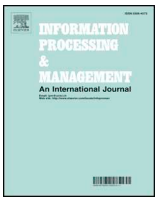
Information Processing and Management 57 (2020) 102238

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman



Evaluation of web service clustering using Dirichlet Multinomial Mixture model based approach for Dimensionality Reduction in service representation

Neha Agarwal[*,a], Geeta Sikka[a], Lalit Kumar Awasthi[b]

[a] Department of Computer Science and Engineering, Dr B R Ambedkar National Institute of Technology, Jalandhar 144011, Punjab, India
[b] Dr B R Ambedkar National Institute of Technology, Jalandhar 144011, Punjab, India

ARTICLE INFO

Keywords:
Web service clustering
Dirichlet Multinomial Mixture (DMM) model
Latent Dirichlet Allocation (LDA)
Topic modeling techniques
Clustering techniques

ABSTRACT

In recent years, mainly the functionality of services are described in a short natural text language. Keyword-based searching for web service discovery is not efficient for providing relevant results. When services are clustered according to the similarity, then it reduces search space and due to that search time is also reduced in the web service discovery process. So in the domain of web service clustering, basically topic modeling techniques like Latent Dirichlet Allocation (LDA), Correlated Topic Model (CTM), Hierarchical Dirichlet Processing (HDP), etc. are adopted for dimensionality reduction and feature representation of services in vector space. But as the services are described in the form of short text, so these techniques are not efficient due to lack of occurring words, limited content, etc. In this paper, the performance of web service clustering is evaluated by applying various topic modeling techniques with different clustering algorithms on the crawled dataset from ProgrammableWeb repository. Gibbs Sampling algorithm for Dirichlet Multinomial Mixture (GSDMM) model is proposed as a dimensionality reduction and feature representation of services to overcome the limitations of short text clustering. Results show that GSDMM with K-Means or Agglomerative clustering is outperforming all other methods. The performance of clustering is evaluated based on three extrinsic and two intrinsic evaluation criteria. Dimensionality reduction achieved by GSDMM is 90.88%, 88.84%, and 93.13% on three real-time crawled datasets, which is satisfactory as the performance of clustering is also enhanced by deploying this technique.

1. Introduction

In the fast-paced development of SOA (Service Oriented Architecture), the number of services or Web APIs is proliferating. Due to the vast benefits of web services, various companies like Microsoft, Amazon, IBM, and many more are providing services to their customers according to their needs. Existing services are mainly categorized into two streams: Simple Object Access Protocol (SOAP) based services and REpresentational State Transfer (REST) based services (Web API). In SOAP-based web services, the service provider creates a service and publishes the Web Service Description Language(WSDL) file in Universal Description Discovery and Integration (UDDI). The client finds the best matching service from UDDI and then communicates to the service provider to consume that service (Bhardwaj & Sharma, 2015). REST-based services mainly rely on URI for resource identification and interaction, and HTTP for message transmission. These services are described by XML based languages such as WSDL and Web Application

* Corresponding author.
E-mail addresses: nehaa.cs.18@nitj.ac.in (N. Agarwal), sikkag@nitj.ac.in (G. Sikka), director@nitj.ac.in (L.K. Awasthi).

https://doi.org/10.1016/j.ipm.2020.102238
Received 7 January 2020; Received in revised form 4 March 2020; Accepted 5 March 2020
Available online 31 March 2020
0306-4573/

Description Language (WADL), and now generally service providers use simple natural language text to explain the functionality of service (Zhang, Wang, He, Li, & Huang, 2019).

As of 13 September 2019, more than 22,000 web services or API are published on ProgrammableWeb[1](PW) and is considered one of the popular service registries which contain various service description formats like WADL, WSDL and natural language texts (Zhao, He, & Qiao, 2018; Zhao, Wang, Wang, & He, 2018). The functionality of published services or APIs in PW have described in unstructured description texts, and due to that, it is complicated to discover appropriate services from service repositories. Therefore, it has become a challenge in service computing to discover appropriate service efficiently.

Web service search engines and web portals are the primary sources for the discovery of services. The keyword-based matching approach is mainly used by service engines, and it tends to return inaccurate and inappropriate results. Due to that discovery process of web service has low recall and low precision (Bhardwaj & Sharma, 2015). To overcome this problem, semantic web services are evolved to enhance service discovery by semantically annotating attributes of services. However, it is identified that lots of services are not annotated semantically, and it is a tedious and time-consuming task to annotate them manually (Aznag, Quafafou, Rochd, & Jarir, 2013; Nisa & Qamar, 2015).

In the last few years, web service clustering has been endorsed by various researchers to improve the performance and accuracy of web service discovery. For web service clustering, service description files are represented in vector space by using the TF-IDF (Term Frequency- Inverse Document Frequency) method, which results in a vast number of attributes. TF-IDF matrix is usually very sparse, and because of that there is a requirement to discover relevant attributes from the matrix. Therefore, for dimensionality reduction and to deal with relevant attributes, topic modeling approaches are efficient. They can deal with abundant data and considered as best techniques for dimension reduction (Bukhari & Liu, 2018; Crain, Zhou, Yang, & Zha, 2012).

The main objective of topic modeling is to determine latent and hidden semantic structure in the service by using statistical and mathematical techniques. A fascinating relationship lies among clustering, topic modeling, and dimension reduction (Crain et al., 2012). The main idea behind clustering is to group similar services based on similar functionality into the same cluster and to reduce the searching space for web service discovery. Soft clustering describes the relationship of service with multiple groups. Topic modeling assimilates the motive of soft clustering by which dimensionality reduction is achieved, and with the relevant features, services are properly plotted in the vector space. It results in a recognizable representation of services that is very effective for interpreting the themes or domains in services.

Various techniques are proposed for efficient web service discovery by utilizing the benefits of topic modeling in which web services are represented in the form of topic vectors. When services are represented in the form of topic vectors, only relevant words are selected in topics, so these techniques are considered for dimensionality reduction (Aznag et al., 2013; Guo et al., 2016; Zhao, Wang, et al., 2018). In the domain of web service, mainly Latent Dirichlet Allocation (LDA) and its amended versions are used in the literature (Cao, Liu, Liu, & Tang, 2017; Chen, Wang, Yu, Zheng, & Wu, 2013; Shi, Liu, Zhou, Tang, & Cao, 2017). However, traditional topic modeling (like LSA, LDA etc) are not efficient for short text due to following reasons (Cheng, Yan, Lan, & Guo, 2014; Jipeng, Zhenyu, Yun, Yunhao, & Xindong, 2019; Yin & Wang, 2014):

1. Short text suffers from data sparsity problem.
2. Due to the lack of repeated words, it is a challenging task for traditional topic modeling approaches to find correlated words in documents or service descriptions.
3. In the short text, context is very limited, so it is challenging to determine the meaning of enigmatic words in that.

In web service description files or Web API, the functionality of services is represented in the short text. So when we apply traditional topic modeling techniques on the assortment of text, they suffer from the data sparsity and high dimensionality problems. In the domain of web service clustering and discovery, researchers have proposed various methods by using topic modeling methods for dimensionality reduction and vectorized representation of services. However, short text topic modeling approaches are not exploited. So our contribution to research is as follows:

1. Crawl the web services or API from PW (i.e., online web repository) to generate datasets of services.
2. Investigate various topic modeling methods to extract latent factors i.e., topics from the services so that services can be represented in the form of topic vectors and dimensionality reduction can be achieved by selecting relevant words in the topics.
3. Deploy Gibbs Sampling algorithm for Dirichlet Multinomial Mixture (GSDMM) model on datasets for the representation of services and dimensionality reduction.
4. Apply various clustering methods on the services, which are represented as topic vectors by applying topic modeling and GSDMM methods, to group similar services into the same domain.
5. Evaluate the performance of methods by applying intrinsic and extrinsic measures and determine the effective dimension reduction and clustering approach for web service.

This paper is organized as follows. Section 2 firstly provides an evaluation of topic modeling techniques indicating its advantages and limitations. After that, the related work in the domain of web service clustering with topic modeling techniques is discussed. Section 3 elaborates the proposed methodology by describing datasets, methods for feature representation and dimensionality

[1] https://www.programmableweb.com/

reduction, clustering algorithms, and evaluation measures used in this study. In Section 4, experimental setup and parameter settings for the models are described. Comparative analysis of topic modeling techniques with clustering algorithms is also performed in this section. Section 5 concludes the paper and throws light on future work.

2. Related work

In this section, we divide literature into two subsections i.e., topic modeling techniques and recent work on web service clustering using topic modeling. In the first subsection, different topic modeling techniques are investigated with their pros and cons. The second subsection presents the literature based on current work in web service clustering and discovery using topic modeling techniques.

2.1. Topic modeling techniques

Topic modeling techniques are referred to as soft clustering methods to group both documents (service description), which contain similar terms, as well as terms that are contained by a set of similar documents. Firstly LSA (Latent Semantic Analysis) (Landauer, Foltz, & Laham, 1998) technique was proposed whose main objective is to decompose the term-document matrix by singular value decomposition technique to learn latent features. LSA topic modeling technique is the simplest method that gives better results as compared to vector space representation, but the main drawback of this method is that it is not able to discover the multiple meaning of the term. Due to the low-level representation of services provided by LSA, it is not efficient, and there is a lack of interpretable embedding. In this model, there is no vigorous background of statistical methods.

To overcome the limitations of LSA, a new model was proposed, which was the probabilistic variant of LSA and having a statistical foundation named PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999). The probabilistic model of PLSA is also called an aspect model which is used to analyze co-occurrences of terms and documents in data. The main problem with PLSA is that this model does not assign probabilities to the documents. When the number of documents increases in the PLSA model, there is a linear growth in the number of parameters of the model. Beacuse of that PLSA model suffers from overfitting issues.

Blei, Ng, and Jordan (2003) presented LDA (Latent Dirichlet Allocation) method, which is a very prominent generative probabilistic topic modeling technique. The primary aim of this model is to represent service files (documents) as a random mixture of latent features i.e., set of topics, and each topic is represented by terms of service files or documents. The main advantage of LDA is that it results in the topical mixture of each document or service, which is in the form of the probability distribution over the topics. Although the LDA model overcomes the deficiencies of PLSA, it is not able to snatch the correlation among topics.

The Correlated Topic Model (CTM) is another probabilistic topic modeling method that extends LDA by discovering the rich topic correlation (Blei & Lafferty, 2006). The key idea used in this model to determine correlation is to replace a topic proportion of the basic LDA method with the logistic normal distribution. The basic difference between LDA and CTM model is that in LDA, it is assumed that topics are independent due to dirichlet prior on topic distribution. CTM adopts logistic normal distribution over topic proportion, which determines the correlation among topics by using covariance matrix (Aznag et al., 2013) and topic weights are generated from Gaussian distribution. Despite this enhanced richer representation, CTM is inefficient for extensive data. Also, the pairwise correlation modeling imposes the inference complexity of $O(T^3)$ where T is the number of topics.

In topic modeling techniques, it is required to tune hyper-parameters to observe the optimal number of topics. This leads to several problems like high training time and the unsubstantial efficiency of the model. In these models, another drawback is that there is no sharing of topics among documents in the collection of a dataset. For example, if we need to fetch the documents regarding the ‘university funding’, then the topics should be drawn for ‘education’ and ‘finance’ both. To address these issues, HDP (Hierarchical Dirichlet Processing) topic modeling technique is introduced (Teh, Jordan, Beal, & Blei, 2005). Basically, this model is a non-parametric Bayesian model which capture the optimal number of topics by itself and grant the permission to mixture components to be shared among the documents. HDP model is having a hierarchy of Dirichlet Processes, so the tree-like structure is having a limitation that it restricts the flexibility of the model.

Topic modeling techniques (LDA, CTM, etc.) do not behave efficiently for short text. Dirichlet Multinomial Mixture (DMM) model is based on the assumption that each document or service file can be represented with only one topic. For short text, this assumption suites more than the assumption that each document is a mixture of topics (Jipeng et al., 2019; Nigam, McCallum, Thrun, & Mitchell, 2000). Yin and Wang (2014) presented Gibbs Sampling algorithm for Dirichlet Multinomial Mixture (GSDMM) model for short text clustering to meet the challenges of short text clustering like sparsity and high dimensional problem. This model is an amended version for LDA, which makes the initial assumption that there will be one topic corresponding to one document.

2.2. Web service clustering using topic modeling

Basically, by applying topic modeling techniques three objectives can be achieved i.e., 1) Semantic meaning is defined by document-term matrix 2) By representing documents in terms of topics, dimensionality reduction is derived 3) Documents in the form of topics can be represented in vector space so that similarity can be computed. A roadmap was presented to extract the semantic relationship among words by utilizing the benefits of Latent Semantic Analysis (Steyvers & Griffiths, 2007). Barnaghi, Cassar, and Moessner (2010) evaluated the performance of PLSA and Latent LDA topic modeling techniques to discover the hidden semantics from an assortment of service description files and cluster services according to these hidden factors. Results show that LDA performs better than PLSA with high accuracy. PLSA is not able to provide high accuracy due to the narrow concepts used in this technique in

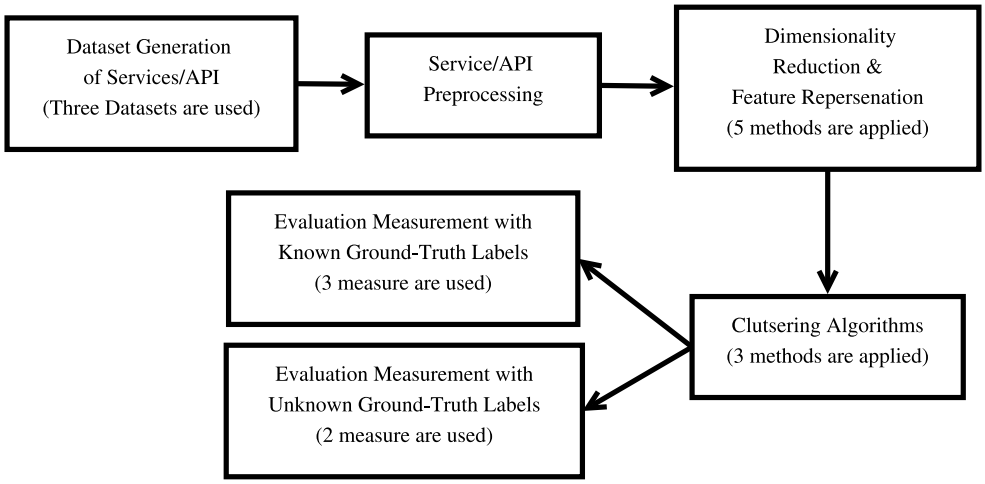


Fig. 1. Proposed pipeline process for service/API clustering.

the training phase.

In web service discovery, when keyword-based matching approaches are used, it provides low recall and precision. Keyword-based discovery is inadequate to capture accurate and appropriate services. To get rid of these limitations, topic modeling techniques (PLSA, LDA, CTM) are analyzed (Aznag et al., 2013). The objectives accomplished in this paper are to use topic modeling techniques as efficient dimension reduction techniques and to find the semantic meaning between word (terms) and topics. It is proved that CTM and LDA provide better performance than PLSA and K-Means clustering algorithm. Bukhari and Liu (2018) presented an efficient search engine that can retrieve the most relevant web services in a short span of time. For generating topic vectors, LDA technique is used, which reduces the dimension of the service representation. After that, K-Means clustering is applied to topic vectors for enhancing the accuracy score of clustering. Onan, Bulut, and Korukoglu (2017) proposed an improved ant algorithm for improving the quality of clustering and utilized the LDA technique as a dimension reduction technique for representing documents in a compact way. In Zhang et al. (2019), a model is proposed to retrieve accurate services according to the user query. In this proposed model, services are represented by using LDA technique. In papers Zhao, Wang, et al. (2018) and Zhao, He, et al. (2018), LDA technique is incorporated with word embedding techniques to improve the efficiency of service clustering. Fletcher (2018) deployed the HDP technique to extract topics from service description and user requirements to enhance the discovery of services.

3. Methodology

In this section, we present different topic modeling techniques adopted for feature representation and dimensionality reduction and evaluate their performance based on standard evaluation measurements. The proposed pipeline process for Service/API Clustering is shown in Fig. 1. Different techniques are used at each pipeline step of Service/API Clustering. Three datasets are generated by crawling desired fields from the PW repository. After that, the various method for feature representation, dimensionality reduction, and clustering are examined on generated datasets. Intrinsic and extrinsic evaluation measurement parameters are utilized for determining the effectiveness of applied methods. Table 1 shows the diversions at each step of the pipeline process. In the rest subsections, we elaborate on each step of our methodology shown in Fig. 1.

3.1. Dataset generation of services / API

ProgrammableWeb (PW) is the popular online repository for web services, and till now, more than 22,000 services are published on this repository. This repository is adopted by many researchers for the creation of a dataset for services or API (Bukhari & Liu, 2018; Zhao, He, et al., 2018). So for our study, we have also endorsed this repository. In Python, Scrapy[2] is a powerful and effective application framework that is designed for data scraping and web site crawling. This is used for extracting data for diverse applications like information retrieval and processing, data mining, or history archive. For extracting desired fields of service (i.e., Service/API name, Category, and Description) from PW, we have designed our crawler in Python with the help of Scrapy and extracted 19,697 services from this repository with more than 500 categories.

Fig. 2 shows the representation of service i.e., Amazon Product Advertising API in PW. For our study, mainly three fields are required, which are the name of service/API, it's category and description i.e., in the form of short text. These desired fields are extracted by crawling more than 800 web pages in PW, and extracted data is saved for study. The category field is obtained for evaluating the performance of our methodology. This field acts as ground truth-labels. By comparing the predicted category and these

[2] https://scrapy.org/ .

Table 1
Outline of datasets, dimensionality reduction and feature representation methods, clustering algorithms, and evaluation measures used in this study.

Datasets
Online repository (ProgrammableWeb) is crawled for services/API and by electing different categories, three datasets are generated. Details are as follows:
DS1 Dataset generated by taking seven categories elected in Zhang et al. (2019)
DS2 Dataset generated by taking ten categories elected in Zhao, Wang, et al. (2018)
DS3 Dataset generated by taking eight categories elected in Pang et al. (2019)
Methods
Dimensionality Reduction and Feature Representation:
DRFR1 LSA (Latent Semantic Analysis)
DRFR2 LDA (Latent Dirichlet Allocation)
DRFR3 CTM (Correlated Topic Model)
DRFR4 HDP (Hierarchical Dirichlet Processing)
DRFR5 GSDMM (Gibbs Sampling algorithm for Dirichlet Multinomial Mixture model)
Clustering Algorithms:
CA1 K-Means Algorithm
CA2 Affinity Propagation Algorithm
CA3 Agglomerative Algorithm
Evaluation Measurement
Extrinsic Measures:
NMI Normalized Mutual Information
AMI Adjusted Mutual Information
ARI Adjusted Rand Index
Intrinsic Measures:
SI Silhouette Index
DB Davies-Bouldin Index

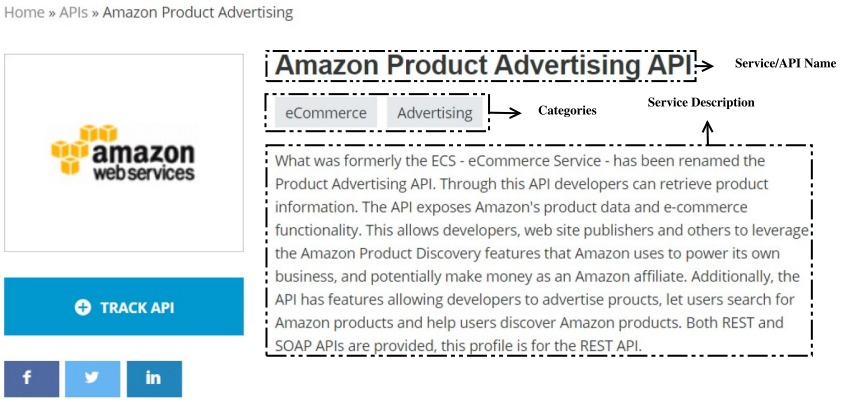


Fig. 2. Example of Web Service/API in ProgrammableWeb.com.

ground truth-labels, extrinsic evaluation measures are performed to evaluate clustering performance. The description field is the main ingredient for conducting an experiment. From this field, the functionality of service is extracted. After applying pre-processing, features are represented in vector space, and dimensionality reduction is achieved. Then on the basis of similarity, clusters are created.

For conducting our methodology, we generate three datasets from the whole extracted data of PW in which different categories or domains are elected. We have referred different research papers in which the PW repository is selected for dataset generation and categories of our datasets are elected accordingly. For our first dataset i.e., DS1, we have taken 1514 services with seven categories. For the selection of categories in DS1, categories which are taken in Zhang et al. (2019), are referred. Dataset 2 i.e., DS2, contains 4330 services with ten categories or domains. The categories of this dataset are selected by referring to the dataset generated in Zhao, Wang, et al. (2018). For the third dataset i.e., DS3, 4360 services are taken by selecting eight categories. For the category selection of this dataset, the dataset created in Pang, Zou, Gan, Niu, and Zhang (2019) is considered. Detailed description of number of services in each category of different datasets i.e DS1, DS2 and DS3 is shown in Tables 2–4 respectively. Table 5 presents the properties of generated datasets in brief.

3.2. Service/API preprocessing

The data preprocessing step is essential because it helps in enhancing the quality of the dataset. This step should be performed properly; otherwise, there will be an impact on the efficiency of feature representation and clustering algorithms. Description fields of each service of datasets are preprocessed before dimensionality reduction and feature representation. Following steps are followed for

Table 2
Number of services/APIs per category in DS1.

Category/Domain	No of Services/APIs
Media	102
Music	208
Photos	208
Transportation	277
Travel	253
Video	282
Weather	184

Table 3
Number of services/APIs per category in DS2.

Category/Domain	No of Services/APIs
Advertising	254
E-Commerce	577
Education	284
Email	314
Enterprise	453
Financial	884
Games	256
Government	401
Mapping	425
Social	482

Table 4
Number of services/APIs per category in DS3.

Category/Domain	No of Services/APIs
Mapping	425
Social	482
E-Commerce	577
Search	281
Tools	815
Messaging	614
Video	282
Financial	884

Table 5
Description of datasets used in study.

Dataset	No of Services	No of Features	Domains
DS1	1514	664,481	7
DS2	4330	1,941,699	10
DS3	4360	1,904,304	8

preprocessing of description fields of each service:

- *Removal of irrelevant characters and Stopwords:* In this step, firstly, irrelevant characters like punctuations marks, URLs, newline, special symbols, and quotes are removed from descriptions because they don't play any role for service clustering. After that, unnecessary words like 'a', 'an', 'the', 'what' etc. are removed. For removal of stopwords 'nltk'[3] package of Python is utilized.
- *Lemmatization:* Lemmatization is used to convert a word into its root form. In our experiment, we have used 'spaCy'[4] package for lemmatization. This is the new and effective way for lemmatization in Python, and it comes with pre-built models that can parse text efficiently. The main benefit of this package is that it also determines the part-of-speech of words and convert them into its root form.
- *Tokenization:*After applying above mentioned steps, the description of services is tokenized by 'word_tokenize'[5] function of nltk package in Python 3.7.

[3] https://www.nltk.org/.
[4] https://spacy.io/api/lemmatizer.
[5] https://kite.com/python/docs/nltk.word_tokenize.

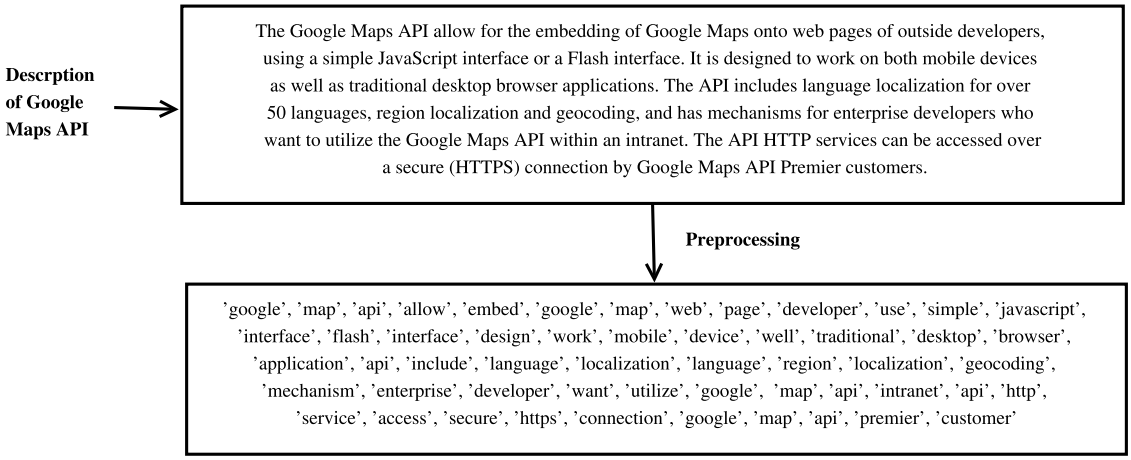


Fig. 3. Example to show the results after preprocessing of Web Service/API's description.

Fig. 3 illustrates the output of preprocessing steps on the description of Google Maps API. This example proves that after applying the mentioned preprocessing steps, we are able to generate relevant features from the description of service.

3.3. Dimensionality reduction and feature representation

In this study, our primary focus is to evaluate the efficiency and effectiveness of five topic modeling techniques used for dimensionality reduction and feature representation of files combined with three clustering algorithms. Topic modeling techniques have the supreme power by which dual tasks can be performed. Firstly, while representing files in the form of topics, the dimensionality of features is reduced. Secondly, files can be represented in the vector space in the form of topics. So topic modeling techniques are the best way to accomplish dimensionality reduction as well as feature representation of files. Table 1 outlines the various methods used for this step, and for the sake of convenience, these methods are coded as *DRFR1-DRFR5*. For visualizing different topic modeling techniques with respect to web service, this section is divided into two subsections: (1) Notations used in topic modeling (2) Functionality of various topic modeling techniques.

3.3.1. Notations used in topic modeling

For topic modeling techniques we need three set of variables:

1. *Service Description Files*: Suppose, $S = S_1, S_2, S_3, S_4$.....,S_n is a set of service description files which contains n number of services.
2. *Distinct Terms*: $W = W_1, W_2, W_3, ...,W_m$, a set of distinct terms from service description files, contains m number of terms. These variables are observing variables.
3. *Topics*: $T = T_1, T_2, T_3, ...,T_k$ is the collection of topics that are derived from service description files and total number of topics are k. These are the latent or hidden variables.

Notations that are used in topic modeling techniques are shown in Table 6.

3.3.2. Functionality of various topic modeling techniques

In brief, the functionality of dimensionality reduction and feature representation methods i.e., topic modeling techniques, outlined in Table 1, is discussed in this section. This section represents the various hyperparameters and generative process of different topic modeling techniques.

1. *LSA:* It is the backbone of topic modeling whose main purpose is to decompose the term-service matrix to learn latent features by using a singular value decomposition technique. In the simplest version of LSA, the service-term matrix A contains the raw count of the number of times jth term appeared in ith service. Now the raw count is replaced by TF-IDF because raw count does not work well for discovering the importance of terms in each service file. The singular value decomposition of term-service matrix A' can be represented by Eq. (1) (Crain et al., 2012).

$$A' = USV' \tag{1}$$

where U is a term-topic matrix, S is the topic importance matrix, and V is a topic-service matrix. We can apply rank-K approximation for dimensionality reduction on term-service matrix A'.
2. *LDA:* LDA is the most popular topic modeling technique which generates topics per file and terms per topic by using Dirichlet distributions. Assume that there are total k topics across all the service files. The generative process of LDA is as follows:
(a) For each service, generate a distribution on topics θ from Dirichlet distribution with parameter α.

Table 6
Notations used in topic modeling techniques.

A	Service-term matrix with the dimension $n \times m$
μ	Mean
k	Number of pre-assumed topics
Σ	Covariance Matrix
α	Per service topic distribution
β	Per topic term distribution
θ	Topic distribution for service S_n
ϕ	Term distribution for topic k
$\mathcal{N}$	Logistic normal distribution
η	Topic weight vector
H	Symmetric Dirichlet
G_0	Base distribution over topics
L	Number of Iterations to train model
P	Conditional Distribution
$T_{n,m}$	Topic assigned to mth term of nth service
$W_{n,m}$	mth term of nth service

(b) For each topic, generate a distribution on terms ϕ from Dirichlet distribution with parameter β.
(c) For each term W_m in service S_n:
(i) Pick a topic $T_{n,m}$ from a Multinomial distribution with parameter θ_n.
(ii) Pick a term $W_{n,m}$ from a Multinomial distribution with parameter $\phi_{T_{n,m}}$

In Fig. 4 i.e., a graphical model of LDA, $W_{n,m}$ is highlighted because it is the only observable variable in the model while others are latent.

3. *CTM:* The key point in CTM topic modeling technique is that for calculating topic distribution for service logistic normal distribution is utilized rather than dirichlet distribution used in LDA. Suppose there are k topics over all service files. The generative process of CTM is as follows:
(a) For each service S_n in a set of service description files S, generate a k-dimensional vector η_n from multivariate Gaussian distribution where $\mu and \Sigma$ are mean and covariance matrix with k dimension and $\eta_n \sim \mathcal{N}(\mu, \sum)$
(b) For each term W_m in a service S_n:
(1) Generate topic assignment $T_{n,m}$ from multinomial distribution over $f(\eta_n)$ where $f(\eta) = \frac{\exp(\eta_n)}{\sum_{i=1}^{n} \exp(\eta_i)}$
(2) Generate term $W_{n,m}$ from a Multinomial distribution with parameter $\beta_{T_{n,m}}$

Graphical model of CTM is shown in Fig. 5.

4. *HDP:* Basically HDP is a Bayesian nonparametric topic model in which there is no requirement to specify the number of topics in advance. The number of topics is figured out by the assortment of files at the time of posterior inference. To construct a Bayesian nonparametric topic model, topic distribution for any nth service i.e θ_n, that is generated by finite dirichlet, is substituted by G_n which is the distribution over topics by using dirichlet process (Blei, Carin, & Dunson, 2010). The generative process of HDP is as follows:
(a) Generate base distribution over topics G_0 by using dirichlet process i.e $G_0 \sim DP(\gamma, H)$
(b) For each service S_n in a set of service description files S, generate per service distribution over topics G_n by using dirichlet process i,e $G_n \sim DP(\alpha, G_0)$
(c) For each term W_m in a service S_n:
(i) Generate the topic for term i.e $T_{n,m} \sim G_n$
(ii) Generate term by using Multinomial distribution i.e $W_{n,m} \sim Multinomial(T_{n,m})$

Graphical model of HDP is shown in Fig. 6.

5. *GSDMM:* GSDMM technique is mainly designed for clustering the short text. Due to the lack of recurring words and limited text, topic modeling techniques coded as DRFR1-DRFR4 (outlined in Table 1) are not able to provide significant results. The key point in this model is that it is assumed that there will be one to one correspondence between topic and service. Suppose there are k topics overall service files. The generative process of GSDMM technique is a follows:

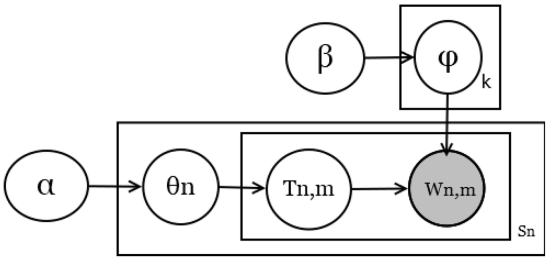


Fig. 4. Graphical model of LDA .

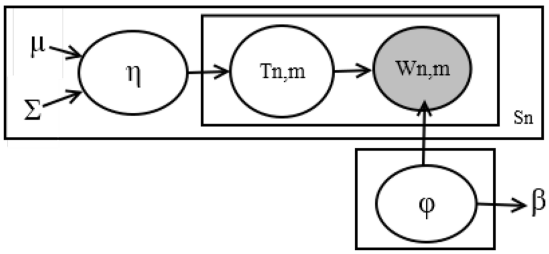


Fig. 5. Graphical model of CTM.

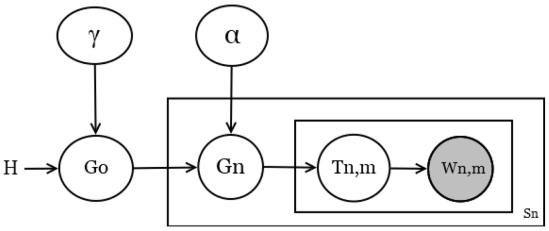


Fig. 6. Graphical model of HDP.

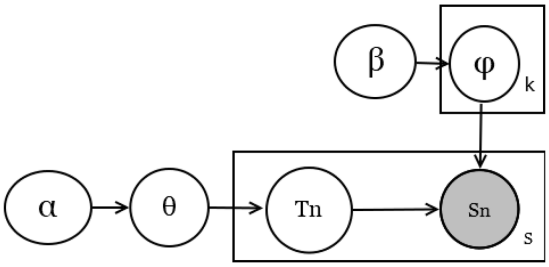


Fig. 7. Graphical model of DMM.

(a) For each service S_n randomly assign topic T_n, where number of topics ranges from i to k.
(b) For 1 to L:
  (i) For each service S_n re-assign topic T_n according to the conditional distribution such that

$$T_n \sim P\left(T_n = T | T_{\neg n, S_n}\right) \tag{2}$$

In Eq. (2), $T_{\neg n}$ means that while re-assigning the topic to service, the previous topic assignment is removed. Eq. (2) is determined by using Dirichlet Multinomial Mixture (DMM) model. The graphical model of DMM is shown in Fig. 7.

In our methodology, the GSDMM technique is utilized as a dimension reduction and feature representation technique for services. As the description of service is generally found in the form of short text, so to meet challenges of short text clustering (discussed in 1), this technique is proposed for web service. For my best knowledge, this technique is not used for dimension reduction and feature representation in the domain of web service or API.

3.4. Clustering algorithms

After applying topic modeling techniques for dimensionality reduction and vector space representation, it is necessary to group similar services in a cluster, so that services can be discovered easily. In this study, we have selected three clustering algorithms i.e K-Means, Affinity Propagation and Agglomerative algorithms which are commonly used in the literature (Bukhari & Liu, 2018; Cong, Fernandez, Billhardt, & Lujak, 2015; Curiskis, Drake, Osborn, & Kennedy, 2019; Fletcher, 2018). The main advantage of assembling similar service in a cluster is that it reduces the space over which search for service is applied, and because of that performance of web service discovery is enhanced. In our study, we have used 'sklearn.cluster'[6] for applying different clustering techniques.

K-Means clustering is a viral and most desired partitioning-based clustering algorithm for grouping similar objects in a cluster. We

[6] https://scikit-learn.org/stable/modules/clustering.html/clustering .

have trained our K-Means clustering model (CA1) for 20,000 iterations, and distance is computed using the Euclidean metric. Steps of K-Mean clustering are illustrated in Algorithm 1. As in CA1, the number of clusters i.e., K is a user-defined parameter so to find the optimal number of clusters in this method, we have used the Elbow method.

Agglomerative algorithm (CA2) is a hierarchical clustering technique in which points or services are firstly assumed in individual clusters, and at each iteration, merging is performed among closest or similar services. Steps of the Agglomerative algorithm are illustrated in Algorithm 2. In our model for agglomerative clustering, distance is measured by using Euclidean distance and linkage criteria is 'ward'. In this algorithm also, we need to find the optimal number of clusters. So with the help of dendrograms, the optimal number of clusters is determined.

Affinity Propagation Algorithm (CA3) is a message-passing clustering algorithm. The critical point of this algorithm is that it finds the number of clusters and center points of clusters (exemplars) itself. Like K-Means and Agglomerative algorithm, it is not required to find the optimal number of clusters. In this algorithm, messages for calculating responsibility and availability are transmitted between the pairs of services until there is a possibility of convergence. If a message is sent from i to k then, the responsibility metric tells the relevancy of k as an exemplar for i, and the availability metric shows that for i, how much it will be pertinent to elect j as an exemplar. We have trained our Affinity Propagation Algorithm (CA3) for 2000 iterations, and distance is computed using the Euclidean metric. Steps of CA3 are illustrated in Algorithm 3.

3.5. Evaluation measurement

Evaluation measures, used for determining the efficiency of the clustering algorithm, mainly fall into two classes, intrinsic (internal) and extrinsic (external) measures. Intrinsic measures calculate the closeness of the clusters by using some similarity measure techniques like the Silhouette score, Davies-Bouldin Index, etc. There is no need for ground-truth labels for intrinsic measurement techniques. Such measures interpret the variations in intra-cluster and inter-cluster. In the extrinsic measures, ground-truth labels are required for calculations. Some extrinsic methods like precision, recall, F1-score, accuracy, are strongly correlated to the ordering of cluster labels to ground-truth labels. Measures like mutual information and the rand index are independent of the absolute value of labels, so they are more relevant.

In our study, we have used NMI, AMI, and ARI as extrinsic measures (Curiskis et al., 2019; Yin & Wang, 2014). Mutual information is the measure of mutual agreement between two assignments. There are two versions of this measure i.e., Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI). Assume there are two label assignments of any document X and Y. Then entropy of these labels can be defined by Eq. (3) and (4). Here $P(i)$ and $P(j)$ are the probabilities of that the document belongs to class X_i and Y_j, respectively.

$$H(X) = -\sum_{i=1}^{|X|} P(i)\log(P(i)) \tag{3}$$

$$H(Y) = -\sum_{j=1}^{|Y|} P(j)\log(P(j)) \tag{4}$$

The mutual information (MI) between X and Y is calculated by Eq. (5). $P(i, j)$ is the probability that document belongs to both classes X_i and Y_j.

$$MI(X, Y) = \sum_{i=1}^{|X|}\sum_{j=1}^{|Y|} \log\left(\frac{P(i, j)}{P(i)P(j)}\right) \tag{5}$$

NMI score scales the value of MI between 0 to 1 by computing the ratio of the result of Eq. (5) and harmonic mean of entropies of X and Y defined in Eqs. (3) and (4). So NMI(X, Y) can be defined by Eq. (6).

$$NMI(X, Y) = \frac{MI(X, Y)}{\sqrt{H(X)H(Y)}} \tag{6}$$

Rand Index (RI) mainly helps to determine what is the percentage of correct decisions. Suppose X is the ground-truth labels and Y are predicted labels in the clustering model. Let x is the number of services in pairs that are lying in the same set of X and in the same set of Y. y is the number of services in pairs that are lying in a different set of X and in a different set of Y. So mathematically, the rand index can be measured by Eq. (7).

$$RI(X, Y) = \frac{x + y}{{}^{S}C_2} \tag{7}$$

Here S is the total number of services in the dataset. Adjusted Rand index(ARI) basically ignores permutation and find out how the two assignments are similar. ARI and AMI are the adjusted versions of RI and MI respectively. ARI and AMI can be computing by Eq. (8) and (9).

$$ARI(X, Y) = \frac{RI - Expected\_RI}{\max(RI) - Expected\_RI} \tag{8}$$

1: **procedure** K-MEANS(ST, K) (Where ST is services represented in vector space in form of topics and K is number of domains/clusters)
2: Firstly elect K points as initial centre points.
3: **repeat**
4: Generate K clusters by measuring similarity on the basis of Euclidean distance.
5: Recompute the centre points for each domain/cluster.
6: **until** No alteration in centre points
7: **end procedure**

Algorithm 1. K-Means Clustering (CA1).

procedure AGGLOMERATIVE ALGORITHM(ST, K) (Where ST is services represented in vector space in form of topics and K is number of domains/clusters)
    Firstly compute distance among services ST on the basis of Euclidean distance and generate distance matrix.
    Consider each single service as individual cluster or domain.
    repeat
        Merge two services which are having minimum distance.
        Re-generate distance matrix.
    until Number of clusters are equal to K
end procedure

Algorithm 2. Agglomerative Algorithm(CA2).

1: **procedure** AFFINITY PROPAGATION ALGORITHM(ST) (Where ST is services represented in vector space in form of topics)
2: Firstly compute distance among services ST on the basis of Euclidean distance and generate distance matrix.
3: Consider each single service as individual exemplar.
4: **repeat**
5: Send messages in pair for calculating responsibility and availability.
6: Re-assign exemplar by using responsibility and availability metrics.
7: **until** There is a possibility of convergence
8: **end procedure**

Algorithm 3. Affinity Propagation Algorithm(CA3).

$$AMI(X, Y) = \frac{MI - Expected\_MI}{\max(MI) - Expected\_MI} \tag{9}$$

Expected_RI and *Expected_MI* are the expected values of RI and MI respectively.

In our study, we have used Silhouette Coefficient Score and Davies-Bouldin Index as intrinsic measures, which are generally used in literature (Bukhari & Liu, 2018; Reddy, Tripathy, Nimje, Ganga, & Varnasree, 2018; Yahyaoui & Own, 2018). Silhouette Coefficient Score is an intrinsic evaluation measure for cluster validation. This evaluation measure determines the pairwise inter-cluster distance and intra-cluster distance to validate results. If the Silhouette Coefficient Score is high, it means the model has defined better clusters. If a_i^j is mean distance between any service s_i and other services in any cluster j and b_i^j is mean distance between any service s_i and other services in nearest cluster then silhouette coefficient for that service (SI_i^j) can be defined by Eq. (10).

$$SI_i^j = \frac{b_i^j - a_i^j}{\max(a_i^j, b_i^j)} \tag{10}$$

By taking the average of silhouette coefficient score of all services in cluster j, silhouette coefficient score SI^j for cluster j is calculated. To determine the Silhouette coefficient score for all clusters we need to take the average of all cluster's silhouette coefficient score. Silhouette coefficient score lies in the range of -1 to $+1$ and a good clustering algorithm will be having a score close to 1.

Davies-Bouldin Index (DB) is another intrinsic evaluation measure in which inter-cluster and intra-cluster distances are computed to determine how much similar are services within the same and different clusters. A lower value of this index represents that services are similar within the clusters and different from other clusters. So a lower value of DB shows that the clustering algorithm is determining better results.

We have used 'sklearn.metrics.cluster'[7] package of Python to determine mentioned measures for performance evaluation of clustering algorithms.

4. Experiments setup and results

In this section, we present the results of our study. We have performed our experiments on windows 10 system with i7-8750H processor having 6 cores, 9 MB cache size, and 8GB RAM. For dataset generation, PW online repository is crawled. The detailed process of dataset generation is discussed in Section 3.1. For analyzing results, this section is divided into three parts. In the first part, the optimal number of topics is determined for each topic modeling techniques. The second part presents the parameter setting of different models. In the last part, the performance of clustering is evaluated based on extrinsic and intrinsic measures.

4.1. Determination of optimal topics for topic modeling techniques

Topic modeling techniques are the prominent tools for analyzing text documents and determining predictive and latent topic representation in the assortment of documents. However, there is a deep-rooted assumption that the latent features identified by the model are generally relevant and valid. As it is unsupervised training techniques, so evaluating such assumptions is a troublesome task. Conventional approaches that are used for evaluation of topic modeling techniques are eyeballing, intrinsic evaluation metrics, human judgments, and extrinsic evaluation metrics. By utilizing these evaluation techniques, the optimal number of topics can be determined for which model provides the best results.

Intrinsic evaluation metrics determine the internal coherence of the topics. Perplexity is one of the intrinsic evaluation metrics, which is widely used for topic model evaluation by measuring normalized log-likelihood of the test set. In web service modeling also perplexity computation is adopted by many researchers to find the optimal number of topics (Bukhari & Liu, 2018; Wang, Gao, Ma, He, & Hung, 2017). However, perplexity is unable to estimate topics according to the understandability and determination of humans. Another intrinsic evaluation measure topic coherence calculates the score of a topic by evaluating the degree of semantic similarity between the words in that topic who scored high value. So in this paper, we have calculated the optimal number of topics by using topic coherence score and the topic models are tuned accordingly. As per my knowledge, in the domain of web service clustering, the topic coherence score parameter is not used earlier, but it is used in document clustering (Abolhassani & Ramaswamy, 2019; Cheng et al., 2014). For calculating the topic coherence score of a topic model, the following steps need to be performed:

1. Determine the top n highly occurring terms W_1, W_2....W_j in each topic T.
2. For all pairs of words determined in step 1, calculate the pairwise score and sum up all scores to determine the coherence score for that particular topic. Coherence C of a particular topic can be computed using Eq. (11).

$$C = \sum_{i<j} score(W_i, W_j) \tag{11}$$

3. After calculating the coherence score of all topics, take the mean of that for computing coherence score of a model.

[7] https://scikit-learn.org/stable/modules/classes.html/module-sklearn.metrics.cluster .

For the computation of the coherence score of different topic modeling techniques, Coherence Model[8] library of 'gensim' package in Python is used. For datasets (DS1-DS3), topic modeling models (DRFR1-DRFR3 and DRFR5) are trained, and the optimal number of topics are determined by analyzing coherence score. In the range of topics from 10 to 90, the coherence score for each model on each dataset is calculated. In the graph between topics and coherence score, if we elect optimal topics for the model when coherence score is high, then it is a good choice, but there can be the probability of repeated terms. So when there is a first declination in coherence score while increasing the number of topics, then at that point value of the number of topics can be taken as optimal. This strategy provides a better result. DRFR4 i.e., HDP model, determines the optimal number of topics itself. The graphs between topics and coherence scores are shown in Fig. 8–11 for different topic modeling techniques. These graphs are used to determine the optimal number of topics so that web services can be efficiently represented in the form of topics in vector space and dimensionality of features is also reduced. Fig. 8 shows the coherence graph for LSA model on dataset DS1. In this figure, after the number of topics 20, the coherence score is declined. So the optimal number of topics for this model is 20. Figs. 9–11 shows the coherence graph for LDA, CTM and GSDMM models on dataset DS3, DS2 and DS1 respectively. In these graphs, after the number of topics i.e. 30, 20 and 40, the value of coherence score is reduced. So the optimal number of topics for LDA, CTM and GSDMM models are 30, 20 and 40 respectively. Dotted line in Figs. 8–11 shows the first declination in coherence score and represents optimal number of topic for that model. Table 7 shows the optimal topics of each model of DRFR technique on datasets DS1-DS3 evaluated by the coherence score.

4.2. Parameter setting

DRFR1-DRFR5 models are trained according to the following parameters:

1. *LSA:* There are no hyper-parameters in this model. Optimal number of topics for DS1, DS2 and DS3 are set 20, 20 and 10 respectively.
2. *LDA:* In this model, we set hyper-parameters $\alpha = 0.1$ and $\beta = 0.01$. Optimal number of topics for DS1, DS2 and DS3 are set 20, 20 and 30 respectively. Total 200 iterations with these hyper-parameters are executed to train the model.
3. *CTM:* In this model, we set hyper-parameters $\alpha = 0.1$ and $\eta = 0.01$. Optimal number of topics for DS1, DS2 and DS3 are set 20, 20 and 10 respectively. Total 200 iterations with these hyper-parameters are executed to train the model.
4. *HDP:* In this model, we set hyper-parameters $\alpha = 0.1$ and $\gamma = 0.1$. Total 200 iterations with these hyper-parameters are executed to train the model. There is no need to pass the number of topics as it finds optimal topics itself.
5. *GSDMM:* In this model, we set hyper-parameters $\alpha = 0.1$ and $\beta = 0.1$. Optimal number of topics for DS1, DS2 and DS3 are set 40, 50 and 30 respectively. Total 20 iterations with these hyper-parameters are executed to train the model.

4.3. Discussion

We have taken two cases for evaluation of DRFR techniques with clustering algorithms i.e., (1) When the ground-truth labels are known (2) When ground truth labels are not known. So in the first subsection, we analyze the performance of DRFR techniques with clustering algorithms based on three evaluation measures. The second subsection presents the evaluation of the method based on two evaluation measure criteria in which ground-truth labels are not required. All these evaluation measure criteria are already discussed in Section 3.5 in detail. Analysis of dimensionality reduction achieved by different DRFR techniques is also elaborated.

4.3.1. Performance evaluation with known ground-truth labels

In this section, we validate the results of DRFR techniques with clustering algorithms based on extrinsic evaluation measures. We have treated the category field of datasets as ground-truth labels. Extrinsic evaluation measures evaluate the performance of clustering by comparing predicted cluster labels and ground truth-labels.

Table 8 shows the performance of DRFR techniques with clustering algorithms based on NMI, AMI, and ARI on dataset DS1. The results show that the GSDMM method with the K-Means clustering algorithm outperforms all other methods on all three evaluation measures. The results of the GSDMM method with agglomerative clustering techniques are also very close to K-Means clustering. There is approx 15% enhancement in the NMI and AMI score when results of GSDMM are compared with LDA model using K-Means clustering.

Table 9 provides the performance of DRFR techniques with clustering algorithms based on NMI, AMI, and ARI on dataset DS2. On this dataset also, the GSDMM method with the K-Means clustering algorithm outshine all other methods. In this experiment also, NMI and AMI scores are increased approx 13% when compared with LDA model. Same on dataset DS3 also, the GSDMM method with the K-Means clustering algorithm is providing better results in comparison to other methods, which is shown in Table 10. NMI and AMI scores are also improved by approx 16% in GSDMM model compared with LDA model.

By applying any technique on one dataset, if we infer any technique best one then it is not a good strategy. That's why, in this study, we have created three datasets with different domains. Short text topic modeling approach based on Dirichlet Multinomial Mixture(DMM) i.e., GSDMM has overcome the limitations of short text and provided a great enhancement in the clustering method. The results of GSDMM method with agglomerative clustering techniques are also very close to K-Means clustering on DS1 and DS2. Mostly LDA topic modeling technique is widely used in web service, and results show that there is a good increment in the evaluation

[8] https://radimrehurek.com/gensim/models/coherencemodel.html .

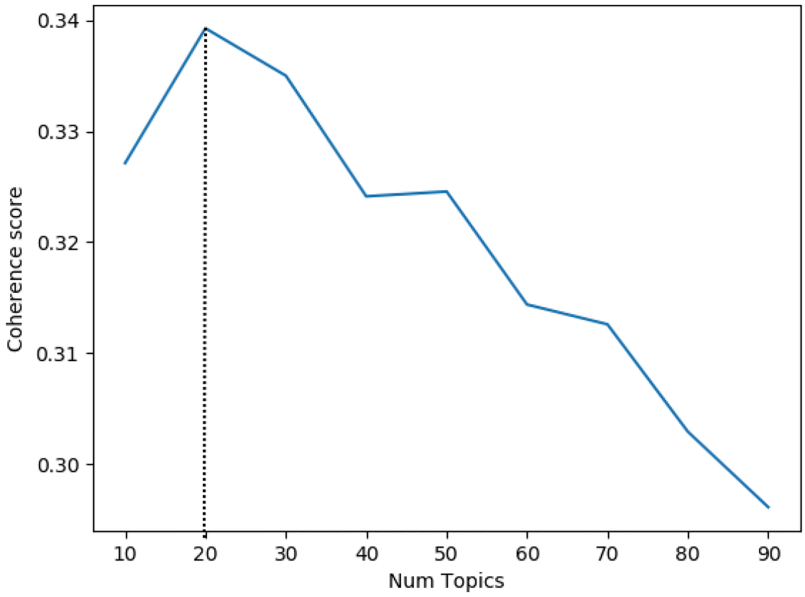


Fig. 8. Coherence Score graph of LSA model on DS1.

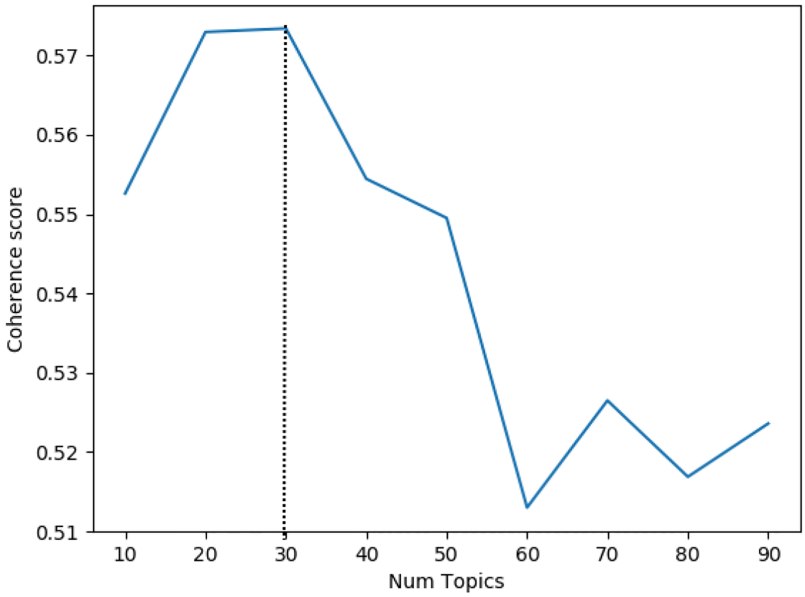


Fig. 9. Coherence Score graph of LDA model on DS3.

measure, particularly in NMI and AMI on all datasets. GSDMM model is outperforming due to the assumption that there should be one to one correspondence between topics and service description file in short text. In LDA, services correspond to a mixture of topics, and due to that LDA model is not able to perform well on the short text.

Results on all datasets show that with any DRFR technique when we apply the affinity propagation algorithm, then there is a large drop in ARI score. It is because that affinity propagation elects the number of clusters itself.

4.3.2. Performance evaluation with unknown ground-truth labels

Generally, in unsupervised clustering, there are no ground-truth labels in datasets. So intrinsic evaluation measures are needed to evaluate the performance of clustering. In the absence of ground-truth labels, before evaluating the performance of clustering, we need to identify the number of clusters in the dataset. In this study, to find the optimal number of clusters in K-Means clustering 'Elbow Method' is used, which is adopted by many researchers (Dai, Nespereira, Vilas, & Redondo, 2015; Yahyaoui & Own, 2018). This is a visualization method to provides optimal clusters by calculating the sum of squared distance (SSE) between points and

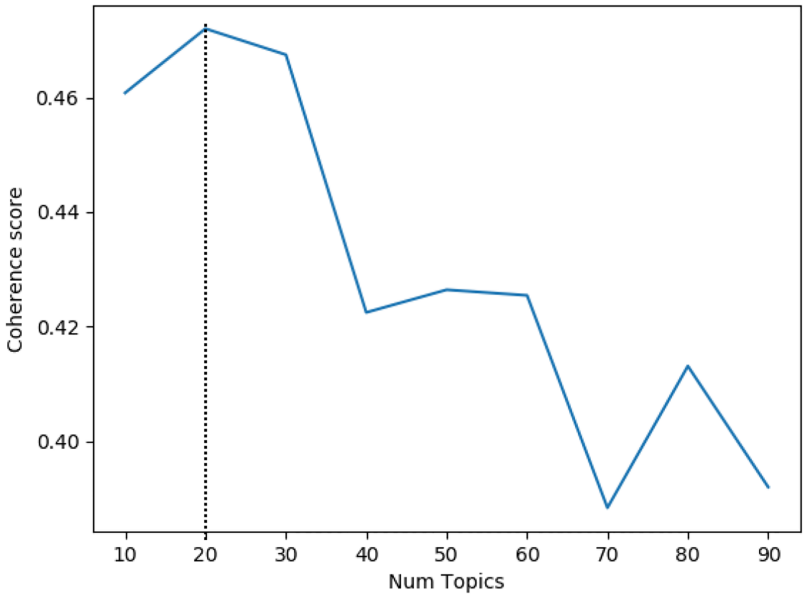


Fig. 10. Coherence Score graph of CTM model on DS2.

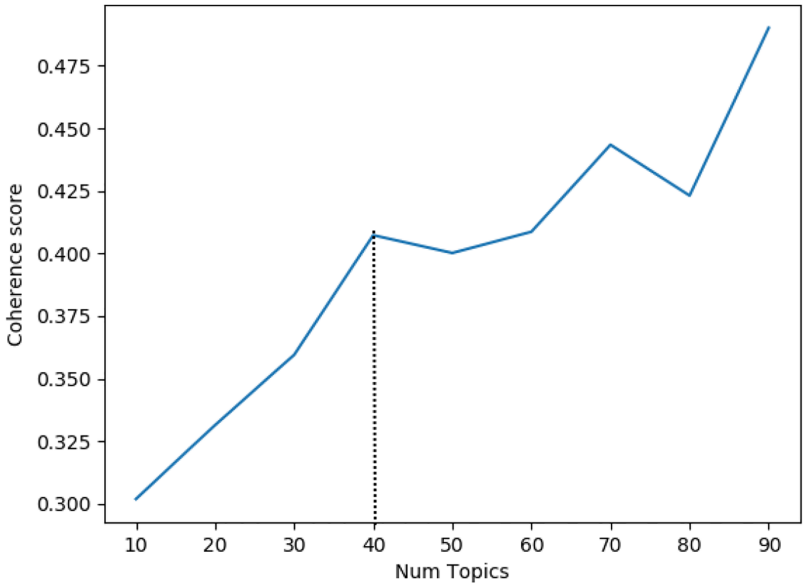


Fig. 11. Coherence Score graph of GSDMM model on DS1.

centroid of that cluster, which are generated by the K-Means algorithm. Fig. 12 shows the graph between number of clusters (in range of 2–15) and sum of squared distance which is evaluated for DS1. When there is a bend between the number of clusters and the sum of squared distance, then that point is considered as the number of optimal clusters. So in Fig. 12 optimal clusters are seven shown by dotted line. In the same way, by elbow method, the optimal number of clusters is determined for DS2 and DS3 also, which are 10 and 8, respectively.

In hierarchical clustering i.e., agglomerative clustering, the number of clusters can be find out by dendrogram. In this method hierarchy of clusters generated by agglomerative clustering is represented in the form of the tree. When the big cluster is generated, the longest vertical distance is elected through which no horizontal line is crossing. The number of vertical lines that are intersected by this newly generated horizontal line is the number of optimal clusters. Fig. 13 shows the dendrogram plotted by using 'scipy.cluster.hierarchy'[9] package in Python on dataset DS1. The dotted horizontal line is intersecting seven vertical lines and it is having

[9] https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html .

Table 7
Optimal topics of each DRFR Technique on each dataset determined by coherence score.

DRFR Techniques	Dataset	Optimal Topics
LSA	DS1	20
	DS2	20
	DS3	10
LDA	DS1	20
	DS2	20
	DS3	30
CTM	DS1	20
	DS2	20
	DS3	10
HDP	DS1	38
	DS2	45
	DS3	45
GSDMM	DS1	40
	DS2	50
	DS3	30

Table 8
Performance evaluation of DRFR techniques and clustering algorithms on DS1 based on extrinsic measures.

DRFR Techniques	Clustering Algorithms	NMI	AMI	ARI
LSA	K-Means	0.3739	0.34955	0.1745
	Agglomerative	0.4307	0.3961	0.2623
	Affinity Propagation	0.3937	0.2155	0.0597
LDA	K-Means	0.3908	0.37635	0.25585
	Agglomerative	0.3771	0.3585	0.223
	Affinity Propagation	0.3921	0.2497	0.1244
CTM	K-Means	0.3588	0.3458	0.23595
	Agglomerative	0.3143	0.2953	0.2229
	Affinity Propagation	0.3246	0.1774	0.0499
HDP	K-Means	0.3087	0.2064	0.1612
	Agglomerative	0.3059	0.2066	0.162
	Affinity Propagation	0.326	0.2121	0.16
GSDMM	K-Means	**0.4492**	**0.4329**	**0.2702**
	Agglomerative	0.449	0.4325	0.27
	Affinity Propagation	0.383	0.0541	0.1012

Table 9
Performance evaluation of DRFR techniques and clustering algorithms on DS2 based on extrinsic measures.

DRFR Techniques	Clustering Algorithms	NMI	AMI	ARI
LSA	K-Means	0.14885	0.13615	0.04805
	Agglomerative	0.2163	0.1926	0.084
	Affinity Propagation	0.2726	0.1249	0.0179
LDA	K-Means	0.36365	0.3532	0.214
	Agglomerative	0.3429	0.3293	0.1795
	Affinity Propagation	0.376	0.2384	0.0867
CTM	K-Means	0.22845	0.217	0.11835
	Agglomerative	0.1983	0.1795	0.0797
	Affinity Propagation	0.2576	0.1299	0.0196
HDP	K-Means	0.26705	0.2396	0.08665
	Agglomerative	0.2739	0.2452	0.0838
	Affinity Propagation	0.3051	0.1952	0.0691
GSDMM	K-Means	**0.4123**	**0.398**	**0.2291**
	Agglomerative	0.4121	0.3978	0.2288
	Affinity Propagation	0.3729	0.0319	0.011

the longest vertical distance. So it is observed from dendrogram that the optimal number of clusters is seven in DS1. In the same way, the optimal number of clusters are determined for DS2 and DS3 also, which are 10 and 8, respectively. Both techniques are providing the same number of clusters or domains in datasets i.e. DS1, DS2 and DS3. So it is validated that we are able to find correct number of clusters in our datasets.

Table 11 shows the performance of different DRFR techniques with clustering algorithms based on intrinsic evaluation measures i.e., SI and DB on dataset DS1. Results show that HDP with the affinity propagation algorithm is giving the highest SI, but the DB

Table 10
Performance evaluation of DRFR techniques and clustering algorithms on DS3 based on extrinsic measures.

DRFR Techniques	Clustering Algorithms	NMI	AMI	ARI
LSA	K-Means	0.1921	0.1786	0.08069
	Agglomerative	0.2259	0.2093	0.11
	Affinity Propagation	0.2822	0.143	0.0202
LDA	K-Means	0.3216	0.30285	0.1979
	Agglomerative	0.3124	0.2768	0.1431
	Affinity Propagation	0.3473	0.1959	0.0551
CTM	K-Means	0.2013	0.1963	0.1491
	Agglomerative	0.1877	0.178	0.0881
	Affinity Propagation	0.2229	0.1145	0.0176
HDP	K-Means	0.3397	0.3115	0.1564
	Agglomerative	0.3525	0.322	0.1695
	Affinity Propagation	0.3708	0.1572	0.0637
GSDMM	K-Means	**0.37365**	**0.34885**	**0.2178**
	Agglomerative	0.3432	0.3202	0.1961
	Affinity Propagation	0.3501	0.067	0.063

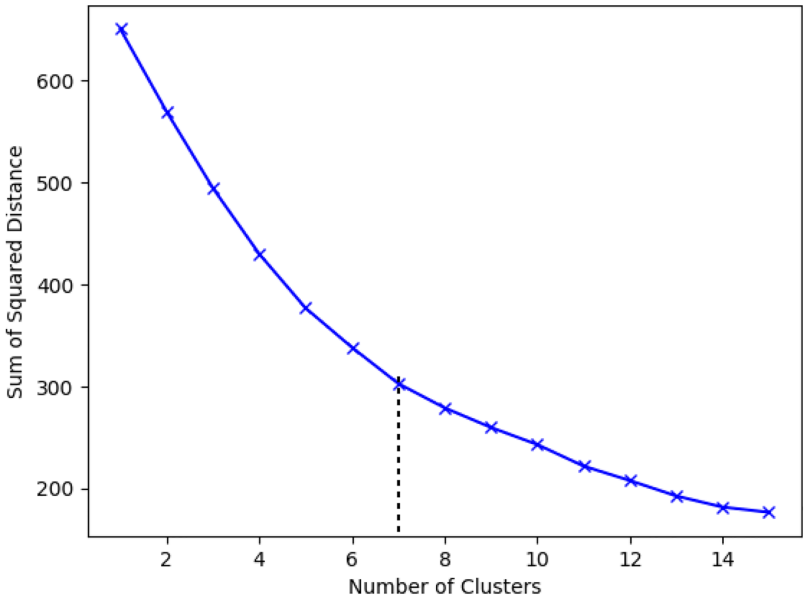


Fig. 12. Elbow method for finding optimal clusters of in DS1.

score is not satisfactory. GSDMM method with both K-means and agglomerative algorithms is providing the lowest DB index.

The performance of different DRFR techniques with clustering algorithms based on intrinsic evaluation measures on dataset DS2 is analyzed in Table 12. In this dataset, results validate that GSDMM method with agglomerative clustering algorithm is generating the highest SI score, and with K-Means DB index is lowest. SI and DB score for GSDMM technique with K-Means and agglomerative algorithms are almost the same, and there is the slightest difference in their index values.

Results of extrinsic measures on DS3 have also proved that GSDMM method with K-Means or agglomerative algorithm is producing better results as compared with other methods, as shown in Table 13. We are getting similar SI and DB indexes in GSDMM with K-Means and agglomerative. So we can conclude that in performance evaluation of DFFR techniques based on the intrinsic measure, our proposed technique that is GSDMM is yielding better results.

By evaluating the performance of DRFR techniques with different clustering algorithms based on intrinsic and extrinsic measures, it is proved that the GSDMM technique with K-Means or agglomerative clustering is enhancing service clustering as compared to LDA which is generally used in this domain. LDA mainly works well for long text in which there is frequent occurrences of words and data is not sparse.

4.3.3. Dimensionality reduction analysis

The main aim of this study is to utilize theses model as dimension reduction techniques. Table 14 presents the percentage of dimensionality reduction achieved by different DRFR techniques. Results show that LSA achieves the highest dimensionality reduction percentage on all datasets, but it is not a probabilistic topic model. There is no statistical background in this model, and there

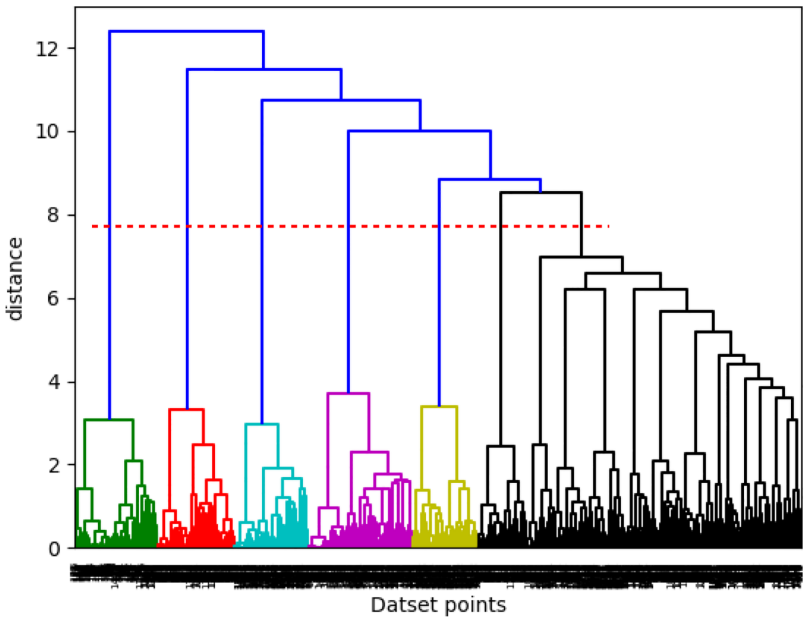


Fig. 13. Dendrogram for finding optimal clusters of in DS1.

Table 11
Performance evaluation of DRFR techniques and clustering algorithms on DS1 based on intrinsic measures.

DRFR Techniques	Clustering Algorithms	SI	DB
LSA	K-Means	0.1123	2.1175
	Agglomerative	0.09593	2.3195
	Affinity Propagation	0.07131	1.5877
LDA	K-Means	0.2934	1.5384
	Agglomerative	0.2575	1.5049
	Affinity Propagation	0.2921	1.2656
CTM	K-Means	0.1400	2.1489
	Agglomerative	0.1134	2.2105
	Affinity Propagation	0.0923	1.9762
HDP	K-Means	0.9534	0.9363
	Agglomerative	0.9509	0.91926
	Affinity Propagation	**0.9571**	0.9366
GSDMM	K-Means	0.7018	**0.8566**
	Agglomerative	0.7018	**0.8566**
	Affinity Propagation	0.1031	0.8762

Table 12
Performance evaluation of DRFR techniques and clustering algorithms on DS2 based on intrinsic measures.

DRFR Techniques	Clustering Algorithms	SI	DB
LSA	K-Means	0.0995	1.9899
	Agglomerative	0.0970	2.0773
	Affinity Propagation	0.0568	1.6806
LDA	K-Means	0.3283	1.4588
	Agglomerative	0.2897	1.4568
	Affinity Propagation	0.3194	1.2299
CTM	K-Means	0.1396	2.1201
	Agglomerative	0.1040	2.0030
	Affinity Propagation	0.0743	2.101
HDP	K-Means	0.3162	1.6318
	Agglomerative	0.2955	1.6863
	Affinity Propagation	0.4765	0.9912
GSDMM	K-Means	0.6997	**0.9000**
	Agglomerative	**0.7056**	0.9019
	Affinity Propagation	−0.0383	1.025

Table 13
Performance evaluation of DRFR techniques and clustering algorithms on DS3 based on intrinsic measures.

DRFR Techniques	Clustering Algorithms	SI	DB
LSA	K-Means	0.1892	1.4464
	Agglomerative	0.1304	1.7326
	Affinity Propagation	0.0989	1.4431
LDA	K-Means	0.2215	1.7502
	Agglomerative	0.1453	1.6348
	Affinity Propagation	0.2197	1.2095
CTM	K-Means	0.2149	1.4200
	Agglomerative	0.1442	1.6721
	Affinity Propagation	0.1037	1.7084
HDP	K-Means	0.3401	1.6813
	Agglomerative	0.3188	1.1806
	Affinity Propagation	0.2704	1.007
GSDMM	K-Means	**0.5983**	**0.9029**
	Agglomerative	**0.5983**	**0.9029**
	Affinity Propagation	0.1099	1.0982

Table 14
Dimensionality reduction achieved by different DRFR techniques.

DRFR Techniques	Dataset	Original Dimension Space	Dimension Obtained by DRFR Techniques	Dimension Reduction in Percentage
LSA	DS1	664,481	30,280	95.4431
	DS2	1,941,699	86,600	95.5399
	DS3	1,904,304	43,600	97.7104
LDA	DS1	664,481	30,280	95.4431
	DS2	1,941,699	86,600	95.5399
	DS3	1,904,304	130,800	93.1313
CTM	DS1	664,481	30,280	95.4431
	DS2	1,941,699	86,600	95.5399
	DS3	1,904,304	43,600	97.7104
HDP	DS1	664,481	57,532	91.3418
	DS2	1,941,699	194,850	89.9649
	DS3	1,904,304	196,200	89.6970
GSDMM	DS1	664,481	60,560	90.8861
	DS2	1,941,699	216,500	88.8499
	DS3	1,904,304	130,800	93.1313

is a lack of interpretable embedding. LDA, CTM, and HDP models have also reduced dimensionality with a satisfactory percentage, but they suffer from the problems of short text. In the lack of co-occurrence words, the models do not provide significant results. GSDMM model has provided the best results on all intrinsic and extrinsic evaluation measures when it is applied to different datasets. Dimensionality reduced by this model is the appropriate one because, with this dimensionality reduction percentage, the clustering algorithm i.e., K-Means or Agglomerative is providing significant results.

5. Conclusion and future work

In this paper, topic modeling techniques that are endorsed in the domain of web services are investigated and analyzed. As the description of services is in the form of short text, so to meet the challenges of the short text clustering GSDMM method is proposed. The results of performance evaluation have validated that the GSDMM method outperformed traditional topic modeling techniques that are used in web services. Due to data sparsity and high dimensionality problems, traditional topic modeling techniques are not appropriate for web services or APIs, which are described in short text form. By deploying GSDMM method for dimensionality reduction and feature representation and applying K-Means clustering to group similar services into the same domain, we can generate improved and effective results as compared to other models. The validity of topic models are verified on intrinsic and extrinsic both evaluation measures criteria. Results have demonstrated that the GSDMM method can be used effectively in the domain of web service also.

In the future, this method can be utilized to enhance the discovery process of mash-up services. In any application in which clustering to be done with short text like tweets, social mining, etc. this method can be incorporated.

CRediT authorship contribution statement

Neha Agarwal: Conceptualization, Funding acquisition, Data curation. **Geeta Sikka:** Conceptualization, Funding acquisition, Data curation. **Lalit Kumar Awasthi:** Conceptualization, Funding acquisition, Data curation.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ipm.2020.102238 .

References

Abolhassani, N., & Ramaswamy, L. (2019). *Extracting topics from semi-structured data for enhancing enterprise knowledge graphs. International conference on collaborative computing: Networking, applications and worksharing*. Springer101–117.

Aznag, M., Quafafou, M., Rochd, E. M., & Jarir, Z. (2013). *Probabilistic topic models for web services clustering and discovery. European conference on service-oriented and cloud computing*. Springer19–33.

Barnaghi, P., Cassar, G., & Moessner, K. (2010). *Probabilistic methods for service clustering. CEUR workshop proceedings: Proceedings of 4th international workshop on service matchmaking and resource retrieval in the semantic webVol. 667*.

Bhardwaj, K. C., & Sharma, R. (2015). Machine learning in efficient and effective web service discovery. *Journal of Web Engineering, 14*(3&4), 196–214.

Blei, D., Carin, L., & Dunson, D. (2010). Probabilistic topic models: A focus on graphical model design and applications to document and image analysis. *IEEE Signal Processing Magazine, 27*(6), 55.

Blei, D., & Lafferty, J. (2006). Correlated topic models. *Advances in Neural Information Processing Systems, 18*, 147.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning research, 3*(Jan), 993–1022.

Bukhari, A., & Liu, X. (2018). A web service search engine for large-scale web service discovery based on the probabilistic topic modeling and clustering. *Service Oriented Computing and Applications, 12*(2), 169–182.

Cao, B., Liu, X. F., Liu, J., & Tang, M. (2017). Domain-aware mashup service clustering based on LDA topic model from multiple data sources. *Information and Software Technology, 90*, 40–54.

Chen, L., Wang, Y., Yu, Q., Zheng, Z., & Wu, J. (2013). *Wt-lda: User tagging augmented LDA for web service clustering. International conference on service-oriented computing*. Springer162–176.

Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering, 26*(12), 2928–2941.

Cong, Z., Fernandez, A., Billhardt, H., & Lujak, M. (2015). Service discovery acceleration with hierarchical clustering. *Information Systems Frontiers, 17*(4), 799–808.

Crain, S. P., Zhou, K., Yang, S.-H., & Zha, H. (2012). *Dimensionality reduction and topic modeling: From latent semantic indexing to latent Dirichlet allocation and beyond. Mining text data*. Springer129–161.

Curiskis, S. A., Drake, B., Osborn, T. R., & Kennedy, P. J. (2019). An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*.

Dai, K., Nespereira, C. G., Vilas, A. F., & Redondo, R. P. D. (2015). Scraping and clustering techniques for the characterization of linkedin profiles. arXiv:1505.00989.

Fletcher, K. K. (2018). *A quality-based web api selection for mashup development using affinity propagation. International conference on services computing*. Springer153–165.

Guo, L., Li, Z., Yang, T., Zhang, H., Mu, D., & Li, Y. (2016). *An improved latent Dirichlet allocation method for service topic detection. 2016 35th Chinese control conference (CCC)*. IEEE7045–7049.

Hofmann, T. (1999). *Probabilistic latent semantic analysis. Proceedings of the fifteenth conference on uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc.289–296.

Jipeng, Q., Zhenyu, Q., Yun, L., Yunhao, Y., & Xindong, W. (2019). Short text topic modeling techniques, applications, and performance: A survey. arXiv:1904.07695.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*(2-3), 259–284.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning, 39*(2-3), 103–134.

Nisa, R., & Qamar, U. (2015). A text mining based approach for web service classification. *Information Systems and e-Business Management, 13*(4), 751–768.

Onan, A., Bulut, H., & Korukoglu, S. (2017). An improved ant algorithm with LDA-based representation for text document clustering. *Journal of Information Science, 43*(2), 275–292.

Pang, S., Zou, G., Gan, Y., Niu, S., & Zhang, B. (2019). Augmenting labeled probabilistic topic model for web service classification. *International Journal of Web Services Research (IJWSR), 16*(1), 93–113.

Reddy, A. J., Tripathy, B., Nimje, S., Ganga, G. S., & Varnasree, K. (2018). *Performance analysis of clustering algorithm in data mining in R language. International conference on soft computing systems*. Springer364–372.

Shi, M., Liu, J., Zhou, D., Tang, M., & Cao, B. (2017). *We-lda: A word embeddings augmented LDA model for web services clustering. 2017 IEEE international conference on web services (ICWS)*. IEEE9–16.

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of Latent Semantic Analysis, 427*(7), 424–440.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2005). *Sharing clusters among related groups: Hierarchical Dirichlet processes. Advances in neural information processing systems*1385–1392.

Wang, J., Gao, P., Ma, Y., He, K., & Hung, P. C. (2017). A web service discovery approach based on common topic groups extraction. *IEEE Access, 5*, 10193–10208.

Yahyaoui, H., & Own, H. S. (2018). Unsupervised clustering of service performance behaviors. *Information Sciences, 422*, 558–571.

Yin, J., & Wang, J. (2014). *A Dirichlet multinomial mixture model-based approach for short text clustering. Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM233–242.

Zhang, N., Wang, J., He, K., Li, Z., & Huang, Y. (2019). Mining and clustering service goals for restful service discovery. *Knowledge and Information Systems, 58*(3), 669–700.

Zhao, Y., He, K., & Qiao, Y. (2018). *St-lda: High quality similar words augmented LDA for service clustering. International conference on algorithms and architectures for parallel processing*. Springer46–59.

Zhao, Y., Wang, C., Wang, J., & He, K. (2018). Incorporating LDA with word embedding for web service clustering. *International Journal of Web Services Research (IJWSR), 15*(4), 29–44.