereas this basic measure has been used in [26], its improved variants are more popular in the literature. PageRank, a well-known implementation of eigenvector centrality, is used in [92, 24]. Different from the above basic measure, PageRank introduces a damping factor which is added to the centrality. Weighted PageRank, weighted HITS, or their variants are used in [102, 101, 100, 98, 27], where centrality is defined as a weighted sum. The weight of an edge between v and u indicates the strength of the connection between them; a stronger connection will transport more centrality score from u to v.

Comments on Eigenvector Centrality Eigenvector centrality and its variants (e.g., PageRank, HITS) have shown their effectiveness in many applications. However, they require iterative computation over all the nodes in a graph until convergence, which is time-consuming for large graphs.

Empirical Comparison of Centrality-based Measures

It seems that the effectiveness of a centrality-based measure is related to the graph model, and may also depend on the specific ontology to be summarized as the application and the domain of an ontology provide a guideline in order to select a proper set of measures.

Specifically, according to the experiment results presented in [92], the simple degree centrality (DC) appears more effective than PageRank (i.e., EC) on some class graphs. However, Zhang *et al.* [101] report that weighted PageRank (i.e., EC) outperforms degree (i.e., DC) on several RDF sentence graphs; both of them are considerably better than betweenness (i.e., BC). Pappas *et al.* [67] find that degree (i.e., DC) and betweenness (i.e., BC, EgC, and BrC) are notably better than closeness (i.e., HC and RA) on a few class graphs.

Unfortunately, we could not draw any reliable conclusions from the current empirical results reported in the literature as they all experiment with a small number of ontologies.

4.3.2 Coverage-based Measures

Top-ranked nodes in a graph representation of an ontology may not form the best ontology summary. For many applications, a good summary is expected to have a good coverage of the contents of an ontology, to form a comprehensive and unbiased overview. Accordingly, the quality of a subset of nodes forming a summary is to be assessed as a whole.

Coverage (Co)

Peroni *et al.* [71] propose the *coverage* criterion which aims to show how well the selected set of classes are spread over the whole class hierarchy. For each node v, let $N^+(v)$ be the set of nodes covered by v, including v and its neighbors, i.e., its subclasses and superclasses in the class hierarchy. The coverage of a set of selected nodes V is defined as the proportion of nodes in the graph that are covered by V:

$$\mathrm{Co}(V) = \frac{|\bigcup_{v \in V} N^+(v)|}{n}\,, \tag{4.12}$$

where n is the number of nodes in the graph.

Further, Peroni *et al.* [71] consider an interesting measure called *balance* which is directly related to coverage. It measures how balanced the selected nodes are, i.e., the degree to which each selected node contributes to the overall coverage of the set, which is characterized by standard deviation.

Diversity-based Re-ranking (Di)

In [101, 100], the coverage of a summary is improved by a *re-ranking* step after centrality-based ranking. In these approaches, nodes are iteratively selected to form a summary. In each iteration, the next node to be selected may not be the top-ranked one among the remaining nodes, which will be re-ranked such that a node similar to those selected in previous iterations will be penalized. Specifically, let $score(v)$ be the centrality score of node v, and let $sim(v, u)$ be the similarity between nodes v and u. Given a set of nodes V_s which are already selected into the summary and a set of candidate nodes V_c, the next node to be selected from V_c is

$$\arg\max_{v \in V_c} \left(score(v) - \sum_{u \in V_s} sim(v, u) \right). \tag{4.13}$$

Zhang *et al.* [101, 100] use this algorithm to rank RDF sentences, where two RDF sentences are similar if they share terms. The resulting ontology summary is diversified with regard to the terms it contains.

Comments on Coverage-based Measures Coverage-based methods complement centrality-based measures, but their current implementations are suboptimal. Coverage in Eq. (4.12) considers the neighborhood of each node, not taking the global graph structure into account. Diversity-based re-ranking in Eq. (4.13) has a greedy nature, and may not find the optimum summary in terms of centrality and diversity.

4.3.3 Application-specific Measures

The following two methods are not graph-based but are designed for specific applications.

Query Relevance (QR)

A special kind of ontology summary is a snippet presented in search results pages of an ontology search engine. In this application, terms [39, 24] or RDF sentences [27, 70] that are *relevant to a user query* (e.g., containing query keywords) are prioritized for being presented in a snippet, to show the relevance of an ontology to the user's information needs.

Frequency of Correspondences (FC)

Pires *et al.* [72] consider applications where an ontology to be summarized can be an integrated ontology obtained by merging several local ontologies. In that case, an important term in the integrated ontology is one that has a high *frequency of correspondences*, namely it finds correspondences to many classes in local ontologies.

4.3.4 Other Measures

In addition to graph-based and application-specific measures, we briefly review other methods used in the literature.

Name Simplicity (NS)

Peroni *et al.* [71] emphasize that *natural categories* or *basic objects* are good representers of an ontology. They propose that a natural category normally has a relatively simple label, and hence they assess the importance of a class by the *simplicity of its name*. A class having compound words in the name will be penalized.

Textual Centrality (TC)

Zhang *et al.* [102] calculate the *textual centrality* of a term in an ontology. Different from the centrality-based measures discussed in Section 4.3.1 which are defined over graph structure, the textual centrality of a term is the similarity between its textual description and the one for the whole ontology.

Popularity (Po)

The wide use of a term on the Web suggests its importance. To measure the *popularity* of a term, Peroni *et al.* [71] submit the name of the term as a keyword query to a Web search engine and resort to the number of returned results. Zhang *et al.* [100] calculate the number of websites hosting RDF documents where the term is instantiated.

Cohesion (Ch)

Cheng *et al.* [27] measure the quality of a summary as a whole. Different from diversity-based re-ranking described in Section 4.3.2 which penalizes an ontology

summary where RDF sentences share terms, such a summary will be awarded in [27] as it exhibits *cohesion*.

4.4 Future Directions

We have investigated different graph models and measures for ontology summarization. We believe that other directions to generate more reliable ontology summaries exist, and we are trying to address some of them to conclude our survey.

Although many algorithms for the ontology summarization problem have been proposed, empirical results reported in the literature suggest that none of them consistently generates the best ontology summary. In an ideal case, the ontology summarization technique needs to be more flexible in the way that users or applications are able to *tune* the model in order to generate different summaries based on different requirements or inputs. In other words, *dynamic or adaptive ontology summarization* can be viewed as an interesting topic to explore.

Defining *new measures*, either graph-based or not, is another research activity in the context of ontology summarization. Ideas may come from thorough investigations into human-made "gold-standard" summaries. Research advances in the field of information retrieval and text summarization, as well as recent research on entity summarization (e.g., [75, 74]) which is closely related to ontology summarization, can also provide inspiration. In particular, machine learning techniques have not been extensively used for ontology summarization.

The available approaches apply *extractive techniques* to generate the final summary. In the extractive scenario, a subset of the terms and/or axioms from the original input ontology are selected as a summary. *Non-extractive or abstractive* ontology summarization will be a new direction in this area. In that scenario, the key research question is how to define the output of ontology summarization, e.g., as some kind of high-level aggregate representation of terms and axioms.

There is a lack of *evaluation efforts*. To the best of our knowledge, experiments presented in the literature are all based on a small number of ontologies. No benchmark for ontology summarization is available so far.

Dozens of software systems, libraries, or APIs for text summarization are available, many of which are open-source. By comparison, it is rare to see any *software tool support* for summarizing ontologies. In fact, if such a tool or an application aims to directly serve ordinary users, it needs to also address the presentation (e.g., verbalization, visualization) of and the interaction with ontologies, in which some other challenges would emerge.

Last but not least, almost all of the methods we have discussed generate ontology summaries to be presented to human users. Summaries may also facilitate computer processing in certain tasks. It would be interesting to explore applications of this kind.

Chapter 5

ES-LDA: Entity Summarization using Knowledge-based Topic Modeling [1]

[1]Seyedamin Pouriyeh, Mehdi Allahyari, Krys Kochut, Gong Cheng, Hamid Reza Arabnia, "ES-LDA: Entity Summarization using Knowledge-based Topic Modeling", Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017). Reprinted here with permission of the publisher.

Abstract

With the advent of the Internet, the amount of Semantic Web documents that describe real-world entities and their inter-links as a set of statements have grown considerably. These descriptions are usually lengthy, which makes the utilization of the underlying entities a difficult task. Entity summarization, which aims to create summaries for real world entities, has gained increasing attention in recent years. In this paper, we propose a probabilistic topic model, ES-LDA, that combines prior knowledge with statistical learning techniques within a single framework to create more reliable and representative summaries for entities. We demonstrate the effectiveness of our approach by conducting extensive experiments and show that our model outperforms the state-of-the-art techniques and enhances the quality of the entity summaries.

5.1 Introduction

With the emergence of Linked Open Data (LOD)[2] as a way of publishing and interacting with the information, many datasets such as DBpedia [17] and YAGO [47] have been created and are publicly available on the Web. For example, DBpedia as part of LOD is a knowledge base extracted from Wikipedia that consists of Wikipedia resources (entities) described as RDF statements (i.e., RDF triples). The Resource Description Framework (RDF) is the Semantic Web standard data model used for representing information on the Web. An RDF triple is represented in the form of $< subject, predicate, object >$. The latest English version of DBpedia contains over 4.5 million entities collectively described by over 1.6 billion triples. This means that each entity description has an average of 355 RDF triples. Human users and computer applications need to consider these lengthy descriptions while performing various semantic tasks. Thus, *entity summarization*, a task of producing more concise, but still sufficient entity description, has garnered a significant amount of attention.

Recently, with the huge growth of information, summarization techniques are becoming some of the main approaches to making the information more readily available. In fact, summarization techniques aim to facilitate the identification of structure and meaning in data. Researchers in different communities have taken a strong interest in this task and, accordingly, have proposed various methods for a wide variety of summarization techniques in multiple areas. Document summarization [66], database summarization [23], and graph summarization [65] are

[2] `http://linkeddata.org`

just a few examples that have been studied by different communities. RDF data summarization and in particular entity summarization, has attracted considerable attentions in recent years as it can benefit many other tasks in the natural language processing area, including entity recognition [103], entity disambiguation [31], and many others. Several approaches have been developed to summarize RDF data with respect to entities, including RELIN [29], FACES [43], and LinkSUM [86]. RDF summarization differs from document summarization in the sense that RDF triples are structured and do not have many frequently used words to help the summarization task, which makes RDF summarization more challenging.

Topic modeling has become a popular method for uncovering the hidden themes from text corpora. Topic models usually consider each document as a mixture of topics, where a topic is a probability distribution over words. When the topic proportions of documents are estimated, they can be used as the themes (high-level semantics) of the documents. Topic models have been widely used for various text mining tasks, such as machine translation [84], word embedding [12, 32], automatic topic labeling [95, 4, 6], and others[5].

In this paper, we propose a novel topic model, called ES-LDA, that integrates prior knowledge with the topic modeling within a single framework for RDF entity summarization. In our approach, each entity, which is considered as a document, is a multinomial distribution over the predicates (properties), where each predicate is a probability distribution over the subjects and objects of the triples in the RDF data. We rank the triples based on their probability distributions and choose the top-k triples that best describe the underlying entity as its summary. We evaluated our approach against state-of-the-art techniques and our experiments

indicate that our approach outperforms other methods in terms of the quality of summarization.

The rest of the paper is organized as follows: Section 5.2 presents an overview of related work. Section 5.3 introduces the baseline for this paper. In Section 5.4, we define the main problem and propose our model in detail and afterwards, in Section 5.5, we explain the configurations of our model and describe the experiments. Finally, in Sections 5.6 and Section 5.7, we discuss the results and conclude the paper, respectively.

5.2 Related Work

Summarization methods can be divided into two main categories, which are called extractive and none-extractive (abstractive) summarization. In extractive approaches, which are usually applicable in text and ontology summarization [50] [101], a set of features is extracted directly from the input data. On the other hand, in non-extractive methods, which generally are employed in graph [65] and database [23] summarization, new sentences from the input data are generated [44] to form a summary. In this research, we focus on extractive summarization. The concept of entity summarization in the form of RDF graph data has attracted more attention in recent years. Cheng et al. [29] proposed entity summarization method, called RELIN, based on the PageRank algorithm to extract representative triples, called representative features for RDF graph entities. Because of the centrality based ranking issue, RELIN highlights the most similar and central triples, while in summarization, the diversity of summarized triples is the key

point.

SUMMARUM [87] is a system for a better navigation within Linked Data through the ranking of triples. This system also uses the PageRank algorithm to rank triples according to the popularity of resources with the help of Wikipedia pages. Two aforementioned approaches could not meet the diversity requirement in the summarization process. FACES [43], on the other hand, tries to keep a balance between the centrality and diversity of the selected triples for each entity. It utilizes a clustering algorithm, called Cobweb [36], to cluster related triples before ranking them to keep the diversity in the summarization. The recent version of SUMMARUM, which is called LinkSUM [86], focused more on the objects instead of the diversity of properties for entities and showed a better result on the same dataset, in comparison with FACES. Beside the aforementioned techniques dedicated to entity summarization, there are various ranking models and tools, including TripleRank [37] and TRank [88] that rank triples and concepts, respectively, incorporating ranking algorithms. However, Cheng et al. [29] indicated that these methods are not appropriate for the entity summarization problem, which needs ranking of feature sets based on their importance to identify the underlying entity.

5.3 Preliminaries

An RDF data graph is a collection of nodes and edges that connect the nodes together. Nodes are usually recognized by unique IDs which are called *Uniform Resource Identifiers (URIs)* or exact values (i.e. numbers, dates, etc) namely

Table 5.1: J.C.Penny entity predicates and corresponding objects with the top-5 ES-LDA summary.

Predicate	**Object**	**Top-5**
http://dbpedia.org/property/areaServed	http://dbpedia.org/resource/United_States	✗
http://dbpedia.org/ontology/foundedBy	http://dbpedia.org/resource/James_Cash_Penney	✓
http://dbpedia.org/property/founder	http://dbpedia.org/resource/James_Cash_Penney	✗
http://dbpedia.org/ontology/industry	http://dbpedia.org/resource/Retail	✓
http://dbpedia.org/property/keyPerson	http://dbpedia.org/resource/Ron_Johnson	✓
http://dbpedia.org/property/homepage	http://www.jcpenney.com/	✗
http://dbpedia.org/ontology/location	http://dbpedia.org/resource/Plano,_Texas	✓
http://dbpedia.org/ontology/regionServed	http://dbpedia.org/resource/United_States	✗
http://dbpedia.org/property/tradedAs	http://dbpedia.org/resource/S&P_500	✗
http://dbpedia.org/ontology/type	http://dbpedia.org/resource/Public_company	✓

Literals. An RDF graph is represented in a form of a collection of triples, each including a *Subject, Predicate, and Object*. In an RDF graph, an entity is defined as a subject with all predicates and corresponding objects to those predicates, collectively forming the entity's description. As Table 5.1 shows, the *J.C.Penny* entity is represented by its predicates (properties) and the corresponding objects in the triple format. For example, the triple $< J.C.Penny, industry, Retail >$ introduces *J.C.Penny*'s industry as *Retail* (due to space limitations we have dropped the first part of the *URIs*).

Definition 1 (Entity summary): Given an entity e and a positive integer k, a summary of the entity e, denoted *Sum(e, k)*, is the top-k subset of all predicates and corresponding objects that are most relevant to that entity. As Table5.1 shows the top-5 summary for *J.C.Penny* entity, which is represented through *foundedBy, industry, keyPerson, location, and type*.

5.3.1 Latent Dirichlet Allocation (LDA)

The Latent Dirichlet Allocation (LDA) is a generative probabilistic model for extracting thematic information (topics) from a collection of documents. LDA assumes that each document is made up of various topics, where each topic is a probability distribution over words.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ be a corpus of documents and $\mathcal{V} = \{w_1, w_2, \ldots, w_{|\mathcal{V}|}\}$ a vocabulary (words) of the corpus. A topic $z_j, 1 \leq j \leq K$ is represented as a multinomial probability distribution over the $|\mathcal{V}|$ words, $p(w_i|z_j), \sum_i^{|\mathcal{V}|} p(w_i|z_j) = 1$. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words, given the document, is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \tag{5.1}$$

The graphical model of LDA is shown in Figure 5.1 and the generative process for the corpus $\mathcal{D}$ is:

1. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \text{Dir}(\beta)$
2. For each document $d \in \{1, 2, \ldots, \mathcal{D}\}$,
   (a) Sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
   (b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
      i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
      ii. Sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

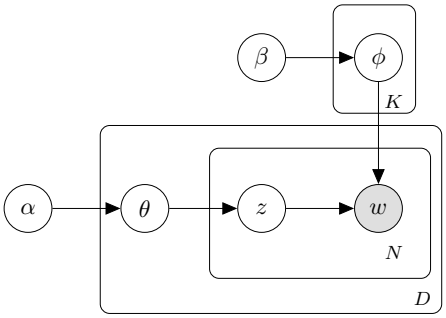


Figure 5.1: LDA Graphical Representation

In the LDA model, the word-topic distribution $p(w|z)$ and topic-document distribution $p(z|d)$ are learned entirely in an unsupervised manner, without any prior knowledge about what words are related to the topics and what topics are related to individual documents.

5.4 Problem Statement

In this section, we first describe the problem and then define how to utilize topic models for RDF graphs. Then, we formally introduce our ES-LDA model and explain how to integrate prior knowledge from RDF data graph within a topic model for entity summarization.

5.4.1 Problem Definition

Generating summaries for voluminous Semantic Web data, and in particular RDF data, for quick identification of entities has gained considerable attention as a

challenging problem in the Semantic Web community. In the literature, *Entity Summarization* is defined as selecting a small but representative subset of the original triples associated with an entity. In this context, given an RDF data set comprising a collection of entities, where each entity is described by a set of its properties (i.e., all triples with the entity as the subject), our goal is to choose *top-k* representative triples for each entity. In other words, since all triples associated with an entity (as its description) share the same subject, our objective is to select *top-k* predicates and their corresponding objects among these triples that best summarize the entity's description.

5.4.2 Topic Models for RDF Graphs

Topic models were originally introduced for text documents, however, they have been applied to other types of data, such as images [18], and recently [83] used topic modeling for RDF graphs. The first step in applying topic models is to define documents and word-like elements as the basic building blocks of documents. Since an RDF graph is usually represented as a set of triples, where each triple t consists of a subject s, predicate p, and an object o, in the form of $<s, p, o>$, we can consider a collection of such triples as a "document".

Definition 2 (document): A document d is defined as a set of triples, $d = \{t_1, t_2, \cdots, t_n\}$, that describe a single entity e. In other words, all triples of a document d have the same subject.

"Words" of a document can be extracted from different parts of its triples. We define a "**word**" w as the subject or object of a triple t in document d. Therefore,

each document is represented by a "bag of words" including all the subjects and objects of its triples. In this paper, all subjects in the triples of a document are the same, because each document corresponds to a single entity, hence, in practice each document is a "bag of objects"[3]

Topic models usually utilize some data preprocessing, such as punctuation removal, downcasting, and abbreviation expansion, etc., to enhance the final performance. We also performed preprocessing on the RDF data and filtered out the schema and dataset dependent predicates, such as *sameAs, wikiPageExternalLink, subject, wikiPageWikiLink*, in addition to *literals*. Since we work with RDF graphs that differ from typical text documents in the sense that RDF data are represented as triples, we need to address several challenges mentioned in [83] to be able to run topic models on RDF data. These challenges include sparseness, use of unnatural language, and the lack of context. RDF data can be affected by **Sparseness**. We consider documents as sets of triples associated with a single entity. Such a set can be very large, leading to a large bag of words with a semantic theme, or small (sparse), resulting in a poor bag of words with less contextual information. It is also possible that a document with a high number of triples ends up having a small bag of words after pre-processing; for example based on Table 5.1, *J.C.Penny* entity comes with *United_States, James_Cash_Penney, Retail, Ron_Johnson, Plano,_Texas, United_States, S&P_500 and Public_company* as a bag of words for *J.C.Penny* entity, which shows sparseness in this document. **Unnatural Language** can be problematic for RDF data. A typical text document contains sentences where each sentence has a natural structure. These extra

[3] "bag of words" and "bag of objects" are interchangeably used.

components of a sentence usually provide a further "**context**" for understanding words that are ambiguous or have multiple meanings, such as polysemous or homonymous ones. The aforementioned example for the *J.C.Penny* entity also confirms the unnatural language problem. The "**lack of context**" can further impact RDF data because they are potentially sparse, described by unnatural language, and often using words that have multiple meanings, difficult to differentiate (*J.C.Penny* bag of words example). Additionally, triples are more prone to pre-processing, because it is not uncommon for triples to contain unexpected characters. RDF data resemble short texts in terms of the aforementioned challenges. Sparseness in a short text causes the model to be less discriminative to recognize how words are related and the limited context makes it hard for the model to identify the meanings of the words in such short text documents [99]. In order to alleviate these issues, researchers usually take two approaches. They either augment the short text or design custom versions of the LDA model that address their specific problems. In this paper, we have used both approaches. We describe how to supplement the RDF data in the following section and describe the details of our model in Section 5.4.4.

5.4.3 Supplementing RDF Data

As topic modeling is based on statistics of the co-occurrence of terms [83], when we are dealing with short texts with a very limited number of repetitions, which is the case with RDF data, we need to find a way to supplement the data to elevate the performance of the topic modeling approach. We augment the documents

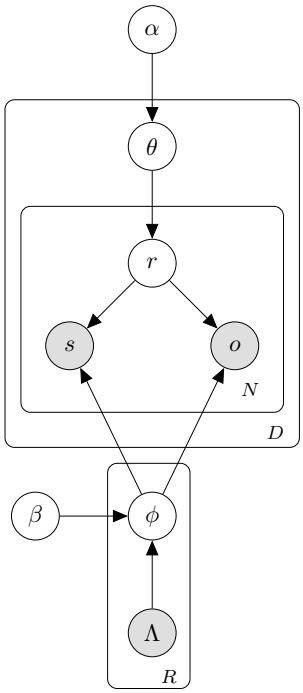


Figure 5.2: Entity Summarization Model

using two different methods. In the first method, we increase the frequency of the words in each document. But the question is "*How many times each word of a document should be repeated?*". Entities in DBpedia have been organized into a category network, therefore, every entity has a number of categories associated with it. The relationship between an entity and a category is defined by the "*http://purl.org/dc/terms/subject*" predicate. Since each word of a document is an object of a triple, and accordingly, an entity in DBpedia, it is related to several categories. We assume that objects (words) of a document that have more categories are likely more important. Thus, We expand each document

Algorithm 1: ES-LDA Model

foreach predicate $r \in \{1, 2, \ldots, R\}$ **do**
  Draw an object distribution $\phi_r \sim \text{Dir}(\beta_r \times \Lambda_r)$
end
foreach document $d \in \{1, 2, \ldots, D\}$ **do**
  Draw a predicate distribution $\theta_d \sim \text{Dir}(\alpha_d)$
  foreach subject s and object o of document d **do**
    Draw a predicate $r \sim \text{Mult}(\theta_d)$
    Draw a subject s from predicate $r, s \sim \text{Mult}(\phi_r)$
    Draw an object o from predicate $r, o \sim \text{Mult}(\phi_r)$
  end
end

by increasing the frequency of each object by the number of its categories. In the second method, instead of repeating each object a certain number of times, we enlarge each document by adding categories of the objects as extra words, directly to the document. There are multiple advantages of supplementing each document by adding object categories: (i) the sparseness in the document, related to each entity, is lowered as we are adding a number of related words to it; (ii) we reduce the ambiguity in the document, because adding extra categories alleviates the lack of context and helps distinguish the appropriate meanings of the words with multiple connotations; and lastly (iii), adding object categories makes the documents semantically more relevant to their topical themes. We evaluated our model using both methods and the results demonstrate that the first method gives significantly better summaries than the second method.

5.4.4 Proposed Model

ES-LDA is a probabilistic generative model for modeling entities in RDF graphs. The key idea behind our model is twofold: (1) we exploit statistical topic models as the underlying quantitative framework for entity summarization; and (2) ES-LDA incorporates the prior knowledge from the RDF knowledge base directly into the topic model. The plate notation is shown in Figure 5.2.

In our model, each document is a multinomial distribution over the predicates. If we consider predicates as topics, at the document level, our model is the same as standard LDA. However, we set the number of topics in ES-LDA to be the number of unique predicates in the corpus. Unlike the standard LDA, where each topic is a multinomial distribution over the vocabulary from the Dirichlet prior β, in our model each predicate is a multinomial distribution over all the subjects and objects of the RDF graph. In our approach, a document consists of a set of triples describing a single entity, i.e. all these triples share the same subject. Thus, we constrain the documents to only have the objects of related triples and also restrict the predicates to be defined only over the objects. In addition, for each predicate r, we further smooth its distribution by Λ_r. Λ is a matrix that has encoded the background knowledge about predicate-object values from DBpedia. Section 5.4.5 explains how Λ is constructed. The generative process of ES-LDA is shown in Algorithm 1.

Following this process, the joint probability of generating a corpus $D = \{d_1, d_2, \ldots, d_{|D|}\}$, the predicate assignments $\mathbf{r}$ given the hyperparameters α, β and

the prior matrix Λ is:

$$
\begin{aligned}
&P(\mathbf{o}, \mathbf{s}, \mathbf{r}|\alpha, \beta, \Lambda) \\
&= \int_{\phi} P(\phi|\beta; \Lambda) \prod_{d} \sum_{r_d} P(\mathbf{o_d}|r_d, \phi) P(\mathbf{s_d}|r_d, \phi) \\
&\times \int_{\theta} P(\theta|\alpha) P(\mathbf{r_d}|\theta, \phi) d\theta d\phi
\end{aligned} \tag{5.2}
$$

5.4.5 Constructing Predicate-Object Prior Matrix Λ

In the ES-LDA model, each predicate has a probability distribution over the objects of the RDF graph. Entity summarization is the task of choosing the top-k predicate-object pairs that best describe an entity. Presumably, if an object is associated with more categories in DBpedia, it is likely more important. We create the the Λ matrix to encode the prior weight of the predicate-object pairs and utilize it to smooth the predicate-object distributions ϕ by incorporating this domain knowledge into the topic model. We build the Λ matrix of size $R \times O$, where R is the number of predicates and O is the number of objects in the RDF graph. Let f be an indicator function where $f(i, j) = 1$ if there is a triple in RDF graph with predicate i and object j, and 0 otherwise, for $1 \leq i \leq R$ and $1 \leq j \leq O$. Additionally, let c be the number of categories assigned to object j. Then, we define Λ_{ij} as follows:

$$
\Lambda_{ij} = \begin{cases} c & \text{if } f(i, j) = 1 \\ 1 & \text{otherwise.} \end{cases} \tag{5.3}
$$

For example, the "*Barack_Obama*" entity has multiple predicate-object pairs in DBpedia, including "*profession-author*", "*profession-lawyer*" and "*profession-professor*" pairs. According to DBpedia, $c_{author} = 2$, $c_{lawyer} = 4$ and $c_{professor} = 2$. It is reasonable to expect a higher probability for the "*profession-lawyer*" pair as it seems to be slightly more important than the other two pairs for "*Barack_Obama*". As a result, $\Lambda_{profession-lawyer} = 4$, which promotes "*profession-lawyer*" in Eq. 5.5.

5.4.6 Inference using Gibbs Sampling

Since the posterior inference of the LDA is intractable, we need to find an algorithm for estimating the posterior inference. A variety of algorithms have been used to estimate the parameters of topic models, such as variational EM [19] and Gibbs sampling [40]. In this paper we use the collapsed Gibbs sampling procedure for our ES-LDA topic model. Collapsed Gibbs sampling [40] is a Markov Chain Monte Carlo (MCMC) [80] algorithm, which constructs a Markov chain over the latent variables in the model and converges to the posterior distribution, after a number of iterations. In our case, we aim to construct a Markov chain that converges to the posterior distribution over $\mathbf{r}$ conditioned on observed subjects $\mathbf{s}$, objects $\mathbf{o}$, hyperparameters α, β, and the prior matrix Λ.

In our modified version of the learning algorithm to infer $p(o_i|r_j)$ and $p(r_j|d)$, we (1) constrain the objects that are not paired with a predicate to have 0 probability, i.e. $p(o_i|r_j) = 0$, if $(r_i, o_j) \notin$ RDF graph, and (2) $P(s|r_j) = 1$, since all the triples of a document have the same subject s. We derive the posterior inference from Eq. 5.2 as follows:

$$
\begin{aligned}
P(\mathbf{r}|\mathbf{o},\mathbf{s},\alpha,\beta,\Lambda) &= \frac{P(\mathbf{r},\mathbf{o},\mathbf{s}|\alpha,\beta,\Lambda)}{P(\mathbf{o}|\alpha,\beta,\Lambda)} \\
&\propto P(\mathbf{r},\mathbf{o}|\alpha,\beta,\Lambda) \propto P(\mathbf{r})P(\mathbf{o}|\mathbf{r})P(\mathbf{s}|\mathbf{r})
\end{aligned} \tag{5.4}
$$

$$
\begin{aligned}
&P(r_i = r|o_i = o, \mathbf{r}_{-i}, \mathbf{o}_{-i}, \alpha, \beta, \Lambda) \propto \\
&\frac{n^{(d)}_{r,-i} + \alpha_r}{\sum_{r'} (n^{(d)}_{r',-i} + \alpha_{r'})} \times \frac{n^{(r)}_{o,-i} + \Lambda_{ro}\beta_o}{\sum_{o'} (n^{(r)}_{o',-i} + \Lambda_{ro}\beta_o)}
\end{aligned} \tag{5.5}
$$

where $n^{(r)}_o$ is the number of times object o is assigned to predicate r. $n^{(d)}_r$ denotes the number of times predicate r is associated with document d. The subscript $-i$ indicates that the contribution of the current object o_i being sampled is removed from the counts. After Gibbs sampling, we can use the sampled predicate to estimate the probability of a predicate, given a document, θ_{dr} and the probability of an object, given a predicate, ϕ_{ro}:

$$
\theta_{dr} = \frac{n^{(d)}_r + \alpha_r}{\sum_{r'} (n^{(d)}_{r'} + \alpha_{r'})} \tag{5.6}
$$

$$
\phi_{ro} = \frac{n^{(r)}_o + \Lambda_{ro}\beta_o}{\sum_{o'} (n^{(r)}_{o'} + \Lambda_{ro}\beta_o)} \tag{5.7}
$$

5.5 Experiments

We evaluated our ES-LDA model against the state-of-the-art LinkSUM [86] and FACES [43] systems. Our goal was to show that the ES-LDA model produces results that are closer to human judgment, in comparison with the other approaches.

We used the same dataset[4] that was used in the experiments conducted with FACES, as well as LinkSUM models. The dataset contained 50 entities randomly selected from DBpedia (English version 3.9) in domains including *politician, actors, scientist, song, film, country, city, river, company, game, etc.*. 15 people in the field of Semantic Web were selected as reviewers and each entity was evaluated by at least 7 reviewers to produce the top-5 and top-10 summaries. The average number of properties for each entity was 44.

Based on the two types of RDF supplement methods we discussed in 5.4.3, we applied two different configurations for the proposed model. In the first experiment, ES-LDA @config-1, we configured the system to supplement each entity (document) by repeating each object based on the *number of categories* that the object has in the DBpedia knowledge base. For example, for the triple $< J.C.Penney, industry, Retail >$ we repeated *Retail* object, 5 times in that document, as *Retail* has five different categories in DBpedia (i.e. *"Retailers, Retailing, French words and phrases, Merchandising, Marketing"*)

In the second experiment, ES-LDA @config-2, we configured the system to supplement each entity (document) by adding the corresponding category(ies) of each object into the document. In this case, each entity is defined as a bag of words including objects and categories of each object. For example, for the aforementioned triple, in addition to the *Retail* we included *"Retailers, Retailing, French words and phrases, Merchandising, Marketing"* as the corresponding categories to the *Retail* object.

[4] `http://wiki.knoesis.org/index.php/FACES`

Table 5.2: Overall quality results of different models. Best result are bold.

Model	**Top-5**	**Top-10**
ES-LDA @ config-1	**1.20**	**3.50**
ES-LDA @ config-2	1.10	3.26
LinkSUM@ config-1	1.20	3.15
LinkSUM@ config-2	1.20	3.20
FACES	0.93	2.92

For the other parameters, we assumed a symmetric Dirichlet prior and set $\beta = 0.01$ and $\alpha = 50/R$, where R is the total number of unique predicates. We ran the Gibbs sampling algorithm for 1000 iterations and computed the posterior inference after the last sampling iteration. We selected the top-5 and top-10 most probable properties for each entity and calculate the quality of the summary for each entity through equation 5.8.

$$Quality(Sum(e)) = \frac{1}{n}\sum_{i=1}^{n} |Sum(e) \cap Sum_i^I(e)| \tag{5.8}$$

In our experiments, we used the quality of the summary proposed in [29], in which n ideal summaries $Sum_i^I(e)$ generated by expert users for $i = 1, ..., n$ and the summaries generated by the system $Sum(e)$ were compared. The average of the overlap between an ideal summary and a summary generated by the system is denoted as the quality of the summary, which is $0 \leq Quality(Sum(e)) \leq k$ in the top-k settings.

Table 5.3: Top-10 predicates for three randomly selected entities after applying three different models.

Marie Curie			Reign of Fire			Seychelles		
ES-LDA	**LinkSUM**	**FACES**	**ES-LDA**	**LinkSUM**	**FACES**	**ES-LDA**	**LinkSUM**	**FACES**
doctoralStudents	birthPlace	spouse	starring	country	starring	leaderName	largestCity	leaderName
doctoralAdvisor	birthPlace	field	producer	starring	country	governmentType	governmentType	governmentType
deathPlace	field	workInstitutions	music	starring	distributor	leaderTitle	governmentType	largestCity
children	field	birthPlace	director	starring	musicComposer	officialLanguage	governmentType	sovereigntyType
knownFor	knownFor	deathPlace	cinematography	studio	director	capital	governmentType	source
spouse	almaMater	doctoralAdvisor	country	producer	editing	currency	sovereigntyType	capital
almaMater	birthPlace	knownFor	distributor	producer	studio	timeZone	source	leaderTitle
birthPlace	knownFor	almaMater	studio	director	music	legislature	capital	language
field	doctoralAdvisor	doctoralStudents	editing	artist	producer	anthem	language	languages
establishedEvent	knownFor	thumbnail	screenplay	producer	thumbnail	callingCode	timeZone	legislature

5.5.1 Experiment Results

The summary in our model is defined as sets of representative triples that can summarize each entity (sets of triples with the same subject) in a way close to a human-created summary. We decided to use the last part of a *URI* to compare the generated summaries with the expert summaries and produce the Summary Quality for each entity and average them. As [86] reproduced the FACES overall Summary Quality based on this criteria and also applied it to their model, we decided to use their result as it was completely aligned with our summary definition.

In Table 5.2, we compare the quality of the results from LinkSUM, FACES, and ES-LDA with two distinct configurations (supplementing by object reputation and object categories). As Table 5.2 shows, the quality of our model outperforms the FACES approach, in both cases. The ES-LDA @ config-2 demonstrates a comparable result with the two configurations of LinkSUM, while ES-LDA @ config-1 outperforms LinkSUM. For some of the entities, the predicates that ES-LDA se-

Table 5.4: Probabilities of top-5 predicates for two randomly selected entities.

Lexus		Mortal Kombat Trilogy	
Predicate	**Probability**	**Predicate**	**Probability**
foundedBy	0.21	platforms	0.30
owner	0.17	publisher	0.18
location	0.15	developer	0.17
keyPerson	0.06	computingMedia	0.07
service	0.04	designer	0.05

Table 5.5: Distributions of two randomly selected predicates over top-5 objects.

Party		Starring	
Object	**Probability**	**Object**	**Probability**
Democratic Party (United States)	0.36	Arnold Schwarzenegger	0.05
Republican Party (United States)	0.17	Angelina Jolie	0.04
Democratic-Republican Party	0.12	Raven Symone	0.03
Communist Party of the Soviet Union	0.08	Matthew McConaughey	0.02
Independent(politician)	0.08	Alan Arkin	0.02

lected as top-5 most probable did not exist in the FACES dataset. It forced us to calculate the quality of summary for some of the entities with just 4 predicates instead of 5. We believe that might to be the only reason why top-5 Quality of Summary was lower than or equal to LinkSUM. Although, we had the same issue for the top-10 results, overall, ES-LDA shows a better performance in two configurations.

5.6 Discussion

We evaluated our approach against the state-of-the-art summarization techniques, including LinkSUM and FACES. LinkSUM primarily focuses on the most relevant facts for each entity, while FACES tries to keep a balance between diversity and relevancy in entity summarization. There is usually a trade-off between diversity and relevancy of the selected predicates. Our ES-LDA model maintains both diversity and relevancy, while representing each entity through *top-k* predicates. As shown in Table 5.2, our model outperforms the state-of-the-art approaches. Table 5.3 illustrates a sample of entities from the dataset along with their top-10 predicates, for all approaches. As Table 5.3 shows, the LinkSUM model is focusing more on the *objects*, while predicate repetition is permitted. For example, <*Marie_Curie, birthPlace, Warsaw*>, <*Marie_Curie, birthPlace, Russian_Empire*>, and <*Marie_Curie, birthPlace, Congress_Polandare*> are representing *Marie_Curie*'s birth place. Although, they differ in terms of objects, it is arguable that referring to the same predicate with multiple objects that are more likely relevant reduces the chance of other important triples that could potentially appear in the summary. It should be noted that in the current ES-LDA configuration, we have not considered predicate repetition, thus, all the predicates of the triples appearing in the resultant summary are unique. FACES on the other hand, considers predicate diversity and tries to keep a balance between the diversity and relevancy but the overall quality of the FACES model is lower than LinkSUM and ES-LDA. In the FACES model, there are selected predicates which seems to be less informative in the sense to be top-10 representative for a particular entity.

For example, <*Marie_Curie, thumbnail, 200px-Marie_Curie_c1920.png*>, which is referring to a *png* file, could be replaced with more descriptive one. Additionally, our proposed technique features several unique characteristics: (1) the ES-LDA is a *knowledge-based probabilistic* model that combines prior knowledge with statistical learning technique into a unified framework for entity summarization; (2) for each entity, it ranks all predicates based on their importance by computing marginal probabilities for the predicates. Table 5.4 illustrates the top-5 predicates for a sample of two entities; and finally (3), each predicate can be represented as a probability distribution over objects in the ES-LDA model, which allows us to describe the relations (predicates) of the RDF graph based on its nodes as shown in Table 5.5.

5.7 Conclusions

We have proposed a knowledge-based probabilistic topic model, called ES-LDA, based on the RDF entity representation for entity summarization. In our experiments, we have applied two different configurations: one based on object repetitions and the other based on adding object's categories, to alleviate common RDF data problems including *sparseness, unnatural language, and lack of context*. We conducted extensive experiments, which show the quality of the top-10 triples in both configurations outperforms the state-of-the-art techniques, LinkSUM and FACES, while for the top-5 quality we surpassed FACES and equaled the LinkSUM results.

There are many interesting future research directions of this work. It would be interesting to investigate how this model and a much richer set of topic models that combine prior knowledge with statistical learning techniques could be used for various tasks in the Semantic Web domain, such as ontology summarization, ontology tagging, and finding similar ontologies.

Chapter 6

Combining Word Embedding and Knowledge-Based Topic Modeling for Entity Summarization[1]

[1]Seyedamin Pouriyeh, Mehdi Allahyari, Krys Kochut, Gong Cheng, Hamid Reza Arabnia, "Combining Word Embedding and Knowledge-Based Topic Modeling for Entity Summarization", 12th IEEE International Conference on Semantic Computing (ICSC 2018).
Reprinted here with permission of the publisher.

Abstract

Word embedding is becoming more popular in the Semantic Web community as an effective approach for capturing semantics in various contexts. In this paper, we combine word embedding technique and topic modeling to model RDF data for the entity summarization task. In our model, ES-LDA$_{ext}$, which is the extended version of our previous model, we utilize the word embedding technique to supplement RDF data before applying entity summarization. In addition, in the model presented here, we use RDF literals as a very good source of information to create more reliable and representative summaries for entities. To do that, we use the Named Entity Recognition approach to extract entities within literals before feeding them into the word embedding model to enrich the RDF data. Experimental results demonstrate the effectiveness of the proposed model.

6.1 Introduction

With the ongoing success of Linked Open Data (LOD)[2] as a way to publish large-scale data, the amount of Web content has been constantly growing. Massive datasets, such as DBpedia [17] and YAGO [47], have been created and are publicly available on the Web. They contain large amounts of knowledge for human and machine consumption in different formats. Resource Description Framework (RDF), is the data modeling language of the Semantic Web and is widely used to encode information and publish data on the Web. RDF represents data in the form of triples $< subject, predicate, object >$, which are the basis of those datasets. The aforementioned knowledge bases consist of millions, or even billions of entities and properties that connect those entities together through proper relations and make-up a large-scale knowledge graph to describe entities via RDF triples. These lengthy descriptions become a problem while performing various semantic tasks, due to the fact that it is difficult to extract or focus on relevant and useful information in that massive data. *Entity summarization*, as a task of producing a condensed, but still sufficient entity descriptions, has gained considerable attentions as a way to absorb and distill descriptive knowledge from RDF datasets. In this work, our contributions are twofold: (1) we combine a word embedding technique with topic modeling within a single framework for RDF entity summarization, and (2) we consider the triples with literals as good resources to represent the underlying entities. We propose a method to augment triples with

[2] `http://linkeddata.org`

datatype properties in such a way that increases their chance to be selected as *top-k* representative triples and the entity summary.

6.2 Related Work

RELIN, proposed by Cheng et al. [29], is based on the PageRank algorithm to extract representative triples, called representative features, for RDF graph entities. The SUMMARUM model [87], which also uses the PageRank algorithm to rank triples according to the popularity, utilizes the Wikipedia pages for better navigation within Linked Data through the ranking of triples. The two aforementioned approaches could not meet the diversity requirement in the summarization process. FACES [43], on the other hand, tries to incorporate the diversity in the selected triples for each entity. It uses the Cobweb algorithm [43] to cluster related triples before ranking them. Both SUMMARUM and FACES also discount literals in entity summarization. The extended version of FACES, FACES-E [42], and RELIN consider literals as good candidates to be utilized in entity summarization. FACES-E gleans types for literals in RDF triples and uses this technique to be able to employ literals in entity summarization while RELIN computes summaries for object and datatype properties (literals). Pouriyeh et al. [75] proposed ES-LDA which is a probabilistic topic model to create more reliable and representative summaries for entities. ES-LDA shows a better performance on the same dataset as compared to the LinkSUM and FACES models.

6.3 Preliminaries

6.3.1 Resource Description Framework (RDF)

An RDF data graph is a collection of entities (nodes) and relationships (edges) between them. Nodes are usually identified by unique IDs which are called *Uniform Resource Identifiers (URIs)* or by literal values (i.e., numbers, dates, strings, etc.) commonly referred to as *Literals*.

Definition 1 (Entity summary): Given an entity e and a positive integer k, a summary of the entity e, denoted *Sum(e, k)*, is the top-k subset of all predicates and corresponding objects that are most relevant to that entity.

6.3.2 Probabilistic Topic Modeling

The Latent Dirichlet Allocation (LDA) [19] is a generative probabilistic model for extracting thematic information (topics) from a corpus of document.

6.3.3 Word Embedding and Word2Vec

The word2vec model [63], which is recognized as one of the most popular and extensively used word embedding techniques, has recently attracted significant attentions from different communities, including machine learning and Semantic Web. The word2vec model focuses on training a neural network to predict neighboring words.

6.3.4 Named Entity Recognition

Named Entities are usually defined as nouns or noun phrases that refer to particular types of entities, such as persons, organizations, locations, and so on. Named Entity Recognition (NER), which is one of the core components in Natural Language Processing, aims to identify, extract and classify named entities in text.

6.4 Problem Statement

6.4.1 Problem Definition

In the literature, *Entity Summarization* is defined as selecting a small but representative subset of the original triples describing an entity. In this context, given an RDF data set, entities are described by sets of their properties (i.e., all triples with the entity as the subject) and corresponding objects. The ultimate goal is to choose *top-k* representative triples for each entity.

6.4.2 Topic Models for RDF Graphs

In topic modeling, defining documents and word-like elements are the key points. Since an RDF graph dataset is a collection of triples connected together where each triple t consists of a subject s, predicate p, and an object o, in the form of $<s, p, o>$, we can consider each entity with its predicates and corresponding objects as a "document".

Definition 2 (document): A document d is defined as a set of triples, $d = \{t_1, t_2, \cdots, t_n\}$, that describes a single entity e. Thus, all triples of a document d have the same subject.

In topic modeling, each document is treated as a "bag of words". The key point is what would be the best definition for the "words". In this paper, we define a "**word**" w as the subject or object of a triple t for document d. As all the subjects in the triples of document d are the same, hence, in practice each document can be considered as a "bag of objects"[3].

In the next step, in the pre-processing phase, in order to enhance the final performance of our model we filtered out the schema and dataset dependent predicates, such as *sameAs, wikiPageExternalLink, subject, wikiPageWikiLink*. Additionally, since RDF data are recognized as short text [83], we need to address several challenges including sparseness, unnatural language, and the lack of context [83], to be able to properly run topic models on RDF data.

6.4.3 Supplementing RDF Data

As topic modeling is based on statistics of the co-occurrence of terms (words), we expand each document by adding similar words, based on their similarity scores. We utilize the Word2Vec model (Skip-Gram architecture) to predict the most related objects (words) to each of the words in our bag of words for each entity, based on their similarity scores generated by Skip-Gram architecture. We use pre-trained entity vectors with Freebase. Entity vectors trained on 100B words from

[3] "bag of words" and "bag of objects" are interchangeably used.

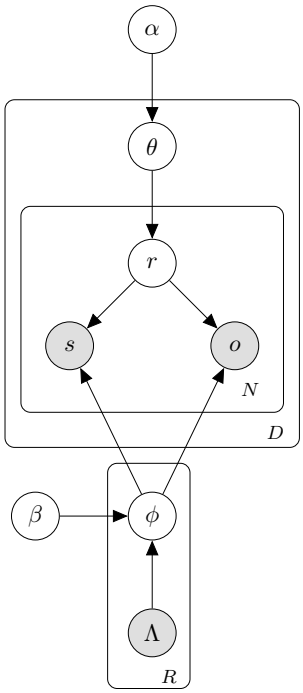


Figure 6.1: Entity Summarization Model

various news articles[4]. To be able to do that, we take advantage of the gensim package [78] to implement the deep learning with the word2vec model.

6.5 Proposed Model

ES-LDA$_{ext}$ is an extended version of ES-LDA, a probabilistic generative model for modeling entities in RDF graphs (Figure 6.1). The key ideas behind our model are as follows: (1) we exploit statistical topic models as the underlying quantitative framework for entity summarization, and (2) ES-LDA$_{ext}$ incorporates the prior

[4] `https://code.google.com/archive/p/word2vec/`

knowledge from the word embedding approach into the topic model, which is the primary difference with ES-LDA model.

In our model, each document is a multinomial distribution over the predicates and each predicate is a multinomial distribution over all the subjects and objects of the RDF graph. Furthermore, a document consists of a set of triples describing a single entity, i.e., all these triples share the same subject. Thus, we constrain the documents to only have the objects of related triples and also restrict the predicates to be defined only over the objects. In addition, for each predicate r, we further smooth its distribution by Λ_r. Λ is a matrix that has encoded the prior knowledge about predicate-object values from DBpedia. Section 6.5.1 explains how Λ is constructed. The generative process of ES-LDA$_{ext}$ is shown in Algorithm2. Please note that since the structure of the ES-LDA and ES-LDA$_{ext}$ models are similar, the generative processes, joint distributions and inference algorithms are essentially the same. The fundamental differences between two models are (i) type of the prior knowledge; (ii) generating the Λ matrix; and (iii) considering triples with object value of literals in ES-LDA$_{ext}$.

Following this process, the joint probability of generating a corpus $D = \{d_1, d_2, \ldots, d_{|D|}\}$, the predicate assignments $\mathbf{r}$ given the hyperparameters α, β and the prior matrix Λ is:

Algorithm 2: ES-LDA$_{ext}$ Model

```
foreach predicate r ∈ {1, 2, ..., R} do
    Draw an object distribution φ_r ~ Dir(β_r × Λ_r)
end
foreach document d ∈ {1, 2, ..., D} do
    Draw a predicate distribution θ_d ~ Dir(α_d)
    foreach subject s and object o of document d do
        Draw a predicate r ~ Mult(θ_d)
        Draw a subject s from predicate r, s ~ Mult(φ_r)
        Draw an object o from predicate r, o ~ Mult(φ_r)
    end
end
```

$$
\begin{aligned}
&P(\mathbf{o}, \mathbf{s}, \mathbf{r}|\alpha, \beta, \Lambda) \\
&= \int_{\phi} P(\phi|\beta; \Lambda) \prod_{d} \sum_{r_d} P(\mathbf{o_d}|r_d, \phi) P(\mathbf{s_d}|r_d, \phi) \\
&\times \int_{\theta} P(\theta|\alpha) P(\mathbf{r_d}|\theta, \phi) d\theta d\phi
\end{aligned} \tag{6.1}
$$

6.5.1 Predicate-Object Prior Knowledge Matrix Λ

We create the Λ matrix to encode the prior weights of the predicate-object pairs and integrate this knowledge into the topic model to smooth the predicate-object distributions ϕ. We build the Λ matrix of size $R \times O$, where R is the number of predicates and O is the number of objects in the RDF graph. Let f be an indicator function where $f(i, j) = 1$ if there is a triple in RDF graph with predicate i and object j, and 0 otherwise, for $1 \leq i \leq R$ and $1 \leq j \leq O$. Additionally, let v be

the number of similar objects (words) assigned to the object j. Then, we define Λ_{ij} as follows:

$$\Lambda_{ij} = \begin{cases} v & \text{if } f(i,j) = 1 \\ 1 & \text{otherwise.} \end{cases} \tag{6.2}$$

6.5.2 Inference using Gibbs Sampling

In this paper we use the collapsed Gibbs sampling [40] procedure in our ES-LDA$_{ext}$ topic model for estimating the posterior inference.

In our modified version of the learning algorithm to infer $p(o_i|r_j)$ and $p(r_j|d)$, we (1) constrain the objects that are not paired with a predicate to have 0 probability, i.e., $p(o_i|r_j) = 0$, if $(r_i, o_j) \notin$ RDF graph, and (2) $P(s|r_j) = 1$, since all the triples of a document have the same subject s. We derive the posterior inference from Eq. 6.1 as follows:

$$\begin{aligned} P(\mathbf{r}|\mathbf{o},\mathbf{s},\alpha,\beta,\Lambda) &= \frac{P(\mathbf{r},\mathbf{o},\mathbf{s}|\alpha,\beta,\Lambda)}{P(\mathbf{o}|\alpha,\beta,\Lambda)} \\ &\propto P(\mathbf{r},\mathbf{o}|\alpha,\beta,\Lambda) \propto P(\mathbf{r})P(\mathbf{o}|\mathbf{r})P(\mathbf{s}|\mathbf{r}) \end{aligned} \tag{6.3}$$

$$\begin{aligned} &P(r_i = r|o_i = o, \mathbf{r}_{-i}, \mathbf{o}_{-i}, \alpha, \beta, \Lambda) \propto \\ &\frac{n^{(d)}_{r,-i} + \alpha_r}{\sum_{r'}(n^{(d)}_{r',-i} + \alpha_{r'})} \times \frac{n^{(r)}_{o,-i} + \Lambda_{ro}\beta_o}{\sum_{o'}(n^{(r)}_{o',-i} + \Lambda_{ro}\beta_o)} \end{aligned} \tag{6.4}$$

where $n^{(r)}_o$ is the number of times object o is assigned to predicate r. $n^{(d)}_r$ denotes the number of times predicate r is associated with document d. The subscript $-i$

indicates that the contribution of the current object o_i being sampled is removed from the counts. After Gibbs sampling, we can use the sampled predicate to estimate the probability of a predicate, given a document, θ_{dr} and the probability of an object, given a predicate, ϕ_{ro}:

$$\theta_{dr} = \frac{n_r^{(d)} + \alpha_r}{\sum_{r'} (n_{r'}^{(d)} + \alpha_{r'})} \tag{6.5}$$

$$\phi_{ro} = \frac{n_o^{(r)} + \Lambda_{ro}\beta_o}{\sum_{o'} (n_{o'}^{(r)} + \Lambda_{ro}\beta_o)} \tag{6.6}$$

6.6 Experiments

We compared our ES-LDA$_{ext}$ model with the the state-of-the-art techniques including ES-LDA [75], LinkSUM[86], FACES [43],FACES-E [42],and RELIN [29]. We demonstrated the effectiveness of our model by showing that it produces closer results to human judgment, as compared to the other systems. We ran two different experiments with multiple configurations and evaluated ES-LDA$_{ext}$ based on the quality of summary (equation 6.7) proposed in [29], in which n ideal summaries $Sum_i^I(e)$ generated by expert users for $i = 1, ..., n$ and the summaries generated by the system $Sum(e)$ were compared.

$$Quality(Sum(e)) = \frac{1}{n} \sum_{i=1}^{n} |Sum(e) \cap Sum_i^I(e)| \tag{6.7}$$

For the other parameters, we assumed a symmetric Dirichlet prior and set $\beta = 0.01$ and $\alpha = 50/R$, where R is the total number of unique predicates. We

ran the Gibbs sampling algorithm for 1000 iterations and computed the posterior inference after the last sampling iteration. We selected the top-5 and top-10 most probable properties for each entity and calculated the quality of the summary for each entity through equation 6.7, for both experiments.

6.6.1 The First Experiment

In the first experiment we operated on the same dataset[5] that was used in the experiments conducted by the FACES and LinkSUM systems. As Table 6.1 shows, our proposed ES-LDA$_{ext}$ model outperformed all other baselines, including our previous ES-LDA model, which shows that exploiting word embedding technique has significantly improved the quality of summary in both top-5 and top-10 cases. For the details of different configurations of other baselines, please see [75].

Table 6.1: Overall quality results of different models. Best result are bold.

Model	**Top-5**	**Top-10**
ES-LDA$_{ext}$	**1.27**	**3.71**
ES-LDA @ config-1	1.20	3.50
ES-LDA @ config-2	1.10	3.26
LinkSUM@ config-1	1.20	3.15
LinkSUM@ config-2	1.20	3.20
FACES	0.93	2.92

6.6.2 The Second Experiment

In this experiment, in addition to the triples with object properties, we involved the triples having literals as object value of datatype properties. We first fil-

[5] `http://wiki.knoesis.org/index.php/FACES`

tered out the triples with numeric literals as they are not very informative and only considered the string literals. Literals often comprise a few words, or even a sentence. Therefore, we needed one more pre-processing step before running word2vec model to extract similar words. We employed Natural Language Tool Kit (NLTK) to recognize named entities from the literals. Then, for each extracted entity, we identified the similar words according to the similarity score and added them to the bag of words. By doing this, we increased the co-occurrence of similar words for literals in the document of each entity and consequently, giving them a chance to be selected as part of top-k representative triples. Among the baseline approaches in entity summarization, only FACES-E [42], RELIN [29], and RELINM (the modified version of RELIN) took literals into account. To evaluate our model, we used the FACES-E dataset which contained 80 unique entities. Table 6.2 shows that ES-LDA$_{ext}$ outperforms significantly other baselines for top-5 summaries. For the top-10 summarizes, ES-LDA$_{ext}$ gives better quality than RELIN and RELINM but could not meet FACES-E result. Since the fine-grained results of FACES-E system were not available[6], the reasons that we can think of are as follows: (i) we discard literals with numeric values, however, some users might have chosen from these triples in the summary; and (ii) some of the triples with literals might unnecessarily get overweighted (large Λ_{ij}) because of the large number of similar words the recognized name entities from these literals may have. This could make some of these triples appear in the top-k summary regardless of being unimportant, which can ultimately lead to results with lower quality.

[6]We contacted the first author, but they could not provide us with their results dataset.

Table 6.2: Overall quality results of different models (considering literals). Best result are bold.

Model	**Top-5**	**Top-10**
ES-LDA$_{ext}$	**1.65**	3.95
FACES-E	1.53	**4.53**
RELINM	1.02	3.65
RELIN	0.96	3.09

6.7 Conclusions

We have proposed ES-LDA$_{ext}$, an extended version of ES-LDA model, that integrates word embedding and knowledge-based probabilistic topic model for entity summarization. In addition, we included literal-valued properties to produce more reliable and comprehensive summaries. We utilized both Named Entity Recognition and Word Embedding techniques to spot entities within literals and extract similar words through word2vec model, respectively, in order to supplement the RDF data. In general, the results of exhaustive experiments confirm that combining word embedding technique with topic models improves the quality of summary. However, in the second experiment, we could not achieve the expected quality for top-10 triples.

There are interesting future directions of this work. Considering the link or relation between the extracted entities from literals and the corresponding subject as a way to weight extracted entities is a potential area of research to improve the quality of summary. Also, taking numeric literals into account could be another path to develop an effective entity summarization model.

Chapter 7

R-LDA: Profiling RDF datasets using Knowledge-based Topic Modeling [1]

[1] Seyedamin Pouriyeh, Mehdi Allahyari, Gong Cheng, Hamid Reza Arabnia, Krys Kochut, "R-LDA: Profiling RDF datasets using Knowledge-based Topic Modeling"
Submitted to 17th IEEE International Conference On Machine Learning And Applications (ICMLA 2018).

7.1 Abstract

Recently, Linked Open Data (LOD) has experienced an exponential growth via publishing huge volume of datasets on the Web. This vast amount of information needs to be searched, queried, and interlinked easier than before. It is recommended that potential data publishers provide recapitulative information about their datasets published on the Web. These information which play as a metadata, will facilitate those datasets to be discovered easily. As it is not always the case, we are faced with a large number of datasets without a proper profile, leading to a high demand for different data profiling techniques. In this paper, we focus on RDF dataset profiling utilizing unsupervised machine learning techniques, namely knowledge based topic modeling. We also investigate the use of Wikipedia categories to represent the topics identified in an RDF dataset. In the proposed model, called R-LDA, we extract a number of representative topics for an RDF dataset and annotate them with Wikipedia categories. The union of the assigned categories serves as a profile of the dataset, in a sense that it provides an overall characterization of the content of the dataset.

7.2 Introduction

With the explosive growth of the Linked Open Data (LOD)[2] and an ever-increasing number of datasets which are published via LOD, there is a real need to create and adopt automatic methods that make those datasets easily discoverable, queried, and used in applications [16]. Recent LOD statistics have shown thousands of publicly available linked datasets in different domains, including over 28 billion of distinct triples [35], suitable for human and machine consumption. Although the LOD cloud is considered as a valuable source of knowledge, mostly represented in the form of Resource Description Framework (RDF) datasets, its application is still largely not exploited, as exploring large and unfamiliar datasets remains a challenge. Given this scale of RDF datasets along with their heterogeneity, often inconsistency and the lack of suitable metadata, finding the resources that can be linked, queried, and used in different applications, such as entity linking, entity summarization, query federation, etc., has become an interesting area of research within the Semantic Web. RDF dataset profiling techniques, which aim to facilitate data consumption and data integration with statistics and useful metadata about the content of the RDF datasets, are considered as a promising approach to address this challenge.

Recently, a considerable amount of research has been dedicated to the Linked-Data profiling task, involved in Web-based techniques to locate, browse, and search RDF datasets *LODlive*[3]. Cheng et al. [28] extract a small representative subset of triples from a dataset as a way to quickly inspect its contents. *LOD-*

[2]http://lod-cloud.net/
[3]http://en.lodlive.it/

Stats introduced by [9] is a framework to produce statistics about RDF datasets. Given an RDF dataset, its profiling aims to create descriptive information about the data and collect statistics about the dataset at data and schema layers. In the current research literature, it is considered as the task of providing valuable insight into the data, such as the statistics about value distributions and locating and extracting information patterns in the data. RDF dataset profiles are usually expected to represent the importance of datasets without any extra need for detailed inspection of the raw data. It is more significant when dealing with an unfamiliar dataset [56], where profiling facilitates assessing the importance of the dataset and improving efficiency of applications utilizing the dataset. Currently, there is no established definition of data profiling and related tasks. However, data profiling typically covers several tasks, including [7]:

- *Statistical profiling:* this type of data profiling mainly focuses on statistical information about the number of entities as well as the distribution of properties and RDF triples, which are often related to data types and patterns in the dataset.

- *Metadata profiling:* dataset profile with respect to the metadata should cover the main informative categories, including the general information (dataset description, release and update dates), practical information (access points, data dumps), and legal information (license information, openness).

- *Topical profiling:* the representative knowledge on the content and structure of the dataset in the form of tags, keywords, categories, informative subgraphs, etc.

In this paper, we focus on Topical profiling techniques in order to address the challenges of generating a descriptive dataset profile. We propose a probabilistic framework based on topic modeling to produce a reliable RDF dataset profile based on Wikipedia categories.

The remainder of the paper is organized as follows. In Section 7.3, we review recent work in this area. Section 7.4 covers important preliminary definitions. Problem definition and the proposed frameworks architecture is discussed in Section 7.5 and 7.6 respectively. In Section 7.7, we evaluate the proposed model and finally conclude and outline some future work in Section 7.8.

7.3 Related Work

As LOD datasets, and in particular RDF datasets, vary with respect to different features, such as statistics, quality, dynamics, etc., discovering reliable information with related to these features is essential in most applications. In general, state-of-the-art approaches focus on different aspects of statistics and topical content in order to generate a profile describing an RDF dataset.

Assaf et al.[7] proposed *Roomba*, a framework to automatically generate, validate, and enrich descriptive dataset profiles in four main categories, including general information, access, ownership or provenance.

The *ExpLOD* [52] tool utilizes the metadata about the structure of an RDF dataset (set of used RDF classes and properties) in order to summarize the dataset. The metadata is augmented with other information such as the number of instances per class or the number of used properties.

Langegger et al. in [53] presented *RDFStats* a model which aims to generate statistics such as entity counts (per class) and histograms (per class, property, value type) for RDF datasets.

LODStats [9] is a statement-stream-based tool which can be used to gather 32 different statistics for a dataset. These statistics mainly represent the dataset at both the data (instance) and schema (vocabulary) levels using frequencies of triples, triples with blank nodes, average length of literals, labeled subjects, class and property usage, number of owl:sameAs links, class hierarchy depth, cardinalities, and others.

Abedjan et al. proposed *ProLOD* [1] as a Web-based tool, which analyzes object and literal values of triples in an RDF dataset and generate statistics, such as data type and patterns distribution upon them. Mining and cleansing datasets are two other available options in the extended version of *ProLOD*, called *ProLOD++* [4], which enables it to generate a profile based on key analysis components such as frequencies, distribution of subjects, predicates, and objects.

Although the existing works primarily focused on different aspects of RDF datasets at the schema and data layers, none of them have provided topic-wise profiling using knowledge-based topic modeling techniques. Additionally, we assign Wikipedia categories, which are semantically relevant to most or all of the discovered topics.

[4]https://www.hpi.uni-potsdam.de/naumann/sites/prolod++/

7.4 Preliminaries

7.4.1 Resource Description Framework (RDF)

RDF [46] is one of the popular data model languages for representing knowledge in the form of resources on the Web. Shortly speaking, an RDF dataset is a collection of resources (entities) and properties describing them. Resource descriptions (their properties), are represented in a form of triples <*subject, predicate, object*>, where subjects are resources and are usually represented by unique identifiers, called *Uniform Resource Identifiers (URIs)*. Objects are either resources or literal values (i.e., numbers, dates, strings, etc.), commonly referred to as *Literals (L)*. Properties are also resources identified by URIs. Some resources do not have URIs and are referred to as blank nodes.

7.4.2 Probabilistic Topic Modeling

The Latent Dirichlet Allocation (LDA) [19] is a generative probabilistic model for extracting thematic information (topics) from a corpus of documents. LDA assumes that each document is a mixture of topics (assuming K topics), where each topic is a multinomial probability distribution over the words.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ be a collection of documents and $\mathcal{V} = \{w_1, w_2, \ldots, w_{|\mathcal{V}|}\}$ the vocabulary (words) of the corpus. A topic $z_j, 1 \leq j \leq K$ is represented as a multinomial probability distribution over the $|\mathcal{V}|$ words, $p(w_i|z_j)$, $\sum_i^{|\mathcal{V}|} p(w_i|z_j) = 1$. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words,

given the document d, is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \tag{7.1}$$

The generative process for the corpus $\mathcal{D}$ is as follows:

1. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \text{Dir}(\beta)$
2. For each document $d \in \{1, 2, \ldots, \mathcal{D}\}$,
   (a) Sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
   (b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
      i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
      ii. Sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

In the LDA model, the word-topic distribution $p(w|z)$ and topic-document distribution $p(z|d)$ are learned entirely in an unsupervised manner, without any prior knowledge about what words are related to the topics and what topics are related to individual documents.

7.5 Problem Statement

In this section, we describe how to utilize topic modeling for RDF dataset profiling.

7.5.1 Problem Definition

With the tremendous growth in both the size and complexity of available RDF datasets, especially those within LOD, the process of exploring and exploiting these datasets in various applications is becoming harder, especially when these RDF datasets do not provide any meta-data describing their overall characteristics and content. In fact, the heterogeneity of these datasets pose significant challenges for applications and users when trying to find useful datasets without having any prior knowledge about those available datasets.

RDF dataset profiling techniques can help to generate descriptions and various statistics that offer the needed insight into the content of the datasets. In this context, a broad range of techniques for RDF dataset profiling have been proposed that typically focus on identifying the following three characteristics of the dataset: its statistics, schema, and some description of its actual content. Up to this point, none of them considered topic-modeling based approaches to describe the dataset's content. Given an RDF dataset, our goal is (i) to find the a suitable number of topics, which represent the RDF dataset well and (ii) to assign a set of representative Wikipedia categories to the identified topics and then to the whole dataset.

Definition 1 (RDF Dataset's Profile): Given an RDF dataset, its *data profile* is a set of Wikipedia categories $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ that are most relevant to the dataset and represent its content well. The categories to be included in the data profile are selected based on the most coherent topics identified in the dataset.

7.5.2 Topic Models for RDF Graphs

Knowledge-based topic modeling techniques have been utilized to discover latent topics in text corpora. Topic modeling techniques have also been applied to other types of data, such as images [18], music analysis [49], etc. However, only recently Pouriyeh et al. [75] applied topic modeling to RDF data and created the ES-LDA model for RDF entity summarization.

In topic modeling, each document is represented as a "bag of words" and a key point is "what should be defined as *documents* and *words*?". Here, we will treat a description of a resource as a document, while predicate-object pairs will be regarded as words.

Definition 2 (Document): A document d is a set of *predicate-object (p-o)* pairs, $d = \{(p-o)_1, (p-o)_2, \cdots, (p-o)_n\}$, that form a description of a single resource r. Thus, all *predicate-object* pairs of a document d describe the same subject (an RDF resource being described).

For example: *Stephen King*, a resource representing an American author, is descrribed by the following triples (due to space limitations we have chosen only a few of triples and dropped the first part of each *URIs*:

<*Stephen_King, award, Hugo_Award*>,

<*Stephen_King, genre, Gothic_fiction*>,

<*Stephen_King, notableWork, Carrie_(novel)*>,

<*Stephen_King, influencedBy, William_Golding*>,

The corresponding document for the *Stephen_King* resource is represented by the following set of predicate-object pairs:

{*award-Hugo_Award, genre-Gothic_fiction, notableWork-Carrie_(novel), influencedBy-William_Golding*}.

Consequently, in our model, a word is a *predicate-object* pair, and so Words come from triples forming a description of a single resource, which we regard as a document. As all subjects in the triples of the same document are the same, we treat each document as a "bag of pairs". In the rest of this paper, we use the terms "word" and "pair" interchangeably.

7.6 Proposed Model

Figure 7.1 illustrates a probabilistic generative model of R-LDA, where each document is a multinomial distribution over topics and each topic is a multinomial distribution over all predicate-object pairs (words). Our model is similar to the standard LDA. However, unlike the standard LDA, where each topic is a multinomial distribution over the vocabulary from the Dirichlet prior β, in R-LDA model, each topic is a multinomial distribution over all predicate-object pairs of the entire RDF dataset.

R-LDA uses a similar generative process as the one used in the standard LDA. Consequently, the joint probability of generating a corpus $D = \{d_1, d_2, \ldots, d_{|D|}\}$, the topic assignments $\mathbf{z}$ given the hyperparameters α, β is:

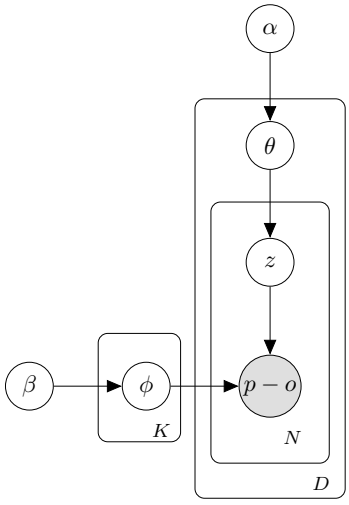


Figure 7.1: Topic model for RDF datasets

$$P(\mathbf{p}-\mathbf{o},\mathbf{z}|\alpha,\beta) = \int_{\phi} P(\phi|\beta) \prod_{d} \sum_{z_d} P(\mathbf{p}-\mathbf{o_d}|z_d,\phi)$$
$$\times \int_{\theta} P(\theta|\alpha) P(\mathbf{z_d}|\theta,\phi) d\theta d\phi \tag{7.2}$$

7.6.1 Inference using Gibbs Sampling

There are various algorithms, such as variational EM [19] and Gibbs sampling [40], which can be used to estimate the parameters of topic models. In this work, we utilize the collapsed Gibbs sampling [40] method in order to estimate the posterior inference of the proposed model.

Table 7.1: Probabilities of top-10 pairs (*predicate* object) for three randomly selected topics from 30 topics (K=30)

Topic 1		Topic 5		Topic 10	
Pairs (*predicate* object)	**Probability**	**Pairs (*predicate* object)**	**Probability**	**Pairs (*predicate* object)**	**Probability**
party Democratic_Party_(United_States)	0.015	*country* United_States	0.017	*bodyStyle* Sedan_(automobile)	0.006
party Republican_Party_(United_States)	0.015	*type* American_Viticultural_Area	0.010	*layout* Front-engine_design	0.005
profession Politician	0.010	*growingGrape* Cabernet_Sauvignon	0.007	*class* Mid-size_car	0.004
profession Lawyer	0.008	*grapes* Cabernet_Sauvignon	0.007	*layout* Four-wheel_drive	0.004
nationality United_States	0.006	*grapes* Chardonnay	0.006	*assembly* Germany	0.003
militaryBranch United_States_Army	0.005	*grapes* Merlot	0.006	*bodyStyle* Hatchback	0.003
battle World_War_II	0.004	*grapes* Cabernet_Franc	0.005	*class* Compact_car	0.003
party Federalist_Party	0.002	*growingGrape* Syrah	0.005	*assembly* Mexico	0.003
allegiance United_States	0.001	*growingGrape* Pinot_noir	0.005	*assembly* United_States	0.002
occupation Politician	0.001	*grapes* Sauvignon_blanc	0.004	*layout* Front-wheel_drive	0.002

Table 7.2: Topic Coherence on top T words (pairs). A higher coherence score means more coherent topics.

	TopWords (predicate − object)			
Topics	**5**	**10**	**15**	**20**
$K = 10$	-10.96	-48.65	-109.68	-199.88
$K = 20$	-9.27	-43.12	-104.66	-191.35
$\mathbf{K = 30}$	$\mathbf{-6.09}$	$\mathbf{-26.15}$	$\mathbf{-60.56}$	$\mathbf{-111.08}$
$K = 40$	-8.93	-37.51	-87.15	-147.06
$K = 50$	-9.35	-50.21	-105.03	-221.32

Collapsed Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm [80], which creates a Markov chain over the latent variables in the model and after a number of iterations will ultimately converge to the posterior distribution. In our model, we aim to construct a Markov chain that converges to the posterior distribution over $\mathbf{z}$ conditioned on observed pairs $\mathbf{p} - \mathbf{o}$, hyperparameters α, and β. We derive the posterior inference from Eq. 7.2 as follows:

$$
\begin{aligned}
P(\mathbf{z}|\mathbf{p}-\mathbf{o},\alpha,\beta) &= \frac{P(\mathbf{z},\mathbf{p}-\mathbf{o}|\alpha,\beta)}{P(\mathbf{p}-\mathbf{o}|\alpha,\beta)} \\
&\propto P(\mathbf{z},\mathbf{p}-\mathbf{o}|\alpha,\beta) \propto P(\mathbf{z})P(\mathbf{p}-\mathbf{o}|\mathbf{z})
\end{aligned}
\tag{7.3}
$$

$$
\begin{aligned}
&P(z_i = z|(p-o)_i = p-o, \mathbf{z}_{-i}, (\mathbf{p}-\mathbf{o})_{-i}, \alpha, \beta) \propto \\
&\frac{n^{(d)}_{z,-i} + \alpha_z}{\sum_{z'}\left(n^{(d)}_{z',-i} + \alpha_{z'}\right)} \times \frac{n^{(z)}_{p-o,-i} + \beta_{p-o}}{\sum_{(p-o)'}\left(n^{(z)}_{(p-o)',-i} + \beta_{p-o}\right)}
\end{aligned}
\tag{7.4}
$$

where $n^{(z)}_{p-o}$ is the number of times object $p-o$ is assigned to predicate z. $n^{(d)}_z$ denotes the number of times predicate z is associated with document d. The subscript $-i$ indicates that the contribution of the current object $(p-o)_i$ being sampled is removed from the counts. After Gibbs sampling, we can use the sampled predicate to estimate the probability of a predicate, given a document, θ_{dz} and the probability of an object, given a predicate, ϕ_{zp-o}:

$$
\theta_{dz} = \frac{n^{(d)}_z + \alpha_z}{\sum_{z'}\left(n^{(d)}_{z'} + \alpha_{z'}\right)} \tag{7.5}
$$

$$
\phi_{zp-o} = \frac{n^{(z)}_{p-o} + \beta_{p-o}}{\sum_{(p-o)'}\left(n^{(z)}_{(p-o)'} + \beta_{p-o}\right)} \tag{7.6}
$$

7.6.2 Topic coherence

In order to find a suitable number of topics, which represent well the RDF dataset, we utilize the topic coherence metric and compute *topic coherence score* to evaluate the quality of the identified topics. We consider the top 5, 10, 15, and 20 predicate-object pairs in each topic. The topic coherence score for measuring the quality of topics has been proposed in [64]. Arguably, this has become the most commonly used topic coherence evaluation method. Given a topic z and its top T words $V^{(z)} = (v_1^{(z)}, \cdots, v_T^{(z)})$ ordered by $P(w|z)$, the coherence score is defined as:

$$C(z; V^{(z)}) = \sum_{t=2}^{T} \sum_{h=1}^{t-1} \log \frac{D(v_t^{(z)}, v_h^{(z)}) + 1}{D(v_h^{(z)})} \tag{7.7}$$

where $D(v)$ is the document frequency of word v and $D(v, v')$ is the number of documents in which words v and v' co-occurred. It has been demonstrated that the coherence score is highly consistent with human-judged topic coherence [64]. Higher coherence scores indicate higher quality of topics.

7.6.3 Category Assignment

Wikipedia has become a vast source of information including huge volume of different resources from numerous areas of knowledge (it is an encyclopedia). Each resource is typically assigned to a number of categories, organized into a hierarchically. Wikipedia categories have been used to support many text processing tasks, including text classification [96]and annotation [61], etc. In this work, we utilize Wikipedia categories as a way to represent and describe the topics identified in an RDF dataset and then to describe the dataset itself. To select suitable categories,

Algorithm 3: Topic Category Assignment

Input : {P - a set of top-l pairs (predicate-object) of a topic T }
Output: {Topic T Category Assignment}

foreach pair $p \in P$ **do**
  $SubjectSet_p \leftarrow$set of all subjects among all triples with predicate and object values p
end
for *h=1* **to** *Max level* **do**
  foreach subject $s \in SubjectSet_p$ **do**
    $CategorySet_{p,h} \leftarrow$ Extract categories of s until level h.
  end
  $CommonCategorySet \leftarrow$ Max common categories of $CategorySet_{p,h}$
  $CScore = \frac{|\text{ p covered by } CommonCategorySet\ |}{|P|}$
  foreach category $c \in CommonCategorySet$ **do**
    $SumOfLevels =$ number of categories in path(s) c to immediate categories of p
    $CatNumber =$ count number of c until level h
    $Avg.height = \frac{SumOfLevels}{CatNumber}$
    $CategoryScore = \frac{\lambda \times CScore}{(1-\lambda) \times Avg.height}$
  end
end
Sort $CommonCategorySet$ based on $CategoryScore$

we use Algorithm 3. Given a topic and its *top-l* pairs, we first identify the subjects of all these pairs. Note, that each pair (predicate-object) may occur in a number of triples (subject-predicate-object), each with a different subject. For example, *Name-Helen* may be used to describe the city in Montana and a person named Helen. Then, our algorithm traverses the Wikipedia category hierarchy (in fact,

Table 7.3: Assigned categories based on Top-10 pairs of a randomly selected topic (K=30)

Topic 4	Assigned Categories
Pairs (*predicate* object)	(Ordered by *CategoryScore*)
modes Single-player_video_game	Games_on_seventh-generation_consoles
computingPlatform Microsoft_Windows	Video_games_by_platform
platforms Microsoft_Windows	Video_games
computingPlatform Xbox_360	Sony_Interactive_Entertainment
computingPlatform PlayStation_3	Games_on_Microsoft_platforms
genre Platform_game	
platforms Xbox_360	
computingPlatform PlayStation_2	
modes Multiplayer_video_game	
publisher Activision	

Wikipedia categories form a graph) to identify categories that serve as direct or indirect categories of subjects in most of the *top-l* pairs. If a high-scoring category (covering most pairs) is not yet found, we consider the next category level (i.e., the parents of all categories from the first level) and compute their scores, in turn. As we increase the level of considered categories (finding parents of parents, etc.), the chance of finding a category with a higher score (*CScore*) increases, as each higher category level consists of progressively more general categories. At the same time, very general categories (*Content* is the highest level Wikipedia category) would not serve as highly descriptive representations of topics. To avoid that, we define a penalty factor (*Avg.height*) to diminish the category score due to its height. It is included in the *CategoryScore* computation, in order to capture a trade off between the *coverage* of a category for a topic and the *specificity* of that category.

At each level, we check the coverage of each category (*CScore*) and then for each category $c \in CommonCategorySet$, we compute the penalty factor as a ratio of the number of categories in a path from category c to all immediate categories over the number of occurrences of category c from the immediate category level to the current level of category c. We compute *CategoryScore* of each category c at each level and produce the final result as an ordered set of categories (*CommonCategorySet*). We choose the top category as a descriptive category for that topic. Additionally, we define λ as a smoothing factor to control the influence of coverage score (*CScore*) and the penalty factor (*Avg.height*). In practice, $\lambda = 0.7$ has shown better results in our experiments.

7.6.4 Evaluation

For quantitative evaluation of our model, we asked ten human assessors to evaluate the extracted categories for different topics. We randomly selected a subset of topics and for each topic, the extracted Wikipedia category for each topic is provided. The assessors need to choose between "Good" and "Unrelated" to evaluate each topic based on the corresponding category and *top-10* predicate-object pairs in each topic. We use the *Precision@k*, taking the extracted Wikipedia category into consideration. We then averaged the precision over all the topics.

Precision for a topic at top-k is defined as follows:

$$Precision@k = \frac{\#\text{ of ``Good'' category with rank} \leq k}{k} \tag{7.8}$$

7.7 Experiments

For experimental evaluation of our model, we randomly chose 50 classes in several different domains from DBpedia and extracted 5,225 instances from those classes, which resulted in 211,512 distinct predicate-object pairs. For other parameters, we assumed a symmetric Dirichlet prior and set $\beta = 0.01$ and $\alpha = 50/K$, where K is the number of topics. We ran the Gibbs sampling algorithm for 1000 iterations with the number of topics set to K=$\{$*10, 20, 30, 40, 50*$\}$ and computed the posterior inference after the last sampling iteration. Table 7.1 shows the top-10 most probable predicate-object pairs in three different topics, selected as an example, selected out of K=*30* topics. Additionally, in order to find the best number of topics to represent the RDF dataset used in our experiment, we computed the topic coherence with varying numbers of topics, set to $K \in \{10, 20, 30, 40, 50\}$ and for *top-5*, *top-10*, *top-15*, and *top-20* predicate-object pairs in each topic. As Table 7.2 depicts, the most coherent topics describing our RDF dataset were obtained with K=*30* topics. It confirms that when we increase the number of topics the chance of having more relevant pairs in each topic also increases, because we expect to have the number of topics to be relatively close to the number of classes we have in the RDF dataset. On the other hand, as the number of topics gets closer to the number of classes, the topic coherence declines. The main reason for this outcome is the possible overlap between classes, which makes it harder for the pairs to be discriminated under different topics. To evaluate the final result, we computed the average precision of each topic, considering Top-5 assigned categories using equation 7.8. We asked ten experts in the Semantic Web field

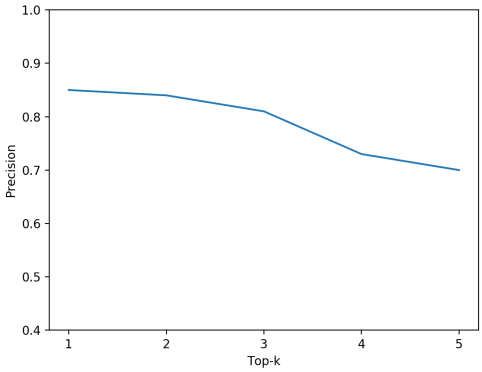


Figure 7.2: The precision of assigned categories using human evaluation

to evaluate the Top-5 assigned categories for each topic. Figure 7.2 shows the averaged precision over topics (*K=30*). As it illustrates, we received the highest precision at Top-1 assigned Wikipedia categories for topics, which also confirms the efficiency of our algorithm for topic category assignment.

7.8 Conclusions

In this paper, we proposed an automatic approach for RDF dataset profiling with Wikipedia categories using a knowledge-based topic modeling (called R-LDA). Given an RDF dataset, our approach, which is a novel *Topical profiling* method, aims to find the best number of representative topics and extract a proper set of Wikipedia categories for each obtained topic. Computing topic coherence enables our model to find the best number of representative topics for an RDF dataset. Additionally, the proposed model utilizes Wikipeida categories as a way

to describe each topic and ultimately, uses those Wikipedia categories as key representative terms in order to generate an RDF profile of the dataset. To the best of our knowledge, there are no similar topic-model based RDF profiling systems, so we decided to use *Precision@k* technique to evaluate the effectiveness of our model.

There are many avenues to extend the current work. It would be interesting to involve the RDFS schema in our model and incorporate the schema knowledge to improve this model. Furthermore, our method can be utilized as a way for RDF dataset partitioning, as it possible to represent each document (resource description) as a vector of K topic membership probabilities, we can apply different similarity measures in order to cluster similar entities with respect to their topics and ultimately partition the given RDF dataset into K partitions.

Chapter 8

Conclusion and Future Work

With an ever-increasing size and number of datasets which are published within the Linked Open Data project, there is a immediate need to create and adopt automatic methods that make those datasets easily discoverable, queried, and used in various applications. In this work, we aimed to address these issues within two broad categories: RDF dataset summarization and RDF dataset profiling, utilizing knowledge-based topic modeling. We proposed two new models, which are dedicated to RDF datasets and applicable in different tasks, such as RDF dataset summarization and RDF dataset profiling.

With respect to RDF dataset summarization, we focused mainly on entity summarization using topic modeling. Experimental results demonstrated the effectiveness of the proposed model, compared to other state-of-the-art approaches. Additionally, the extended version of our model for entity summarization confirmed the efficiency of our model. Considering RDF dataset profiling, we proposed a model based on knowledge-based topic modeling technique to identify a

number of topics that represent a given RDF dataset and then create a profile based on Wikipedia categories corresponding to those topics. Additionally, we conducted a comprehensive review about different ontology summarization techniques with respect to the ontology schema layer. We presented different types of graphs used for representation of an ontology and different measures of node importance.

8.1 Summary of Contributions

The major contributions of this dissertation is as follows:

1. **Graph-based Ontology Summarization: A Survey.** In Chapter 4, we have investigated different graph models and measures for ontology summarization. We focus mainly on graph-based methods, which represent an ontology as a graph and apply centrality and other measures to identify the most important elements of an ontology as its summary. After analyzing their strengths and weaknesses, we highlight a few potential directions for future research.

2. **ES-LDA: Entity Summarization using Knowledge-based Topic Modeling.** In Chapter 5, we have proposed a knowledge-based probabilistic topic model, called ES-LDA, for entity summarization task. In our model, we combined prior knowledge with statistical learning techniques within a single framework to create more reliable and representative summaries for entities. We have applied two different configurations to alleviate common RDF data problems including *sparseness, unnatural language, and lack of context*. We demonstrate the effectiveness of our approach by conducting extensive experiments and show that our model outperforms the state-of-the-art techniques and enhances the quality of the entity summaries.

3. **Combining Word Embedding and Knowledge-Based Topic Modeling for Entity Summarization.** In Chapter 6, we have proposed ES-LDA$_{ext}$, an extended version of the ES-LDA model, which integrates word embed-

ding and knowledge-based probabilistic topic modeling for entity summarization. In addition, we include literal-valued properties as a good source of information to produce more reliable and comprehensive summaries. We utilize both Named Entity Recognition and Word Embedding techniques to spot entities within literals and extract similar words through the Word2Vec model, respectively, in order to supplement the RDF data. The results of our exhaustive experiments confirm that combining word embedding technique with topic models improves the quality of summary.

4. **R-LDA: Profiling RDF datasets using Knowledge-based Topic Modeling.** In Chapter 7, we proposed a new topic model for RDF datasets and applied it to RDF dataset profiling. Given an RDF dataset, our approach, which is a novel *Topical profiling* method, identifies a number of coherent topics in the dataset and assigns a set of representative Wikipedia categories to each obtained topic. Collectively, the categories can be used as the profile of the RDF dataset.

8.2 Future Work

1. **Ontology Summarization using Knowledge-based Topic Modeling.**

   There are many interesting future research directions for this work. It would be interesting to investigate how to use the prior knowledge from the schema layer in entity summarization summarization task.

Additionally, considering the link or relation between the extracted entities from literals and the corresponding subject as a way to weight extracted entities is a potential area of research to that could lead to the improved quality of summaries. Also, taking numeric literals into account could be another path leading to the development of an effective entity summarization model.

2. **Dynamic Ontology Summarization.**

   Although many algorithms for the ontology summarization problem have been proposed, empirical results reported in the literature suggest that none of them consistently generates an ontology summary of sufficiently high quality. Ideally, an ontology summarization technique needs to be more flexible in the way to enable users or applications to *tune* the model in order to generate different summaries based on different requirements or inputs. In other words, *dynamic or adaptive ontology summarization* can be viewed as an interesting topic to explore.

3. **Abstractive or Hybrid Ontology Summarization.**

   The available approaches apply *extractive techniques* to generate the final summary. In the extractive scenario, a subset of the terms and/or axioms from the original input ontology are selected as a summary. *Abstractive (Non-extractive) or Hybrid (Extractive and Abstractive)* ontology summarization will be a new area with great potential. In that scenario, the key research question is how to define the output of ontology summarization, i.e., as some kind of a high-level aggregate representation of terms and axioms.

4. **Ontology Partitioning** Ontology partitioning an interesting avenue for research. In ontology partitioning is an ontology is divided into subsets, called partitions, to alleviate some of the challenges posed by large ontologies, such as scalability, complexity, and maintenance. Utilizing topic modeling techniques in order to partition a given ontology based on the extracted topics could be highly effective in future research.

Bibliography

[1] Ziawasch Abedjan, Toni Gruetze, Anja Jentzsch, and Felix Naumann. Profiling and mining rdf data with prolod++. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1198–1201. IEEE, 2014.

[2] Kat R Agres, Stephen McGregor, Karolina Rataj, Matthew Purver, and Geraint A Wiggins. Modeling metaphor perception with distributional semantics vector space models. In *C3GI@ ESSLLI*, 2016.

[3] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets with the void vocabulary. 2011.

[4] Mehdi Allahyari and Krys Kochut. Automatic topic labeling using ontology-based topic models. In *14th International Conference on Machine Learning and Applications (ICMLA), 2015*. IEEE, 2015.

[5] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[6] Mehdi Allahyari, Seyedamin Pouriyeh, Krys Kochut, and Hamid R Arabnia. A knowledge-based topic modeling approach for automatic topic labeling. *International Journal of Advanced Computer Science and Applications*, 8(9):335–349, 2017.

[7] Ahmad Assaf, Raphaël Troncy, and Aline Senart. Roomba: An extensible framework to validate and build dataset profiles. In *International Semantic Web Conference*, pages 325–339. Springer, 2015.

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[9] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. Lodstats–an extensible framework for high-performance dataset analytics. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 353–362. Springer, 2012.

[10] Sören Auer, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Introduction to linked data and its lifecycle on the web. In *Proceedings of the 7th international conference on Reasoning web: semantic technologies for the web of data*, pages 1–75. Springer-Verlag, 2011.

[11] Krisztian Balog. *Encyclopedia of Database Systems*, chapter Entity Retrieval, pages 1–6. Springer New York, New York, NY, 2017.

[12] Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. Nonparametric spherical topic modeling with word embeddings. *arXiv preprint arXiv:1604.00126*, 2016.

[13] Tim Berners-Lee and James Hendler. Publishing on the semantic web. *Nature*, 410(6832):1023, 2001.

[14] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

[15] Timothy J Berners-Lee. Information management: A proposal. Technical report, 1989.

[16] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International journal on semantic web and information systems*, 5(3):1–22, 2009.

[17] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

[18] David M Blei and Michael I Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM, 2003.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[20] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[21] Janez Brank, Marko Grobelnik, and Dunja Mladenić. A survey of ontology evaluation techniques. 2005.

[22] Dan Brickley. Rdf vocabulary description language 1.0: Rdf schema. *http://www. w3. org/TR/rdf-schema/*, 2004.

[23] Shaofeng Bu, Laks VS Lakshmanan, and Raymond T Ng. Mdl summarization with holes. In *Proceedings of the 31st international conference on Very large data bases*, pages 433–444. VLDB Endowment, 2005.

[24] Anila Sahar Butt, Armin Haller, and Lexing Xie. DWRank: Learning concept ranking for ontology search. *Semantic Web*, 7(4):447–461, 2016.

[25] Bob Carpenter. Integrating out multinomial parameters in latent dirichlet allocation and naive bayes for collapsed gibbs sampling. Technical report, Technical report, LingPipe, 2010.

[26] Yixi Chen, Keting Yin, and Xiaohu Yang. A semantic-aware algorithm to rank concepts in an OWL ontology graph. *Information Technology Journal*, 9(4):825–831, 2010.

[27] Gong Cheng, Feng Ji, Shengmei Luo, Weiyi Ge, and Yuzhong Qu. BipRank: Ranking and summarizing RDF vocabulary descriptions. In *Joint Interna-*

tional Semantic Technology Conference, pages 226–241, Hangzhou, China, December 2011. Springer.

[28] Gong Cheng, Cheng Jin, Wentao Ding, Danyun Xu, and Yuzhong Qu. Generating illustrative snippets for open data on the web. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 151–159. ACM, 2017.

[29] Gong Cheng, Thanh Tran, and Yuzhong Qu. Relin: relatedness and informativeness-based centrality for entity summarization. *The Semantic Web–ISWC 2011*, pages 114–129, 2011.

[30] Gong Cheng, Danyun Xu, and Yuzhong Qu. C3d+ p: A summarization method for interactive entity resolution. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:203–213, 2015.

[31] Hong-Jie Dai, Richard Tzong-Han Tsai, Wen-Lian Hsu, et al. Entity disambiguation using a markov-logic network. In *IJCNLP*, pages 846–855, 2011.

[32] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *ACL (1)*, pages 795–804, 2015.

[33] Mike Dean, Guus Schreiber, S Bechhofer, F van Harmelen, J Hendler, I Horrocks, DL McGuinness, PF Patel-Schneider, and LA Stein. Owl web ontology language reference. w3c recommendation, 10 feb. 2004. *World Wide Web Consortium. http://www. w3. org/TR/owl-ref/(2009-03-27)*, 2004.

[34] Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Edith Schonberg, Kavitha Srinivas, and Li Ma. Scalable semantic retrieval through summarization and refinement. In *AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, July 2007. AAAI Press.

[35] Javier D Fernández, Wouter Beek, Miguel A Martínez-Prieto, and Mario Arias. Lod-a-lot. In *International Semantic Web Conference*, pages 75–83. Springer, 2017.

[36] Douglas H Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172, 1987.

[37] Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. Triplerank: Ranking semantic web data by tensor decomposition. *The Semantic Web-ISWC 2009*, pages 213–228, 2009.

[38] Anuradha Gali, Cindy X Chen, Kajal T Claypool, and Rosario Uceda-Sosa. From ontology to relational databases. In *International Conference on Conceptual Modeling*, pages 278–289. Springer, 2004.

[39] Weiyi Ge, Gong Cheng, Huiying Li, and Yuzhong Qu. Incorporating compactness to generate term-association view snippets for ontology search. *Information Processing & Management*, 49(2):513–528, March 2013.

[40] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[41] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[42] Kalpa Gunaratna, Krishnaprasad Thirunarayan, Amit Sheth, and Gong Cheng. Gleaning types for literals in rdf triples with application to entity summarization. In *International Semantic Web Conference*, pages 85–100. Springer, 2016.

[43] Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit P Sheth. Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering. 2015.

[44] Udo Hahn and Inderjeet Mani. The challenges of automatic summarization. *Computer*, 33(11):29–36, 2000.

[45] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[46] Hlomani Hlomani and Deborah Stacey. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, pages 1–5, 2014.

[47] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[48] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[49] Diane J Hu and Lawrence K Saul. A probabilistic topic model for music analysis. In *Proc. of NIPS*, volume 9. Citeseer, 2009.

[50] Karen Spärck Jones. Automatic summarising: The state of the art. *Information Processing & Management*, 43(6):1449–1481, 2007.

[51] Amit Krishna Joshi, Pascal Hitzler, and Guozhu Dong. Logical linked data compression. In *Extended Semantic Web Conference*, pages 170–184. Springer, 2013.

[52] Shahan Khatchadourian and Mariano P Consens. Explod: summary-based exploration of interlinking and rdf usage in the linked open data cloud. In *Extended Semantic Web Conference*, pages 272–287. Springer, 2010.

[53] Andreas Langegger and Wolfram Woss. Rdfstats-an extensible rdf statistics generator and library. In *Database and Expert Systems Application, 2009. DEXA'09. 20th International Workshop on*, pages 79–83. IEEE, 2009.

[54] Kisung Lee and Ling Liu. Scaling queries over big rdf graphs with semantic hash partitioning. *Proceedings of the VLDB Endowment*, 6(14):1894–1905, 2013.

[55] Isaac Lera, Carlos Juiz, and Ramón Puigjaner. Ontology summarization through simple pruning measures. In *International Conference on Knowl-*

edge Engineering and Ontology Development, pages 339–342, Barcelona, Spain, October 2012. SciTePress.

[56] Huiying Li. Data profiling for semantic web data. In *International Conference on Web Information Systems and Mining*, pages 472–479. Springer, 2012.

[57] Ning Li, Enrico Motta, and Mathieu d'Aquin. Ontology summarization: an analysis and an evaluation. 2010.

[58] Eetu Mäkelä. Aether–generating and viewing extended void statistical descriptions of rdf datasets. In *European Semantic Web Conference*, pages 429–433. Springer, 2014.

[59] Frank Manola, Eric Miller, Brian McBride, et al. Rdf primer. *W3C recommendation*, 10(1-107):6, 2004.

[60] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[61] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011.

[62] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[63] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[64] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics, 2011.

[65] Saket Navlakha, Rajeev Rastogi, and Nisheeth Shrivastava. Graph summarization with bounded error. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 419–432. ACM, 2008.

[66] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. *Mining text data*, pages 43–76, 2012.

[67] Alexandros Pappas, Georgia Troullinou, Giannis Roussakis, Haridimos Kondylakis, and Dimitris Plexousakis. Exploring importance measures for summarizing rdf/s kbs. In *European Semantic Web Conference*, pages 387–403. Springer, 2017.

[68] Peter F Patel-Schneider. Owl web ontology language semantics and abstract syntax, w3c recommendation. *http://www. w3. org/TR/2004/REC-owl-semantics-20040210/*, 2004.

[69] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great mi-

gration. In *Proceedings of the 25th international conference on world wide web*, pages 1419–1428. International World Wide Web Conferences Steering Committee, 2016.

[70] Thomas Penin, Haofen Wang, Thanh Tran, and Yong Yu. Snippet generation for semantic web search engines. In *Asian Semantic Web Conference*, pages 493–507, Bangkok, Thailand, December 2008. Springer.

[71] Silvio Peroni, Enrico Motta, and Mathieu dAquin. Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. *The Semantic Web*, pages 242–256, 2008.

[72] Carlos Eduardo Pires, Paulo Sousa, Zoubida Kedad, and Ana Carolina Salgado. Summarizing ontology-based schemas in pdms. In *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, pages 239–244. IEEE, 2010.

[73] Seyedamin Pouriyeh, Mehdi Allahyari, Krys Kochut, and Hamid Reza Arabnia. A comprehensive survey of ontology summarization: Measures and methods. *arXiv preprint arXiv:1801.01937*, 2018.

[74] Seyedamin Pouriyeh, Mehdi Allahyari, Krys Kochut, Gong Cheng, and Hamid Reza Arabnia. Combining word embedding and knowledge-based topic modeling for entity summarization. In *Semantic Computing (ICSC), 2018 IEEE 12th International Conference on*, pages 252–255. IEEE, 2018.

[75] Seyedamin Pouriyeh, Mehdi Allahyari, Krzysztof Kochut, Gong Cheng, and Hamid Reza Arabnia. Es-lda: Entity summarization using knowledge-based

topic modeling. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 316–325, 2017.

[76] Seyedamin Pouriyeh, Mehdi Allahyari, Qingxia Liu, Gong Cheng, Hamid Reza Arabnia, Yuzhong Qu, and Krys Kochut. Graph-based ontology summarization: A survey. *arXiv preprint arXiv:1805.06051*, 2018.

[77] Paulo Orlando Queiroz-Sousa, Ana Carolina Salgado, and Carlos Eduardo Pires. A method for building personalized ontology summaries. *Journal of Information and Data Management*, 4(3):236, 2013.

[78] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[79] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.

[80] Christian P Robert and George Casella. *Monte Carlo statistical methods*, volume 319. Citeseer, 2004.

[81] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[82] Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, January 2013.

[83] Jennifer Sleeman, Tim Finin, and Anupam Joshi. Topic modeling for rdf graphs. In *LD4IE@ ISWC*, pages 48–62, 2015.

[84] Jinsong Su, Deyi Xiong, Yang Liu, Xianpei Han, Hongyu Lin, Junfeng Yao, and Min Zhang. A context-aware topic model for statistical machine translation. In *ACL (1)*, pages 229–238, 2015.

[85] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008.

[86] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. Linksum: using link analysis to summarize entity data. In *International Conference on Web Engineering*, pages 244–261. Springer, 2016.

[87] Andreas Thalhammer and Achim Rettinger. Browsing dbpedia entities with summaries. In *European Semantic Web Conference*, pages 511–515. Springer, 2014.

[88] Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. Trank: Ranking entity types using the web of data. In *International Semantic Web Conference*, pages 640–656. Springer, 2013.

[89] Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. Rdf digest: Efficient summarization of rdf/s kbs. In *European Semantic Web Conference*, pages 119–134. Springer, 2015.

[90] Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. Ontology understanding without tears: The summarization approach. *Semantic Web*, (Preprint):1–19, 2017.

[91] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

[92] Yannis Tzitzikas, Dimitris Kotzinos, and Yannis Theoharis. On ranking RDF schema elements (and its application in visualization). *Journal of Universal Computer Science*, 13(12):1854–1880, 2007.

[93] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[94] W3C. Linked data @ONLINE, https://www.w3.org/standards/semanticweb/data.

[95] Xiaojun Wan and Tianming Wang. Automatic labeling of topic models using text summaries. In *ACL (1)*, 2016.

[96] Pu Wang, Jian Hu, Hua-Jun Zeng, and Zheng Chen. Using wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, 19(3):265–281, 2009.

[97] Buwen Wu, Yongluan Zhou, Pingpeng Yuan, Ling Liu, and Hai Jin. Scalable sparql querying using path partitioning. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 795–806. IEEE, 2015.

[98] Gang Wu, Juanzi Li, Ling Feng, and Kehong Wang. Identifying potentially important concepts and relations in an ontology. In *International Semantic Web Conference*, pages 33–49, Karlsruhe, Germany, October 2008. Springer.

[99] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456. ACM, 2013.

[100] Xiang Zhang, Gong Cheng, Wei-Yi Ge, and Yu-Zhong Qu. Summarizing vocabularies in the global semantic web. *Journal of Computer Science and Technology*, 24(1):165–174, 2009.

[101] Xiang Zhang, Gong Cheng, and Yuzhong Qu. Ontology summarization based on rdf sentence graph. In *Proceedings of the 16th international conference on World Wide Web*, pages 707–716. ACM, 2007.

[102] Xiang Zhang, Hongda Li, and Yuzhong Qu. Finding important vocabulary within ontology. In *Asian Semantic Web Conference*, pages 106–112, Beijing, China, September 2006. Springer.

[103] Hai Zhao and Chunyu Kit. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *IJCNLP*, pages 106–111, 2008.