Mapping Lexical Gaps In Cloud Ontology Using BabelNet and FP-Growth

Mustafa M. Al-Sayed *mostafamcs@gmail.com*
Faculty of Computers and Information
Minia University
Minia, Egypt

Hesham A. Hassan *h.hassan@fci-cu.edu.eg*
Faculty of Computers and Information
Cairo University
Cairo, Egypt

Fatma A. Omara *f.omara@fci-cu.edu.eg*
Faculty of Computers and Information
Cairo University
Cairo, Egypt

Abstract

In spite of the rapid growth of cloud services, a wide spectrum of enterprises and users do not buy such services as a result of insufficient knowledge about cloud technology. On the other hand, the majority of cloud service discovery and repository frameworks depend on cloud ontologies as one of the main building blocks. Accordingly, constructing a suitable cloud service discovery mechanism will exclude an immense obstacle especially for non-expert users due to the difficulty to select suitable concepts from referenced cloud ontology. Users always prefer to compose queries using their natural languages. But, it is difficult to truly match such queries because of natural language ambiguities. In this paper, we propose a new mechanism bridging the gap between English natural language terms and concepts of the referenced cloud ontology using BabelNet knowledge base and FP-Growth mining algorithm. The contribution of this paper is two folded: firstly, classifying cloud services into their corresponding cloud ontology concepts; secondly, anticipating a system that enables non-specialized users to flexibly compose their cloud service searching queries using English natural language. We have applied the proposed mechanism on two sets of cloud services related to concepts in the referenced cloud ontology; Streaming and Multimedia (S&M), and Human Resource (HR). According to our experimental results, the proposed mechanism has achieved 90%, and 86% in the F-Score measure for classifying S&M, and HR cloud services, respectively. For matching users' queries, the results have shown that 80% of S&M and 82% of HR relevant queries have been assigned correctly.

Keywords: Cloud Ontology, Query Composition, Semantic Search, Word Sense Disambiguation, Cloud Service Discover.

1. INTRODUCTION

Due to the characteristics of cloud computing, many organizations and customers are being motivated to move their business applications to cloud computing. According to IBM survey [1], cloud computing and its related services will continue to play the key role in the future. On the other hand, Cloud Computing like other modern technologies has some challenges. Discovering the suitable cloud services represents one of these challenges. The rapid publishing of different types of cloud services using non-standard naming conventions and non-standard description languages made an crucial problem to the automatic identification of the right services [2] [3] [4] [5].

Ontology can provide a well-defined knowledge for cloud services using what is so-called cloud ontology. So, semantic technologies, embodied in the ontology, can play an important role in overcoming such challenges, in particular those are related to standardization, and improving the communication among Cloud agents (i.e., human and software) [6] [7]. Also, ontology can provide a unified taxonomy of cloud services, which facilitates the comparison process of these services [8] [9].

Therefore, cloud ontology can serve as a one of the main building blocks in the cloud service repository and discovery frameworks, where vendors can publish their cloud services in the repository, and customers can query for the suitable service [10]. It can serve as a mapping layer to improve the process of retrieving services that match customer demands. So, the discovery process can use cloud ontology to facilitate and accelerate the identification of relevant services according to customer needs [11].

On the other hand, many semantic search tool evaluations have reported that users prefer using natural languages (NLs) in composing their discovery requests, more than controlled or view-based interfaces [12]. While, composing a request using NLs may represent a certain obstacle for non-expert users (i.e., users with insufficient knowledge about cloud) due to the difficulty of selecting suitable concepts of a referenced cloud ontology [13]. A large sector of enterprises and users avoid using cloud services due to their insufficient knowledge about cloud computing [14] [15]. Additionally, using NLs for composing user requests might cause ambiguity in the meaning, which makes it difficult to really match such requests. Unfortunately, there are not user-friendly applications that enable users to discover the cloud services using NLs familiar terms with considering the ambiguity problem [16].

The work in this paper introduces a mechanism to bridge the gap between the English NL terms and cloud ontology concepts using BabelNet knowledge base [17] and FP-Growth frequent pattern detection algorithm [18], which is an efficient and scalable mining algorithm used to extract associated key-terms. This paper contributes to classify cloud services into their corresponding cloud ontology concepts and also to enable non-expert users to flexibly compose their discovery queries using English NL terms that may differ based on their working fields.

BabelNet is a semantic network, which connects concepts and named entities in a huge knowledge base of semantic relations. This network includes about 14 million entries. Each entry represents a given meaning and its related synonyms. BabelNet is an integration of fifteen resources. WordNet, Open Multilingual WordNet, Wikidata, Wikitionary, Wikipedia, and Omegawiki are the most important of these resources (see Fig. 1) [17]. So, BabelNet can be considered as a reference for many terminologies related to new technologies, such as cloud computing. Therefore, BabelNet may be useful to be used in our proposed mechanism to achieve the required convergence between English NL terms and cloud ontology concepts.

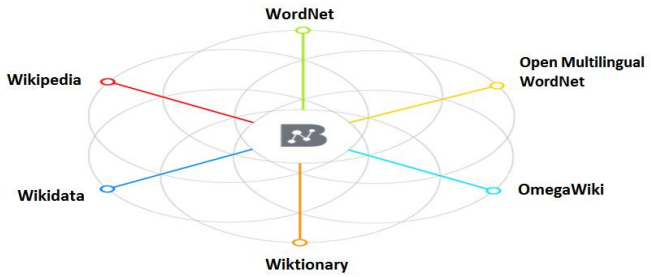

FIGURE 1: Important Resources of BabelNet Knowledge Base [17].

Also, BabelNet classifies synonyms into a list of predefined domains [19]. This may contribute to solve the problem of ambiguity in the meaning. From our point of view, this classification is not enough in case of synonyms related to cloud technology because synonyms of a cloud ontology

concept may belong to many domains. For example, identifying synonyms related to the concept “streaming and multimedia (S&M)” may be difficult as they may be fallen under many BabelNet domains, such as Music, Games & video games, Media, and Education. On the other hand, synonyms related to some cloud concepts, such as Human Resource (HR), Productivity, Sales and Marketing, and Project management, cannot be differentiated as they all are classified under one BabelNet domain, which is "business, economics, and finance". In this paper, those synonyms will be classified into new domains that are more specific and related to cloud ontology concepts.

This paper is organized as follows; the related work is discussed in Sections 2. The proposed mechanism and the experimental results are described in Sections 3, and 4, respectively. Finally, the conclusion and future work are presented in Section 5.

2. RELATED WORK

Many researchers have used ontology to solve many problems. For instance, authors in [20] has developed a web service ontology to describe web service domains. They depended on Term Frequency-Inverse Document Frequency (TF-IDF) to build the ontology by looking at WSDL files’ content. For modeling concepts and relationships extracted from web resources, Giuseppe F. and Sabrina S. [13] have proposed a web service selection system that exploits the fuzzy formal concept analysis. By selecting conceptual terms instead of using strict syntax formats (i.e. using a navigation approach), users can build their requests.

According to the cloud environment, the discovery process remains a hard problem due to non-standardized naming conventions, and heterogeneous types and features of services. But, this does not preclude the presence of many approaches, such as [4], [21], [2], [3], and [22], to address such problems using ontology.

In an attempt to achieve a high accuracy degree of searching about cloud services, Abdullah A., et al. [4] have proposed a search engine based on ontology and cosine similarity. The aim of this engine is to automatically identify cloud service categories by detecting concepts related to cloud services from web sources in a real environment.

Also, Shengjie G. and Kwang M. [23] have developed a new cloud service search engine, that is named CB-Cloudle, with a crawler for each provider. Authors used k-means clustering algorithm to accelerate the search process and discover groups of similar cloud service.

In [4], and [23], authors did not consider the insufficient knowledge of many users about cloud computing concepts. They provided their discovery engines to users in controlled and view-based interfaces, which are based on ontology concepts.

To provide users with a flexible data access within a constrained time and to release them from the need to IT expertise, Martin G., et al. [24] have discussed how Optique project [25] contributes to solve such problem. Optique project relies on Ontology Based Data Access (OBDA) to provide an automated end-to-end connection to data sources. OBDA allows users to formulate intuitive queries of familiar terms, and to translate these queries into syntactic queries over data sources.

Although the provisioned flexibility of using terms of free NL as an input query is considered a significant advantage, it is also a major difficulty. This may increase the difficulty to match terms with the underlying data. In an attempt to achieve the required convergence between user free terms and ontology concepts, properties, and entities, Authors in [12] have combined between Named-entity recognition, word sense disambiguation, and ontology-based heuristics together in one semantic search approach.

Although authors in [24] and [12] are interested to provide a semantic search and flexible data access for non-expert users, they are not concerned about cloud services and their challenges.

In sum, the current cloud service discovery studies and systems have considered that experience and sufficient knowledge about cloud technology are necessary prerequisites to users for discovering cloud services. Also, these studies allow users to discover services by using only controlled and view-based interfaces, which lack the required flexibility to build discovery queries [16]. As a result, a large sector of users, who have little knowledge about cloud and who prefer composing their queries using NL terms, goes away from the cloud services.

Because users mostly do not possess necessary concepts and knowledge to compose queries for obtaining the suited services, there are two usually practiced scenarios. According to the first scenario, a static set of queries are previously defined; while the second scenario relies on using an expert to translate non-expert users' queries to the formal format. The first is limited, while the second takes a long time until the expert writes special purpose queries and optimizes these queries for efficient execution [26].

Therefore, it is important to support cloud service discovery techniques with the automatic composition of queries using NL terms. But, this can represent a certain obstacle due to the difficulty of really matching the user query, which sometimes is ambiguous in the meaning [13].

In this paper, we propose a flexible ontology-based cloud service discovery and classification mechanism to solve such problems. The proposed mechanism depends on FP-growth algorithm, and BabelNet knowledge base to achieve the end-to-end connection between users and registers of cloud services. It would enable users to rapidly formulate intuitive discovery queries using familiar vocabularies and conceptualizations. This would relieve the burden of query composition and then enhance the effectiveness of the cloud service retrieval. Also, the proposed mechanism would contribute to classify newly registered cloud services based on their text description.

3. THE PROPOSED MECHANISM

The objective of the new proposed mechanism is to allow users to compose their queries using English NL terms that are familiar in their working fields. The idea to achieve that is to scan the input queries for detecting properties attached to concepts of referenced cloud ontology. All these concepts together represent the hierarchical taxonomic structure of the available services in a cloud service repository, while the attached properties represent a set of associated key-terms (i.e., keywords or key-expressions) that are collected from the available cloud services during the learning phases. At the end, the mechanism should return to the users a list of available services, which match the input queries, to select from. Also, the mechanism allows new cloud services, which are planned to be added to the repository, to be classified into the suitable concepts. We will demonstrate the effectiveness of the mechanism by conducting experiments.

3.1 Concepts' Properties Extraction

The objective of this step is to extract associated key-terms from a set of available cloud services for each concept in the referenced cloud ontology. These terms will be attached to concepts as properties, which will be used in the mapping phase to achieve the required convergence between the NL terms and the ontology concepts. In this section, we will illustrate how to extract these properties using suitable tools and algorithms.

As shown in Fig. 2, for each concept in the referenced cloud ontology, the textual descriptions of related services provisioned by a set of well-known cloud vendors are collected. Then, the following phases will be applied:

3.1.1 Phase (A): Automatic Detection for Expressions

In this phase, real expressions related to the specified concept will be detected and collected together with their synonyms according to the following steps:

1. *TextRank* algorithm [27], which is a graph-based ranking model for processing text documents, will be applied on each service description to extract a list of candidate expressions (i.e., phrases of two or three words). This algorithm is a common key-phrase extraction algorithm that provides a high degree of readability among other automatic summarization algorithms [28] [29].

2. *BabelNet knowledge base*, which is a very large encyclopedic semantic network and dictionary, will be applied on the generated list, obtained from step (1) above, to exclude the fake expressions (e.g., phrases without any entry in BabelNet).

Then, for each service description, replace space between words of real expressions with underscore. For example, replace “cloud computing” with “cloud_computing” to be treated after that as tokens.

3.1.2 Phase (B): Construction of Cloud Domains Dictionary

The main function of this phase is the construction of our cloud dictionary to avoid ambiguity in the meaning. Entries of this dictionary are the most common terms related to each concept of the referenced cloud ontology. These terms will be collected together with their synonyms under what so-called cloud ontology domains, where each domain refers to a specific concept in the referenced cloud ontology. The construction of our dictionary is discussed in the following steps:

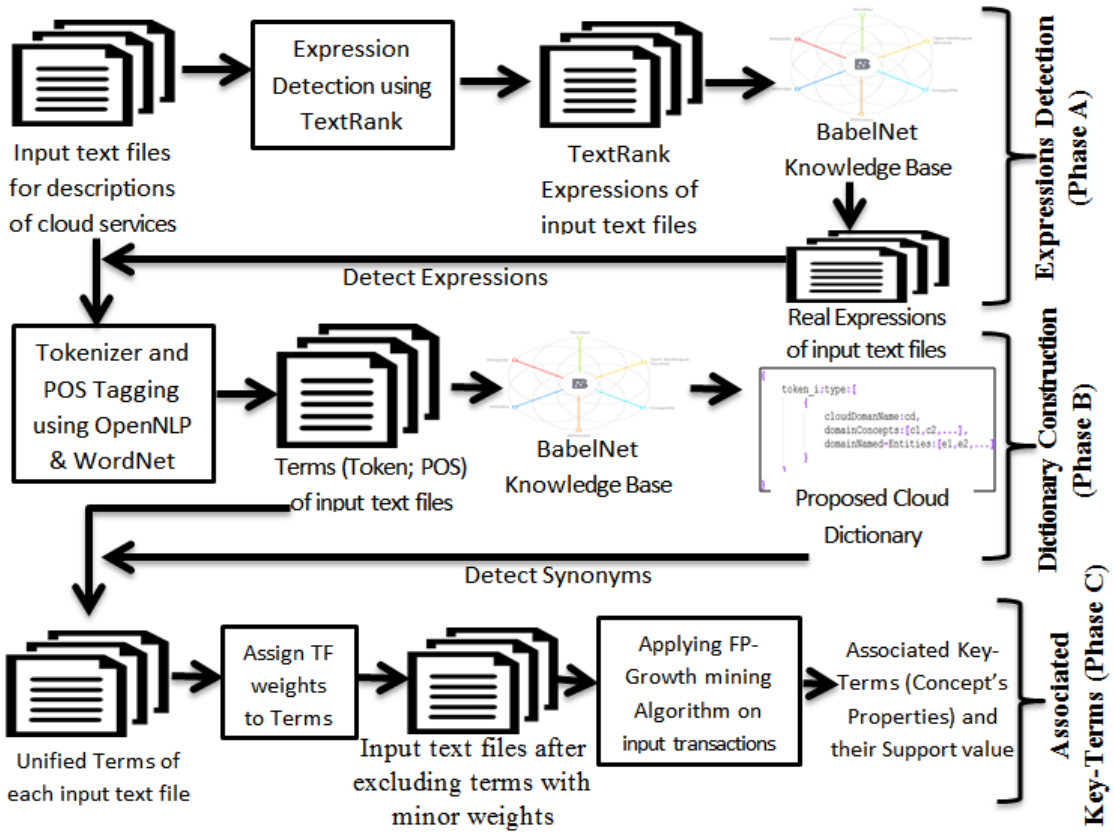

FIGURE 2: Proposed Mechanism to extract properties of a cloud ontology concept.

3. Apply *OpenNLP toolkit*[1] [30] on each service description for tokenization and part-of-speech (POS) tagging. Tokens will be converted to its base by using WordNet, then concatenated with their POS tagging in so-called terms “token;POS” (e.g., “broadcast;verb”,

[1] OpenNLP is a public and easy to use machine learning based toolkit for processing NL text.

“cloud_computing;expression”, "distributed;adjective", "system;noun", and "securely;adverb"). We will exclude secondary tokens (e.g., cc/coordinating conjunction, cd/cardinal number, and ym/symbol) by keeping verbs (i.e., vb/base verb, vbd/past verb, vbg/verb-ing, vbn/past participle verb, vbp/non-3rd person singular present verb, and vbz/3rd person singular present verb), nouns (i.e., nn/noun, and nns/plural noun), entities (i.e., nnp/singular proper noun for entities, and nnps/plural proper noun for entities), adjectives (i.e., jj/adjective with excluding comparative and superlative adjectives), and adverbs (i.e., rb/adverb with excluding comparative and superlative adverbs).

4. To maintain the small size of the dictionary and therefore accelerate the search process, which is a recursive process (see Fig. 4), the generated terms that have a low frequency (e.g., less than 20% of the number of available services in the learning dataset) in the services descriptions as a whole will be ignored. Relevant synonyms of the remaining real terms (i.e., terms that have entries in BabelNet) will be collected using BabelNet knowledge base. These synonyms will be classified according to BabelNet into domain concepts and domain named-entities. As shown in Fig. 3, terms and their synonyms will be stored in our cloud dictionary using JSON format, where *token_i* represents token name and $i \in$ {1: number of all tokens}, *type* $\in$ {noun, verb, adjective, adverb, expression}, and *cd* $\in${all concepts of the referenced cloud ontology}.

For a given input text (i.e., input query, or input cloud service description) to be classified, the absence of a property or its synonyms does not mean the absence of this property. For example, suppose that “broadcast;verb” is property related to the cloud ontology concept "S&M", and “spread” is one of its synonyms. This property and its synonym are not exist in the given input. But, the term “spread;verb” has the synonym “propagate”, which is exist in the given input. So, the search process inside our generated cloud dictionary should be recursive as shown in Fig. 4.

```
{
  token_i;type:
    {
      cloudDomanName:cd,
      domainConcepts:[c1,c2,...],
      domainNamed-Entities:[e1,e2,...]
    }
}
```

FIGURE 3: Data Structure of our Cloud Dictionary using JSON format.

3.1.3 Phase (C): Associated Key-Terms Detection

During this phase, properties of concepts of the referenced cloud ontology will be defined by detecting the associated key-terms, which are the most common terms across all descriptions of the available services in the learning dataset specified to each concept. Unfortunately, there are some terms appear with the same spelling but have different meaning (i.e., homonym), while some other terms are different in spelling but sharing the same meaning (synonym). Therefore, the extracted key-terms may suffer from the existence of some fake key-terms due to homonym terms, and the absence of important key-terms due to synonym terms. Our generated dictionary will be used in this phase to overcome such problems, where synonyms are collected using BabelNet and homonyms are avoided by classifying these synonyms into cloud domains.

To extract these properties, we will apply the following steps:

5. As a result of step (4), each service description is represented by a text file of a set of real terms (considering terms “broadcast;verb” and “broadcast;noun” are different). Then, apply the dictionary on these files to unify terms using synonym relations. For example, all synonyms (in file#i to file#*n*, where ***n*** is number of files) of term “broadcast;verb” (in file#i) will be replaced by this term. As a result, all terms of the same meaning in all the ***n*** files will be unified in spelling and meaning.

6. Frequent terms of each file generated from the previous step will be extracted. Then, the largest number of these terms that can be associated with the highest support value across all files will be specified as properties. This process may be computationally complex. So, an efficient and scalable mining algorithm, which is called FP-Growth [18], will be exploited to extract these properties. This algorithm provides a better performance, which outperforms other popular algorithms for the same purposes [31].

Finally, the obtained properties will be used to distinguish services of a particular concept in the referenced cloud ontology. According to the ontology definition, such services are treated as individuals of the specified concept.

The proposed mechanism can be applied to all concepts, which have such individuals, to obtain their properties. So, some concepts (with individuals) will have properties and others not (without individuals). Regarding the latter, the proposed mechanism can provide properties to such concepts (i.e., super-concepts) by obtaining the intersected properties of their sub-concepts recursively, as shown in pseudo code of Fig. 5.

3.2 Mapping Inputs to Cloud Ontology Concepts

The proposed mechanism will enable users with a little knowledge about cloud technology to discover a list of services, which its functionalities may match the intent of users. Users may not know about cloud technology, but they own enough knowledge about their working fields. So, they can compose queries using familiar NL terms that can describe the required functionality in details (more details mean more accurate results). Obtained properties of each concept in the referenced cloud ontology will be matched with terms that form the user input query using the generated cloud domains dictionary. Mapping linguistic terms of the input query to cloud ontology concepts may provide an end-to-end connection between users and cloud services registers.

```
1  String TrackingSynonyms(TokenCheckedList,
       Dictioanry,InputText,PropertyName;
       PropertyType,SpecifiedDomain){
2  term=PropertyName;PropertyType
3  if(TokenCheckedList.contains(term))
4    return null
5  TokenCheckedList.add(term)
6  JSONArray relevantDomains=Dictionary.get(term)
7  if(!relevantDomains.isEmpty()){
8    JSONObject domain=(JSONObject)relevantDomains.
       get(SpecifiedDomain)
9    if(domain!=null){
10   JSONArray synonyms=(JSONArray)domain.get("
       domainConcepts")
11   for(Object syn:synonyms)
12     if(InputText.indexof(syn)>-1)
13       return syn
14     else if(TrackingSynonyms(TokenCheckedList,
       Dictioanry,InputText,syn)>-1)
15       return syn
16   synonyms=(JSONArray)domain.get("domainNamed-
       Entities")
17   for(Object syn:synonyms)
18     if(InputText.indexof(syn)>-1)
19       return syn
20     else if(TrackingSynonyms(TokenCheckedList,
       Dictioanry,InputText,syn)>-1)
21       return syn
22   }else return null
23   }
24 return null
25 }
```

FIGURE 4: Tracking synonyms of a given term according to a specified cloud domain.

```
1  Property[] SuperConceptPropertiesDetection (Concept
       con){
2   if(con.properties!=null)return con.properties
3   ArrayList<Property[]> prprts
4   for (Concept concept:con.subConcepts){
5     if((temp=concept.properties)==null)
6       temp=SuperConceptPropertiesDetection(concept)
7     if(temp!=null)prprts.add(temp)
8   }
9    if (prprts.size()>0)return commonProperties(prprts)
10 return null
11 }
```

FIGURE 5: Pseudocode to obtain properties of super-concepts.

As shown in Fig. 6, the user query (as an input) will pass through a set of steps in order to be mapped to the intended ontology concept. As a first step, TextRank and BabelNet will be applied to the input query for detecting expressions (as discussed in section 3.1/Phase A). In the second step, OpenNLP and WordNet will be used to obtain a set of terms in the form "*token;POS*" (as discussed in section 3.1/Phase B/Step 3) to be matched with concepts' properties. In the third step, starting from the root concept of the referenced cloud ontology, the method "MappingInputQuery-ToCocnept" will be used to recursively return the intended concept, which its properties achieve the highest matching degree with the obtained terms. This method depends on the developed dictionary to avoid ambiguity caused by synonym or homonym terms. The recursion process will stop once obtaining concept that has cloud services as individuals.

The matching degree is measured using relative entropy (H_{rel}) as shown in the following equation [32].

$$H_{rel} = \sum_{i}^{n} \frac{P_i \log P_i}{\log_n} \qquad H_{rel} \in [0,1]$$

, where ***n*** is the number of concept's properties, and P_i is the probability of property ***i***. In other words, H_{rel} measures how these properties are uniformly distributed inside the given user input query. For example, the maximum H_{rel} of matching query terms with six properties occurs when these properties are distributed in this query with probability equal 1/6.

For classifying new services, the same three steps can be used to identify their relative concepts.

```
//the input is a service description or a user
    query
ModifiedInput=expressionsDetection(Input)
terms=Tokenizer_and_POSTagging(ModifiedInput)
MappingInputToCocnept(terms,RootConcept)
........
Concept MappingInputToCocnept(terms,concept){
 if(concept.isIndividualConcept())return concept
 HashMap csd
 Dictionary=readDictionary(DomainName)
 for (Concept con:concept.getSubConcepts){
   MatchingDegree=MatchingTerms(terms,con.
    getProperties(),con.getCloudDomainName(),
    Dictionary)
   csd.put(con.getName(),M_Degree) }
 return MappingInputToCocnept(terms,
    MaxMatchingDegreeConcept(csd))
}
double MatchingTerms(terms,properties,DomainName,
    Dictionary){
 for(Property prty:properties){
   count[i]=getFrequency(terms,prty)
   temp=TrackingSynonyms(Dictionary,terms,prty,
    DomainName)
   count[i++]+=getFrequency(terms,temp)
   totalCount+=count  }
 for(int i=0;i<properties.size();i++)
   if(count[i]>0){
     pi=count[i]/totalCount
     RelativeEntropy+=(pi*log(pi))/log(properties
    .size()) }
 return -RelativeEntropy;
}
```

FIGURE 6: Pseudocode to map an input to ontology concepts.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Cloud computing is a modern technology. Therefore, it is difficult to find a specialized resource for vocabularies related to this technology. They are scattered over many resources. BabelNet is the combination of many important resources. So, from our point of view, this makes BabelNet as the preferred resource for this technology.

We have applied our proposed mechanism on two sets of services related to concepts “S&M”, and “HR” in the referenced cloud ontology. Where, S&M includes services for storing, editing, encoding, and streaming multimedia (e.g., audio, video, on-demand gaming...etc.). While HR concept includes services to allow payroll management, to assign candidates for jobs, to manage benefits (e.g., medical insurance, pension plans, vacation...etc.), and to manage the maintenance and Development for the employees of an organization.

For evaluating the performance of the proposed mechanism, a dataset of seventy-six services descriptions (thirty S&M services, thirty-six HR services, and ten other services) has been collected from popular cloud providers. Fifteen S&M services and eighteen HR services of this dataset are used as two learning datasets to extract properties of S&M concept and HR concept, respectively. The remaining services are used as a testing dataset to be mapped to S&M or HR concepts. Each service is represented by a textual file containing a full description and is named with the service provider name. Providers of services in the learning datasets are shown in Table (1).

In the following paragraphs, the previous phases, which are defined in Section 4.1of the proposed mechanism will be applied to extract properties of S&M and HR concepts.

S&M Providers		HR Providers	
Azure	Telestream	BarbeloGroup	Amcheck
Telvue	Streambox	Trinet	CjchrServices
StreamShark	Serverroom	Insperity	InfinitiHR
Anvato	Brightcove	Aon	Prestigepeo
JwPlayer	Adobe	Ajg	AllevityHR
Wowza	WebcastCloud	Paychex	AlliedEmployer
Kollective		AccessPoint	AlignHumanResources
Zencoder		BasicOnline	DHR
Primcast		AlphaStaff	HRsolutions

TABLE 1: Providers of S&M and HR cloud services in the learning datasets.

As a result of applying Phase (A) on the learning dataset, all real expressions have been detected. Table (2) shows expressions that have been detected in S&M file “Azure” and HR file “Amcheck”, as an example.

Azure S&M Service File	Amcheck HR Service File
content_delivery	human_capital_management
flash_player	applicant_tracking_system
quick_preview	employee_satisfaction
dynamic_packaging	customer_service
optical_character	benefit_management
microsoft_silverlight	streamlined_system
content_protection	insurance_broker
emotion_recognition	employee_self-service
surveillance_footage	employee_benefit
streaming_content	employee_retention
customer_support	real-time_information
microsoft_playready	workforce_management
catch-up_tv	retirement_plan
on-the-fly_encryption	
smooth_streaming	
aes_encryption	

TABLE 2: Expressions detected in the descriptions of “Azure” S&M and “Amckeck” HR cloud services using TextRank algorithm combined with BabelNet.

As a result of applying Phase (B)/Step (3), all files from Phase (A) have been converted into terms. Fig. 7 shows a snap shot from terms of files “Azure” and “Amcheck”.

As a result of applying Phase (B)/Step (4), our cloud dictionary for terms, which have been obtained from Phase (B)/Step (3), has been generated. Terms and their synonyms in this dictionary are classified into two cloud domains; S&M and HR. Fig. 8 represents a snap shot from that dictionary.

Regarding Phase (C)/Step (5), the generated dictionary has been applied on all files from Phase (B) to unify terms by replacing their synonyms, as discussed previously.

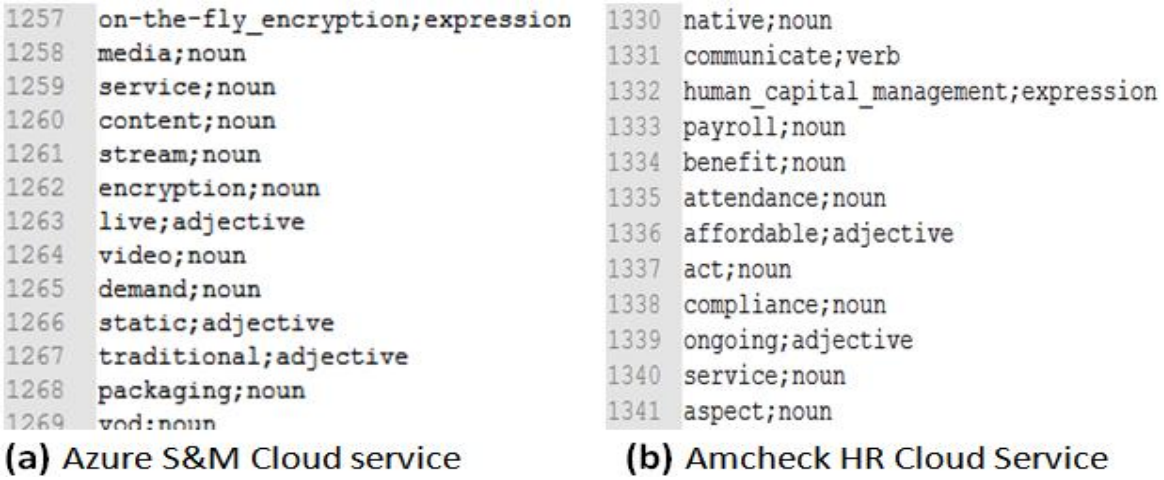

FIGURE 7: A snap shot from terms of Azure S&M and Amcheck HR cloud services using OpenNLP and WordNet.

```
{
.....
"on-the-fly_encryption;expression":{
        "cloudDomanName" : "S&M",
        "concept":["on_the_fly_encryption", "otfe", "real_time_encryption"],
        "entity":[ ] },
"encryption;noun":{
        "cloudDomanName" : "S&M",
        "concept":["encoding", "cipher"],
        "entity":["encryption_(album)"] },
"media;noun":{
         "cloudDomanName" : "S&M",
         "concept":[
                  "memory", "storage", "computer_memory", "media_(computer)",
                  "real_storage", "computer_data_storage", "main_memory",
                  "media_ut", "media_(communication)", "medium", "communication",
                  "news_media", "journalism"],
         "entity":[
                  "medes", "indo_european", "media_(ancient_country)",
                  "media_(region)","illinois","bhedia","bhediya","g_-_star_africa",
                  "grand_production", "castra_media", "media_(castra)",
                  "pennsylvania","media_station_(septa)","medja","delta_sagittarii",
                  "media_(star)","media_(album)",
                  "media_(ak-83)","media_(automobile_company)"] },
.....
"payroll;noun":{
        "cloudDomanName":"HR",
        "concept":["paysheet", "payroll_department"],
        "entity":["payroll_(film)"] },
"attendance;noun":{
        "cloudDomanName":"HR",
        "concept":["attending"],
        "entity":[ ]},
"human_capital_management;expression":{
         "cloudDomanName":"HR",
         "concept":[
                  "human_resource_management", "hr", "hrm",
                  "aims_of_human_resource_management"],
         "entity":[ ]},
.....
}
```

FIGURE 8: A Snap Shot From The Cloud Dictionary.

As a result of implementing FP-Growth, many associations, with minimum support equals 60%, have been obtained for each concept; S&M and HR. Using the testing dataset, these associations have been evaluated in order to identify which association has the highest average of recall[1] and precision[2] validation measures (i.e., F-Score[3] measure). As shown in Table (4), we had an association of six key-terms in case of S&M concept and an association of ten key-terms in case of HR concept. Therefore, new cloud services or user input queries will be mapped to a specific concept in the referenced cloud ontology, if they achieve $H_{rel} \geq 0.6$ with properties of this concept.

The evaluation results of mapping services to their suitable concepts, using pseudocode in Fig. 6 with and without tracking synonyms, are shown in Tables (5, 6). Fourteen and sixteen services have achieved $H_{rel} \geq 0.6$ for properties of S&M and HR concepts, respectively in case of tracking synonyms using our cloud dictionary (Dic). Thirteen and fifteen of these services are actually classified into S&M and HR concepts, respectively. This achieves F-score measure equals 90% in case of S&M and 86% in case of HR, while using the pseudocode without tracking synonyms (No Dic) achieves 87% for S&M and 80% for HR.

Azure S&M Cloud service		**Amcheck HR Cloud Service**	
Term	**TF**	**Term**	**TF**
on-the-fly_encryption;expression	0.0007	native;noun	0.0017
media;noun	0.0628	communicate;verb	0.0008
service;noun	0.0153	human_capital_management;expression	0.0008
content;noun	0.0007	payroll;noun	0.0119
stream;noun	0.0117	benefit;noun	0.0102
encryption;noun	0.0212	attendance;noun	0.0038
live;adjective	0.0102	affordable;adjective	0.0013
video;noun	0.0007	act;noun	0.0008
demand;noun	0.0095	compliance;noun	0.0162
static;adjective	0.0007	ongoing;adjective	0.0004
traditional;adjective	0.0007	service;noun	0.0081
packaging;noun	0.0022	aspect;noun	0.0038

TABLE 3: A snap shot from TF weights of terms of Azure S&M and Amcheck HR cloud services.

Properties of S&M Concept		**Properties of HR Concept**	
Key-Terms	**Support**	**Key-Terms**	**Support**
media;noun	100%	hr;noun	100%
live;adjective	93%	payroll;noun	83%
stream;verb	80%	benefit;noun	100%
stream;noun	80%	employee;noun	100%
cloud;noun	67%	team;noun	100%
broadcast;noun	60%	compliance;noun	100%
		professionalism;noun	100%
		employer;noun	94%
		perfect;adjective	72%
		compensation;noun	89%

TABLE 4: Properties of S&M and HR Cloud Ontology Concepts.

In order to evaluate the extracted properties for mapping user input queries to their suitable concepts, we have used a set of twenty use cases as a testing query dataset (i.e., five S&M use cases, eight HR use cases, and seven other use cases) due to the lack of queries related to cloud services. Fig. 9 illustrates an example for both HR and S&M use cases. As shown in Table (7), three and five use cases have achieved $H_{rel} \geq 0.6$ for properties of S&M and HR concepts, respectively in case of "Dic". All these use cases are actually classified into the right concepts

[1] Recall is the fraction relevant services that are retrieved.
[2] Precision is the fraction of retrieved relevant services.
[3] F-score is the average of Precision and Recall.

achieving F-score measure equals 80% in case of S&M and 82% in case of HR. While in case of "No Dic", the mechanism achieves 70% for S&M and 82% for HR.

Cloud Domains	Service Providers	Relative Entropy (H_{rel})			
		For S&M Properties		For HR Properties	
		Dic	No Dic	Dic	No Dic
Streaming and Multimedia (S&M)	Utelisys	**0.78**	**0.81**	0.0	0.0
	Switchboard	**0.70**	**0.70**	0.0	0.0
	Jvcvideocloud	**0.71**	**0.71**	0.0	0.0
	Ibm	**0.76**	**0.68**	0.30	0.0
	Cloudcovermusic	**0.83**	**0.83**	0.28	0.30
	Bluemix	**0.67**	**0.72**	0.0	0.0
	Alibabacloud	**0.96**	**0.92**	0.0	0.0
	Goeasylive	**0.62**	0.57	0.30	0.0
	Stylejukebox	**0.71**	**0.69**	0.0	0.0
	Gamefly	**0.67**	0.27	0.0	0.0
	Filmhubhq	0.46	0.39	0.0	0.0
	Audiobox	**0.67**	**0.65**	0.0	0.0
	Radiojar	**0.64**	**0.64**	0.0	0.0
	Cloudload	**0.62**	**0.60**	0.0	0.0
	Spotify	0.32	0.32	0.0	0.0
Computing	Azure	0.15	0.15	**0.60**	**0.62**
	Amazon	0.46	0.37	0.47	0.24
	IBM	0.08	0.0	0.56	0.50
	Oracle	0.0	0.0	0.20	0.0
	Alibabacloud	0.0	0.0	0.45	0.48
	Google	0.13	0.10	**0.62**	0.47
Hosting	Primcast	0.57	0.50	0.42	0.0
Storage	Alibabacloud	0.0	0.0	0.48	0.48
Scheduling	Alibabacloud	0.0	0.0	0.0	0.0
CDN	Alibabacloud	0.25	0.25	0.30	0.30
Human Resource (HR)	Wageworks	0.19	0.0	**0.63**	0.57
	Avitusgroup	**0.60**	0.0	**0.73**	**0.79**
	Abacushcm	0.0	0.0	**0.68**	**0.61**
	ADP	0.25	0.0	**0.87**	**0.82**
	coadvantage	0.0	0.0	**0.80**	**0.75**
	amihr	0.0	0.0	**0.91**	**0.86**
	Alliantbenefits	0.11	0.0	**0.67**	**0.60**
	Apprize	0.0	0.0	**0.63**	**0.60**
	intuit	0.0	0.0	0.50	0.41
	akinassoc	0.36	0.0	**0.73**	**0.71**
	Ampayroll	0.0	0.0	**0.60**	0.51
	discoverybenefits	0.53	0.0	0.56	0.52
	Allstarhrconsulting	0.0	0.0	**0.66**	**0.60**
	Atlantichradvisors	0.33	0.0	**0.64**	**0.60**
	Assuranceagency	0.27	0.0	**0.67**	**0.61**
	Acahelpers	0.0	0.0	**0.62**	0.56
	21oakhr	0.28	0.0	0.52	0.51
	execupay	0.0	0.0	**0.70**	**0.70**

TABLE 5: Relative Entropy of S&M and HR concepts' properties for cloud services in the testing dataset with and without tracking synonyms.

All results show that the extracted properties using the proposed mechanism are representative to their concepts. They achieve high precision and recall degrees to map cloud services and user queries to their relative concepts.

Our developed cloud dictionary is considered the main building block of the proposed mechanism. It does not contribute only to unify terms in the phase of extracting properties, but also map cloud services and user input queries to their suitable concepts by tracking synonyms of these properties.

For mapping services, the results show that tracking synonyms improves the recall measure by 15% on average (i.e., 16% for HR services and 14% for S&M services) and the F-Score measure by 5%. While it decreases the precision measure by 6% on average. For mapping queries, tracking synonyms improves the recall measure by 10% on average and the F-Score measure by 5%.

Cloud ontology concepts may have some common properties that could prevent the achievement of the full precision measure. So, many concepts may have properties that achieve $H_{rel} \geq 0.6$ for the same cloud service. But, the concept with the highest H_{rel} value is expected to be the intended. As shown in Table (5), the S&M and HR properties achieve H_{rel} equals 0.6 and 0.73, respectively with tracking synonyms for the HR cloud service "Avitusgroup". Considering this observation, this service will be mapped to the HR concept. Hence, the precision measure of properties of the S&M concept will become 100% (i.e., instead of 93%).

Table (6): Validation Measures	**Dic**		**No Dic**	
	S&M	**HR**	**S&M**	**HR**
Recall	87%	83%	73%	67%
Precision	93%	88%	100%	92%
F-Score	90%	86%	87%	80%

TABLE 6: Validation measures of our proposed mechanism to classify S&M and HR cloud services with and without tracking synonyms.

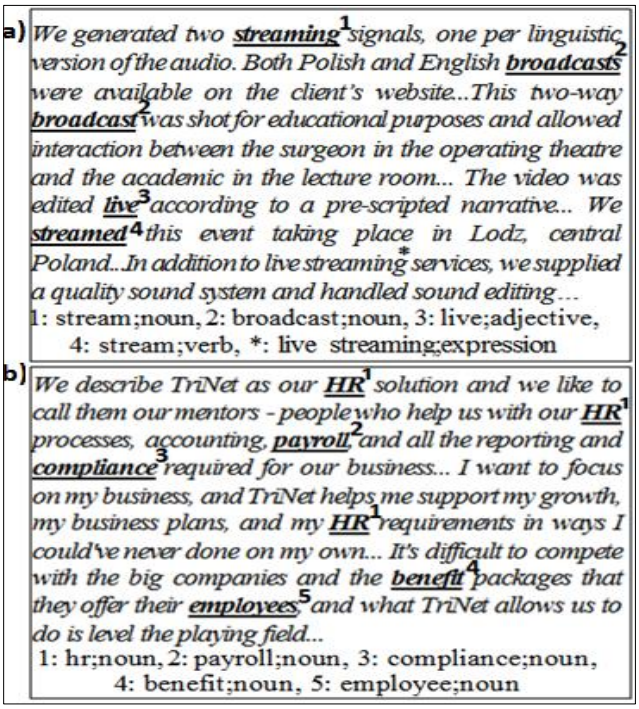

FIGURE 9: Two examples of S&M (a) and HR (b) use cases.

Validation Measures	Dic		No Dic	
	S&M	HR	S&M	HR
Recall	60%	63%	40%	63%
Precision	100%	100%	100%	100%
F-Score	80%	82%	70%	82%

TABLE 7: Validation measures of our proposed mechanism to map user input queries to S&M and HR concepts with and without tracking synonyms.

So, assigning properties for all other concepts in the referenced cloud ontology is expected to make the precision measure closer to the full mark. For example, the S&M and HR properties achieve H_{rel} equals 0.15 and 0.6, respectively with tracking synonyms for the Computing cloud service "Azure". Assigning properties for the Computing concept may cause this service map to the right concept instead of HR.

5. CONCLUSION AND FUTURE WORK

Despite the rapid growth of cloud services, the majority of enterprises do not buy such services, due to the insufficient knowledge about cloud technology. So, it is difficult for non-expert users to specify suitable concepts of the referenced cloud ontology. Users always prefer to compose their queries using vocabularies that are familiar within their working fields. But, it is difficult to really match their queries because sometimes they are ambiguous in the meaning. The work in this paper proposed a mechanism that combines between a cloud ontology, and BabelNet knowledge base to construct cloud dictionary. Terms of this dictionary are classified into their relative cloud ontology concept. That dictionary is used with FP-Growth algorithm to extract properties of the cloud ontology concepts and to track synonyms in case of mapping cloud services and user queries to the intended concept. The proposed mechanism contributes to find the right compromise between the human expression and cloud ontology concepts. According to the experimental results, the proposed mechanism has achieved high validation measures for mapping cloud services and user queries into the intended cloud ontology concept.

There are several ideas that could be addressed in the future work. These ideas include (1) the accuracy and the performance of our proposed mechanism need to be studied on all concepts of the referenced cloud ontology using a larger number of cloud services related to these concepts. 2) The effectiveness of using other pattern detection algorithms on our mechanism needs to be studied. 3) We plan to allow non-expert users to compose their queries using their native languages (i.e., French, Spanish, Arabic…). Finally, we plan to conduct a comparative evaluation of the proposed mechanism with other related researches to support the validity of this work.

6. REFERENCES

[1] Richter, F. (2019, Nov) The Statistics Portal. [Online]. http://www.statista.com/

[2] Zhang, M.; Ranjan, R.; Haller, A.; Georgakopoulos, D.; Menzel, M.; Nepal, S., "An Ontology-based System for Cloud Infrastructure Services' Discovery," in 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Pittsburgh, United States, 2012, pp. 524-530.

[3] Zhang, M.; Ranjan, R.; Menzel, M.; Nepal, S.; Strazdins, P.; Wang, L., "An Infrastructure Service Recommendation System for Cloud Applications with Real-time QoS Requirement Constraints," IEEE Systems Journal, vol. 11, no. 4, pp. 2960-2970, 2017.

[4] Alfazi, A.; Sheng, Q. Z.; Qin, Y.; Noor, T. H., "Ontology-Based Automatic Cloud Service Categorization for Enhancing Cloud Service Discovery," in IEEE 19th International Enterprise Distributed Object Computing Conference (EDOC), Adelaide, Australia, 2015, pp. 151-158.

[5] Al-Sayed, M. M.; Khattab, S.; Omara, F. A., "Prediction mechanisms for monitoring state of cloud resources using Markov chain model," Journal of Parallel and Distributed Computing, vol. 96, pp. 163-171, 2016.

[6] Kang, J.; Sim, K. M., "Ontology-enhanced agent-based cloud service discovery," International Journal of Cloud Computing, vol. 5, no. 1-2, pp. 144-171, 2016.

[7] Nagireddi, V S. K. N; Mishra, S., "An Ontology Based Cloud Service Generic Search Engine," in The 8th International Conference on Computer Science & Education (ICCSE), Colombo, Sri Lanka, 2013, pp. 335-340.

[8] Hoefer, C.N.; Karagiannis, G., "Taxonomy of cloud computing services," in IEEE GLOBECOM Workshops (GC Wkshps), Miami, Florida, USA, 2010, pp. 1345-1350.

[9] Al-Sayed, M. M.; Hassan, H. A.; Omara, F. A., "Towards Evaluation of Cloud Ontologies," Journal of Parallel and Distributed Computing, vol. 126, pp. 82–106, 2019.

[10] Tahamtan, A.; Beheshti, S. A.; Anjomshoaa, A.; Tjoa, A. M., "A Cloud Repository and Discovery Framework Based on a Unified Business and Cloud Service Ontology," in 2012 IEEE Eighth World Congress on Services (SERVICES), Hawaii, USA, 2012, pp. 203-210.

[11] Rodríguez-García, M. Á.; Valencia-García, R.; García-Sánchez, F.; Samper-Zapater, J. J., "Ontology-based annotation and retrieval of services in the cloud," Knowledge-Based Systems, vol. 56, pp. 15–25, 2014.

[12] Elbedweihy, K.; Wrigley, S.; Ciravegna, F., "Using BabelNet in Bridging the Gap Between natural language queries and linked data concepts," in The 12th International Semantic Web Conference (ISWC2013), Sydney, Australia, 2013, pp. 62-73.

[13] Fenza, G.; Senatore, S., "Friendly web services selection exploiting fuzzy formal concept analysis," Soft Computing, vol. 14, no. 8, pp. 811-819, 31 July 2009.

[14] Friedrich-Baasner, G.; Fischer, M.; Winkelmann, A., "Cloud Computing in SMEs: A Qualitative Approach to Identify and Evaluate Influential Factors," in Proceedings of the 51st Hawaii International Conference on System Sciences, Hawaii, USA, 2018, pp. 4681-4690.

[15] Afolabi, A.; Amusan, L.; Owolabi, D.; Ojelabi, R.; Joshua, O.; Tunji-Olayeni, P., "Assessment of the Linkages and Leakages in a Cloud-Based Computing Collaboration among Construction Stakeholders," in Construction Research Congress (CRC), New Orleans, USA, 2018, pp. 673-683.

[16] Androcec, D.; Vrcek, N.; Seva, J., "Cloud Computing Ontologies: A Systematic Review," in Third International Conference on Models and Ontology-based Design of Protocols, Architectures and Services, Chamonix, France, April 2012, pp. 9-14.

[17] (2019) BabelNet. [Online]. http://babelnet.org/about

[18] Han, J.; Pei, J.; Yin, Y., "Mining frequent patterns without candidate generation," in SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data, USA, 2000, pp. 1-12.

[19] Camacho-Collados, J.; Navigli, R., "BabelDomains: Large-Scale Domain Labeling of Lexical

Resources," in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Spain, 2017, pp. 223–228.

[20] Segev, A.; Sheng, Q. Z., "Bootstrapping Ontologies for Web Services," IEEE Transactions on Services Computing, vol. 5, no. 1, pp. 33-44, March 2012.

[21] Liu, L.; Yao, X.; Qin, L.; Zhang, M., "Ontology-based Service Matching in Cloud Computing," in IEEE International Conference on Fuzzy Systems, China, July 2014, pp. 2544 - 2550.

[22] Karim, R.; Ding, C.; Miri, A.; Liu, X., "End-to-End QoS Mapping and Aggregation for Selecting Cloud Services," in Collaboration Technologies and Systems (CTS), Minnesota, 2014, pp. 19-23.

[23] Gong, S.; Sim, K. M., "CB-Cloudle and Cloud Crawlers," in 5th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2014, pp. 9-12.

[24] Giese, M.; Soylu, A.; Vega-Gorgojo, G.; Waaler, A.; Haase, P.; Jiménez-Ruiz, E.; Lanti, D.; Rezk, M.; Xiao, G.; Özçep, Ö.; Rosati, R., "Optique – Zooming In on Big Data Access," Computer, vol. 48, no. 3, pp. 60 - 67, Mar. 2015.

[25] Giese, M.; Haasez, P.; Jimenez-Ruizx, E.; Lantiy, D.; Ozcepk, O.; Rezky, M. (2019) optique. [Online]. http://optique-project.eu/

[26] Soylu, A.; Giese, M.; Jimenez-Ruiz, E., "OptiqueVQS – Towards an Ontology-based Visual Query System for Big Data," in 5th International Conference on Management of Emergent Digital EcoSystems, Luxembourg, 2013, pp. 119-126.

[27] Mihalcea, R.; Tarau, P., "TextRank: Bringing Order into Texts," in Conference on Empirical Methods in Natural Language Processing, Spain, 2004, pp. 404-411.

[28] Litvak, M.; Last, M., "Cross-lingual training of summarization systems using annotated corpora in a foreign language," Information Retrieval, vol. 16, no. 5, pp. 629–656, 2013.

[29] Wang, Z.; Feng, Y.; Li, F., "The Improvements of Text Rank for Domain-Specific Key Phrase Extraction," International Journal of Simulation Systems, Science & Technology, vol. 17, no. 20, pp. 1-11, 2016.

[30] (2019) OpenNLP. [Online]. https://opennlp.apache.org/

[31] Kumar, B. S.; Rukmani, K. V., "Implementation of web usage mining using APRIORI and FP growth algorithms," International Journal of Advanced networking and Applications, vol. 1, no. 6, pp. 400-404, 2010.

[32] Taatgen, N. A., "Extending the past-tense debate: A model of the German plural," in Proceedings of the Twenty-third Annual Conference of the Cognitive Science Society, Scotland, 2001, pp. 1018-1041.