DATA SCIENCE JOURNAL

Zhao, C, et al. 2018. ROCP: A Rapid Ontology Construction Platform from Unstructured Data. *Data Science Journal,* 17: 23, pp. 1–16. DOI: https://doi.org/10.5334/dsj-2018-023

RESEARCH PAPER

ROCP: A Rapid Ontology Construction Platform from Unstructured Data

Chongchong Zhao[1], Chao Dong[1] and Xiaoming Zhang[2]

[1] School of computer and communication engineering, University of Science and Technology Beijing, CN

[2] School of information science and engineering, Hebei University of Science and Technology, CN

Corresponding author: Chao Dong (Dcb2014_ustb@163.com)

The domain ontology, which plays a significant role in knowledge-based systems, still needs the manual work of domain experts to be constructed currently. The main motivation of this paper is to provide a semi-automatic platform which can construct fairly comprehensive domain ontology from unstructured data. Firstly, a brief QA process is proposed to simplify the interaction with the domain experts. A novel algorithm MPVW, which extends from the classical algorithm TF-IDF, is proposed to extract the terminologies from domain documents. MPVW balanced more parameters and factors to evaluate the feature of terminologies. The 3-layers taxonomy and terminology hyponymy height provide sufficient guide and prompt for domain experts to construct ontology from terminologies. According to our approach we have developed ROCP, a rapid ontology construction platform which has been applied in the space debris mitigation domain. The experimental data indicates that ROCP has sufficient accuracy to extract terminologies. Meanwhile, it is effective to relieve the labor of domain experts to construct domain ontology.

Keywords: Ontology learning; semi-automatic; unstructured data; semantic web; terminology extraction

1. Introduction

The data integration brings great convenience for knowledge acquisition and association in many domains. The ontology, which is an explicit specification of a conceptualization (Gruber, 1993), has been widely used as an important data modeling tool for data integration and knowledge-based systems. Ontologies are often used to describe a specific domain. These ontologies are known as domain ontologies.

The construction of domain ontologies has been mainly relying on manual work. However, the automatic construction attracts more attention recently. An ontology can be generally divided into schema-layer and instance-layer. The schema-layer mainly depicts the domain knowledge structure through ontology classes, object properties, data type properties, axioms and rules. The instance-layer mainly contains big concrete domain data, which can usually be extracted from domain databases. In our previous work (Zhao et al., 2016), we proposed a method of the semi-automatic mapping between a domain database and an existing ontology. However, database metadata can only provide few terminologies, which are insufficient to construct a new ontology. Hence, more unstructured documents such as PDF and web text are necessary for the automatic construction of a domain ontology (Lee, 2007; Rios-Alvarado et al., 2013; Astrakhantsev and Turdacov, 2013).

Currently, more and more scholars have attempted to construct a domain ontology from unstructured data. For instance, Kara and David (2013) try to automatically construct the gene ontology; Küçük and Arslan (2014) construct the wind energy ontology; Wei et al. (2012) construct the agricultural ontology from web resources. Compared with full-automatic ontology construction, semi-automatic methods get much higher accuracy and more adoption.

However, there still exist many difficulties in the process of the semi-automatic ontology construction from unstructured data, e.g. (1) the automatic extraction of ontology relationships, (2) the hyponymy establishment of ontology classes and especially (3) the communication between domain experts and informatics experts. Since the communication often encounters trouble and misunderstanding, a lot of domain ontologies are constructed by domain-informatics experts such as the MaterialInformation Ontology (Ashino, 2010). Therefore this communication, just as the server and the client, needs a standard "protocol" to ensure efficiency and reliability. This "protocol" defines the details of request and response. In this paper we proposed a novel communication mechanism. The informatics experts utilize a QA (Question and Answer) mechanism rather than face-to-face manner to communicate with domain experts.

In order to implement our idea, we develop ROCP (Rapid Ontology Construction Platform) for domain experts. Currently, ROCP has been applied in the space debris mitigation domain. The main contribution of our approach is shown as follows:

- A convenient and concise communication mechanism with domain experts. A QA mechanism instead of face-to-face meeting reduces a lot of unnecessary troubles. The majority of the manual work for domain experts can be accomplished by simple selections in ROCP.
- The Multiple Parameters Variable Weight (MPVW) algorithm is proposed for terminology extraction. This algorithm extends from classical TF-IDF algorithm and adds some new strategies to balance the parameters.
- The Terminology Hyponymy Height (THH) algorithm and 3-layers nodes taxonomy are proposed for ontology construction. These methods can provide clear guidelines and relieve the labor of domain experts.

The remainder of this paper is organized as follows. Section 2 reviews the state of the art of ontology-learning. Section 3 briefly introduces the QA process and illustrates the overview and methodology of our approach by a flow chart. Section 4 elaborates the terminology extraction from unstructured domain documents. Section 5 depicts the semi-automatic ontology construction from terminologies. Section 6 shows a case study and the analysis of the experimental data. Section 7 summarizes our approach and puts forward issues for our future work.

2. Related Work

The ontology construction from unstructured data can be regarded as a form of ontology-learning (Maedche and Staab, 2001). Generalized ontology-learning contains not only the construction of an ontology through learning, but also the enrichment and expansion of an ontology through learning (Astrakhantsev and Turdakov, 2013), which is called ontology evolution (Sellami and Camps, 2012). The basic framework of a new ontology can be constructed from unstructured data, and the refinement can be achieved by the ontology evolution in the future.

A number of systems are proposed for the extraction from unstructured data, e.g. Text-to-Onto (Maedche and Staab, 2000), TextOntoEx (Dahab, 2008), OntoLearn (Navigli et al., 2003), ASIUM (Faure et al., 1998), PKS (Manganello, 2013) and YAMO (Dutta, 2015). Currently, the mainstream methods of the ontology construction from unstructured data can be mainly divided into three categories: (1) statistics-based methods, (2) linguistics-based methods and (3) dictionary-based methods (Zhang and Wu, 2012).

Statistics-based methods are the most popular methods. Especially in the era of big data, the booming of text-clustering makes statistics-based methods more advanced. The general idea of statistics-based methods is the calculation of a "total score" which can evaluate the candidate words. For example, Marciniak and Mykowiecka (2014) propose the "C-value" to evaluate the candidate words. Statistics-based methods have greater advantages in terminologies extraction (Macken et al., 2013; Choi and Myaeng, 2012; Bernth et al., 2003; Chung, 2002). Therefore, they are more widely used in semi-automatic systems (Wei et al., 2012; Küçük and Arslan, 2014).

Linguistics-based methods utilize more NLP (Natural Language Processing) algorithms to discover the hyponymy and further relationships (Liu et al,. 2008; Niu et al,. 2015). Therefore, this kind of methods has greater advantages to search for the relationships of ontology concepts, especially non-taxonomy relationships (Sánchez and Moreno, 2008).

Dictionary-based methods can make use of the semantic annotation of a custom or external knowledge base. For instance, Erdmann et al. (2009) utilize Wikipedia to extract terminologies; Küçük and Arslan (2014) utilize Wikipedia to construct a wind-energy ontology. Dictionary-based methods often work with another kind of methods (Weng et al., 2006). Moreover, multi-strategy methods are also widely used (Shamsfard and Barforoush, 2003).

In conclusion, ROCP chooses statistics-based methods to generate a domain-correlativity ranking of the result terminologies, which can help domain experts with the manual work. Moreover, ROCP proposes a novel interaction mechanism with the domain experts, which is an innovation currently.

3. Overview and Methodology

The main motivation of ROCP is to rapidly construct fairly comprehensive domain ontology rather than to spend a lot of time to construct encyclopedic ontology. Therefore, ROCP should enhance the degree of automation under the premise of ensuring the basic accuracy. The convenient QA process between ROCP and domain experts, which is illustrated in **Figure 1**, is an important way to improve the automation.

Figure 1 shows the QA process between ROCP and domain experts. The whole process can be divided into two major phases (i.e., terminology extraction from domain documents and ontology construction from terminologies). Firstly, if domain experts start domain ontology construction, ROCP will request them to upload domain documents. After the first response of domain experts, ROCP performs the pre-processing such as word segmentation and document validation. In order to enable the domain experts to regulate the final result, ROCP requests them to configure the parameters (e.g. the weight of factors in terminology extraction). After the second response of domain experts, ROCP performs the terminology extraction and request domain experts to make a simple classification of the extracted terminologies. After the third response of domain experts, ROCP can generate ontology nodes and request users to establish the relationships of the nodes. Finally, ROCP returns OWL files as the result ontology (Pascal et al., 2007).

Figure 2 comprehensively illustrates the process of ontology construction from unstructured data. The left part of **Figure 2** depicts the terminology extraction from unstructured domain documents. The domain experts provide domain documents and ROCP integrates large amount of domain-independent documents as corpus. Invalid domain documents can be removed through the cosine-similarity algorithm. All the words in domain documents will be segmented and ROCP will calculate the domain-correlativity of every appeared word. The Multiple Parameters Variable Weight (MPVW) algorithm is designed to implement the calculation. This algorithm extends from the classical algorithm TF-IDF and can freely balance the weight of all parameters. High domain-correlativity words will be extracted as terminologies, which are sorted according the Terminology Hyponymy Height (THH) for the next step.

The right part of **Figure 2** describes the ontology construction from terminologies. The domain experts firstly put the extracted terminologies into three layers (i.e., class layer, property layer and individual layer) and discard incorrect terminologies. Afterwards the domain experts can construct the hierarchy of the class layer under the guide of Terminology Hyponymy Height (THH). Subsequently, the ontology properties and instances will be linked to corresponding ontology classes.

The word segmentation of domain documents can be achieved by means of Apache Lucene, and the ontology construction can be achieved with the help of Apache Jena. An ontology model will be created by Apache Jena to display the temporary ontology being edited by the domain experts. Finally, an OWL file will be generated according to the result ontology model.

Throughout the whole process of ROCP running, a new ontology grows from scratch to rich. This is the process of ontology learning from unstructured data. However, it is not sufficient to learn axioms and rules from

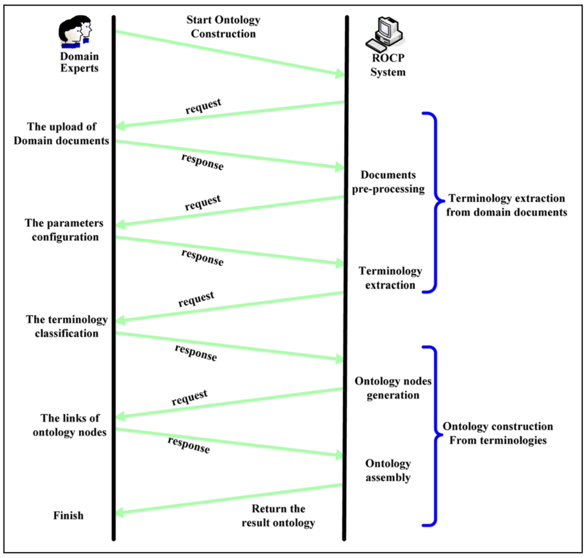

Domain Experts
Start Ontology Construction
ROCP System
request
The upload of Domain documents
response
Documents pre-processing
request
The parameters configuration
response
Terminology extraction
Terminology extraction from domain documents
request
The terminology classification
response
Ontology nodes generation
request
The links of ontology nodes
response
Ontology assembly
Ontology construction From terminologies
Finish
Return the result ontology

Figure 1: The QA process between ROCP and domain experts.

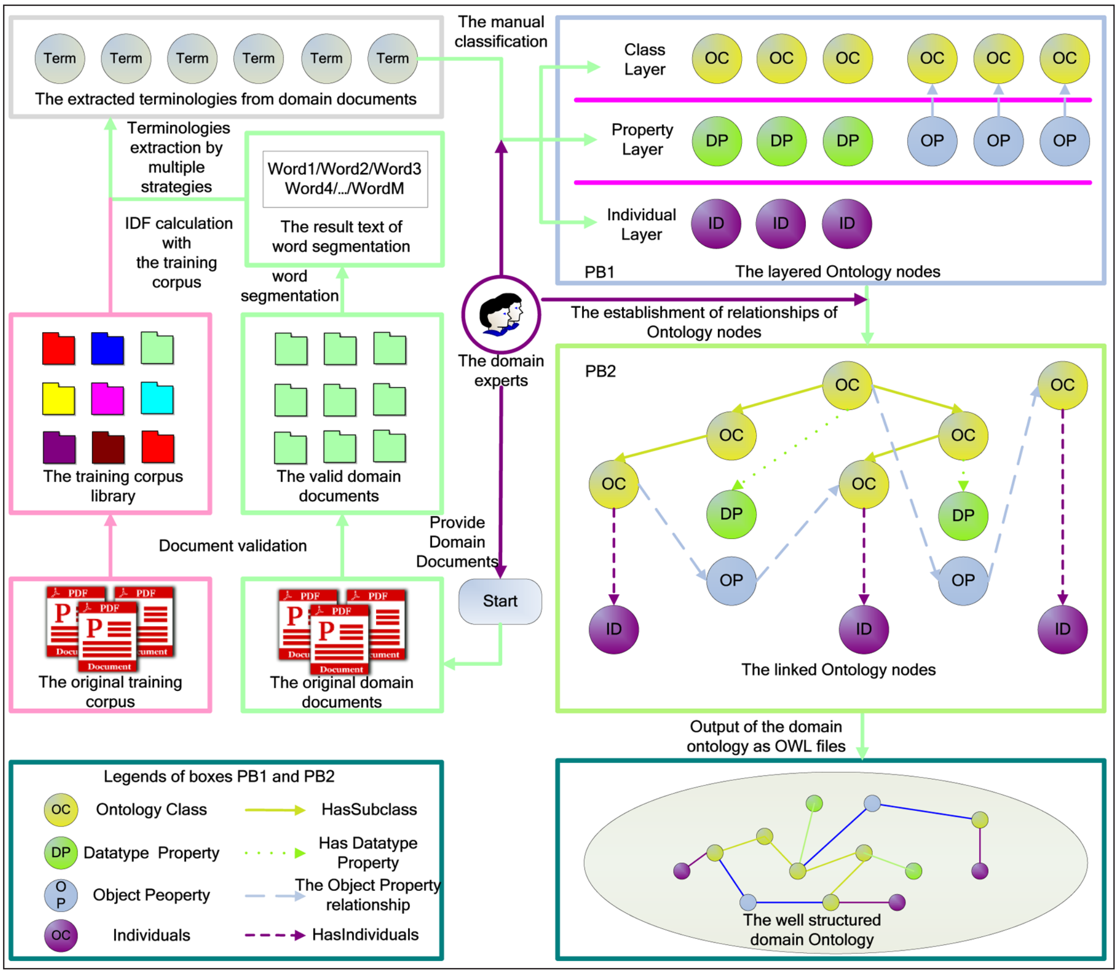

Term
The extracted terminologies from domain documents
The manual classification
Class Layer
OC
Property Layer
DP
OP
Individual Layer
ID
PB1
The layered Ontology nodes
Terminologies extraction by multiple strategies
Word1/Word2/Word3 Word4/.../WordM
The result text of word segmentation
IDF calculation with the training corpus
word segmentation
The domain experts
The establishment of relationships of Ontology nodes
PB2
The training corpus library
The valid domain documents
Document validation
Provide Domain Documents
Start
The original training corpus
The original domain documents
The linked Ontology nodes
Output of the domain ontology as OWL files
Legends of boxes PB1 and PB2
Ontology Class
HasSubclass
Datatype Property
Has Datatype Property
Object Peoperty
The Object Property relationship
Individuals
HasIndividuals
The well structured domain Ontology

Figure 2: The overview of our approach.

merely unstructured domain documents. The ontology is not immutable after construction. On the contrary, it can improve itself through further learning.

4. The extraction of terminologies from domain documents

4.1. Text Preprocessing

ROCP firstly converts domain documents into statistics available words by words segmentation, stemming and stopping. Stemming can make different forms of a word be treated as a same word. Stopping can make function words (e.g., at, the, of) in a stop list be excluded from statistics.

Table 1 shows an algorithm of domain document words segmentation. ROCP uses a two dimensional list W to return the segmented words (Line 1). All domain documents will be segmented into words (Line 2–3), and all words will be further processed by stemming and stopping (Line 4–9). Finally, the list W is statistics available for the terminology extraction.

In order to compare the similarity between documents, we construct the Vector Space Model (VSM) from the statistics available words. The dimension of the vector is the number of the words which appeared at least once. Each document corresponds to a vector. Each element of the vector represents the occurrences of the words in a document.

Table 2 shows an algorithm of the VSM construction. Actually, ROCP selects N highest-frequency words to obtain the dimension of the vectors (Line 3–6). Afterwards, the vectors will be generated according to the occurrences of the words (Line 7–11).

4.2. Document Validation

In order to distinguish between terminologies and non-terminologies, ROCP integrates large amount of domain-independent documents as corpus. However, a few of the documents in the corpus may happen to be domain-related documents. Meanwhile, a small part of the domain documents may be invalid. Therefore, the document validation is necessary.

Table 1: The algorithm of the Domain document word segmentation.

Algorithm 1.1 Domain document word segmentation

Input: Domain documents List D
Output: The segmented words W;

```
List W;
for each i in D
  List $W_i$ = D.WordSegmentationByLucene();
    for each j in $W_i$
    $W_{ij}$.stemming();
      if $W_{ij}$ in stopwordlist
        Wi.remove($W_{ij}$);
      end if
    end for
  end for
  return W;
```

Table 2: The algorithm of the construction of VSM.

Algorithm 1.2 The construction of VSM

Input: The segmented words W, words number N.
OutPut: The vector space model of each document VSM;

```
List HFW;
List VSM;
  for each i in W
      $HFW_i$=$W_i$.findHighFrequencyWords(WN);
  end for
List WA=HFW.allHighFrequencyWords();
  for each j in W
      for each k in WA
          VSMk=$WA_k$.appearedTimesIn($W_j$);
      end for
end for
```

Figure 3 shows the process of document validation. The cosine-similarity algorithm is used to search for invalid domain documents. Firstly, all domain documents will be converted into vectors as shown in **Figure 4**. The included angle between two vectors can indicate their similarity. Most of the domain documents provided by domain experts should be similar (v1–v5 in **Figure 4**). Only a minority of the documents may be quite different from others (v6 in **Figure 4**). Therefore, ROCP can locate the invalid documents by calculating the average cosine value (AVC) of each vector and all the vectors.

The detailed calculation process of AVC can be expressed by formula 1–4. The domain documents (DD) is converted to vectors A, B, etc. The value N is the amount of the domain documents DD. The value n is the dimension of the vectors. The similarity of vectors A and B is expressed by Sim(A, B) in formula 3. The average cosine value of each vector can be calculated by formula 4.

$$DD = \{A, B, \ldots\} \tag{1}$$

$$A = (a_1, a_2, \ldots a_n); B = (b_1, b_2, \ldots b_n) \tag{2}$$

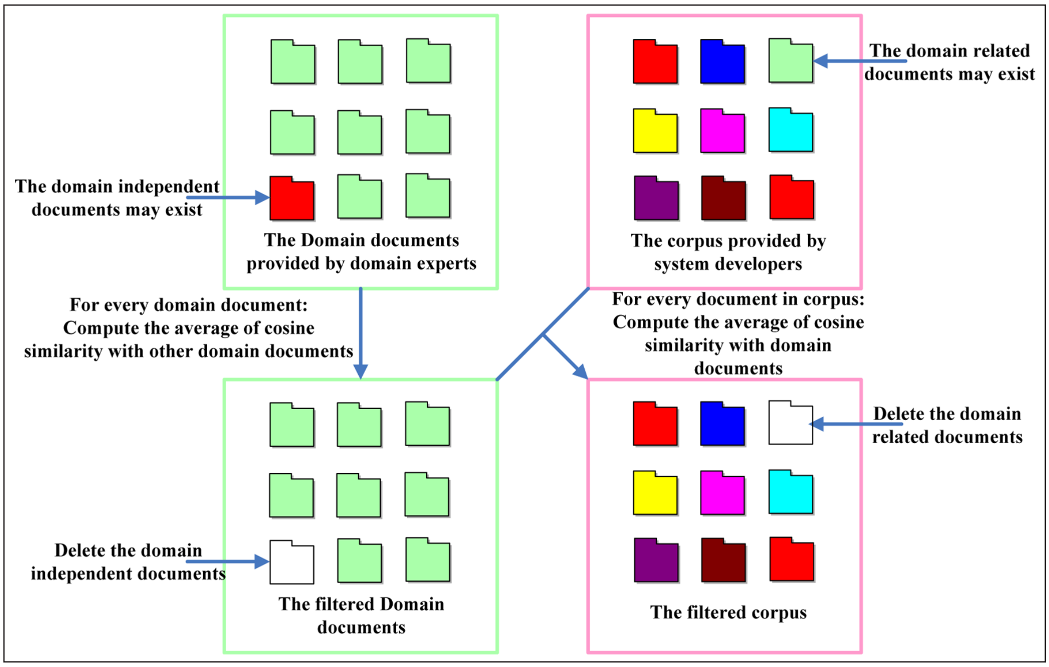

The domain independent documents may exist
The Domain documents provided by domain experts
The domain related documents may exist
The corpus provided by system developers
For every domain document: Compute the average of cosine similarity with other domain documents
For every document in corpus: Compute the average of cosine similarity with domain documents
Delete the domain related documents
Delete the domain independent documents
The filtered Domain documents
The filtered corpus

Figure 3: Documents Validation.

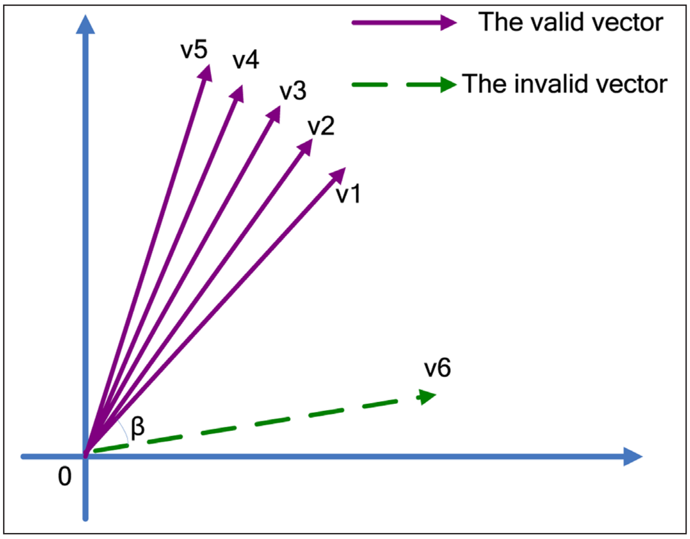


Figure 4: The cosine similarity algorithm to locate invalid documents.

$$Sim(A,B) = \cos\beta = \frac{\sum_{i=1}^{n}(a_i \times b_i)}{\sqrt{\sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2}} \tag{3}$$

$$AVC(A) = \frac{1}{N}[Sim(A,A) + Sim(A,B) + \ldots + Sim(A,N)] \tag{4}$$

Table 3 shows an algorithm to remove invalid documents. ROCP will calculate each of the average cosine-similarity with all other vectors (Line 2–7). All the average cosine-similarity of the vectors will take the average again as the total average cosine-similarity (Line 8–10). Domain experts can set a threshold CT to remove the invalid documents. If the absolute value of the difference between the average cosine-similarity of a vector and the total average cosine-similarity exceeds the threshold CT, the document which corresponds to this vector will be removed as an invalid document (Line 11–15).

Table 3: The algorithm to remove invalid documents.

Algorithm 1.3 Remove invalid documents

Input: The Vector Space Model VSM, the cosSimilarity threshold CT, The domain documents D;
Output: The valid domain documents D;

```
sumcos$_1$=0; sumcos$_2$=0;
for each i in VSM
    for each j in VSM
        cosSim$_{ij}$=VSMi.computeCosSimilarityWith(VSM$_j$);
        sumcos$_1$+=cosSim$_{ij}$;
    end for
      avgCosSimi=sumcos$_1$/j
      sumcos$_2$+= avgCosSim$_i$
end for
totalAvgCosSim=sumcos$_2$/i
for each i in avgCosSim
    if Math.abs(totalAvgCosSim-avgCosSim$_i$)>CT
    D.removeDocumentByItsVSMIndex(i)
    end if;
end for;
return D;
```

4.3. MPVW Algorithm for terminology extraction

In order to achieve terminology extraction, ROCP uses a new algorithm based on the classical algorithm TF-IDF. For term ***i*** in document ***j***, TF (Term Frequency) can be calculated by formula 5, IDF (Inverse document frequency) can be calculated by formula 6. The numerator in formula 5 stands for the number of the occurrences of term ***i*** in document ***j***. The denominator in formula 5 can be regarded as the total number of words in document ***j***. The numerator in formula 6 stands for the total number of documents in the corpus. The denominator in formula 6 stands for the number of the documents in the corpus which contains the term ***i***. In addition, the denominator should add 1 in case of the zero denominators.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{5}$$

$$IDF_i = \log \frac{|D|}{1+|\{j : t_i \in d_j\}|} \tag{6}$$

$$TFIDF_{ij} = TF_{ij} \times IDF_i \tag{7}$$

The TF-IDF value can be regarded as a score to evaluate whether a word is a key word. The values of TF and IDF are equally important for the result. Therefore, the TF-IDF value is calculated by simple multiplication of TF and IDF as shown in formula 7.

However, terminologies are different from key words. Key words generally appear many times in the documents, while the terminologies may appear only one or two times in the domain documents. The characteristic of terminologies is the quite low occurrences in domain-independent documents. Therefore, IDF is more important than TF for terminologies extraction. Weighting should be used in order to compute a score which can evaluate whether a word is a terminology.

In classical TF-IDF algorithm, the IDF value is obtained by logarithm. If the logarithm is not taken to the IDF, the range of IDF will become quite large. If so, the influence of IDF on the TF-IDF value will be much greater than that of TF. Therefore, taking logarithm of IDF can balance the influence of TF and IDF on the TF-IDF value. In fact, this is also a method of weighting.

Obviously, the weight of a variable in a multiplication product corresponds to its range. For TF and IDF, their range can be expressed in formula 8. Meanwhile, the TF-IDF value can be expressed in formula 9. A and B are defined in formula 10, the range of A and B have a common lower limit 1 in formula 11. At this time, the weight of A and B depends on their upper limit. Similarly, for any variable ***p***, we define the nature weight NW in formula 12.

$$TF \in [TF_{\min}, TF_{\max}]; IDF \in [IDF_{\min}, IDF_{\max}] \tag{8}$$

$$TFIDF = TF \times IDF = TF_{\min} \times IDF_{\min} \times \frac{TF}{TF_{\min}} \times \frac{IDF}{IDF_{\min}} \tag{9}$$

$$\frac{TF}{TF_{\min}} = A; \frac{IDF}{IDF_{\min}} = B \tag{10}$$

$$A \in [1, \frac{TF_{\max}}{TF_{\min}}]; B \in [1, \frac{IDF_{\max}}{IDF_{\min}}] \tag{11}$$

$$p \in [a,b]; NW_p = \frac{b}{a} \tag{12}$$

Therefore, the weight of a variable can be changed by zooming its range. If the weight of a variable need to be scaled by coefficient k, the lower limit of its range can be kept unchanged, the upper limit of its range should be scaled by coefficient k.

For variable ***P***, whose range is from **a** to **b**, a new variable ***$N_k(P_i)$*** can express the scaled variable ***P*** by coefficient **k**. The upper limit of ***$N_k(P_i)$*** is enlarged to **k*b**. Meanwhile, the general terms ***$N_k(P_i)$*** maintain the origin ratio of distance in the number axis. Thus, the value of ***$N_k(P_i)$*** can be calculated by substituting ***P_i*** into formula 15.

$$P_i \in [a,b];\ length(P_i) = b - a \tag{13}$$

$$N_k(P_i) \in [a, kb],\ length(N_k(P_i)) = kb - a, k > 1 \tag{14}$$

$$N_k(P_i) = a + (kb - a)\frac{P_i - a}{b - a} \tag{15}$$

ROCP can arbitrarily set weights for parameters by formula 15. Moreover, some new parameters besides TF and IDF can join in. The terminologies are generally longer than other normal words. Thus, a new parameter ***WL*** is defined in formula 16. For term ***i*** in document ***j***, the numerator in formula 16 stands for the word length of term ***i***. The denominator in formula 16 stands for the length of the longest word in document ***j***.

Corresponding to IDF, a new parameter DDF (Domain-Document Frequency) is defined in formula 17. The numerator in formula 17 stands for the total number of the domain documents. The denominator in formula 17 stands for the number of the documents in the domain document set which contains the term ***i***.

$$WL_{i,j} = \frac{length(i)}{\max length(j)} \tag{16}$$

$$DDF_i = \log \frac{|D'|}{1 + |\{j : t_i \in d_j\}|} \tag{17}$$

If the domain experts input w_1, w_2, w_3 and w_4 for the weights of TF, IDF, DDF and WL, the ratio of four new parameters a, b, c and d can be calculated by formula 18. Afterwards, the ***TermScore*** which can evaluate whether a word is a terminology can be drawn by formula 19. ***$N_k(P_i)$*** can be calculated by formula 15. Terminology extraction can be easily achieved according to ***TermScore***. In this paper, the algorithm to calculate ***TermScore*** is called MPVW (Multiple Parameters Variable Weight) algorithm.

$$a\frac{TF_{\max}}{TF_{\min}} : b\frac{IDF_{\max}}{IDF_{\min}} : c\frac{DDF_{\max}}{DDF_{\min}} : \mathrm{d}\frac{WL_{\max}}{WL_{\min}} = w_1 : w_2 : w_3 : w_4 \tag{18}$$

$$TermScore_{ij} = N_a(TF_{ij}) \times N_b(IDF_i) \times N_c(DDF_i) \times N_d(WL_{ij}) \tag{19}$$

5. The ontology construction from terminologies

5.1. 3-Layers taxonomy

Terminology extraction is mainly achieved by automatic methods. However, the ontology construction needs more manual work of domain experts. Simple selections are obviously the most convenient for the domain documents. Therefore, ROCP allows domain experts to complete the interaction by selecting. Meanwhile, a number of recommendation data are provided to help users complete the selections.

After the terminology extraction the domain experts can set a threshold to search for a number of words which get highest ***TermScore*** as candidate terminologies. A minority of the candidate terminologies will be discarded as invalid terminologies by the domain experts. Most candidate terminologies will be converted into ontology nodes, which have different types such as ontology class, object property, datatype property and individuals. Therefore, it is necessary to make a preliminary taxonomy of terminologies.

Figure 5 briefly illustrates the process of the taxonomy. The valid terminologies are divided into class layer, property layer and individual layer. Domain experts can achieve the taxonomy via an interface in **Figure 6**. In particular, the object properties are not directly selected. A part of ontology classes are the range of object properties (e.g. mitigation and orbit in **Figure 6**). Corresponding object properties will be created based on these classes in **Figure 7**. Default names of the new object properties (e.g. HasMitigation and HasOrbit) are provided for convenience. Domain experts can modify them if necessary.

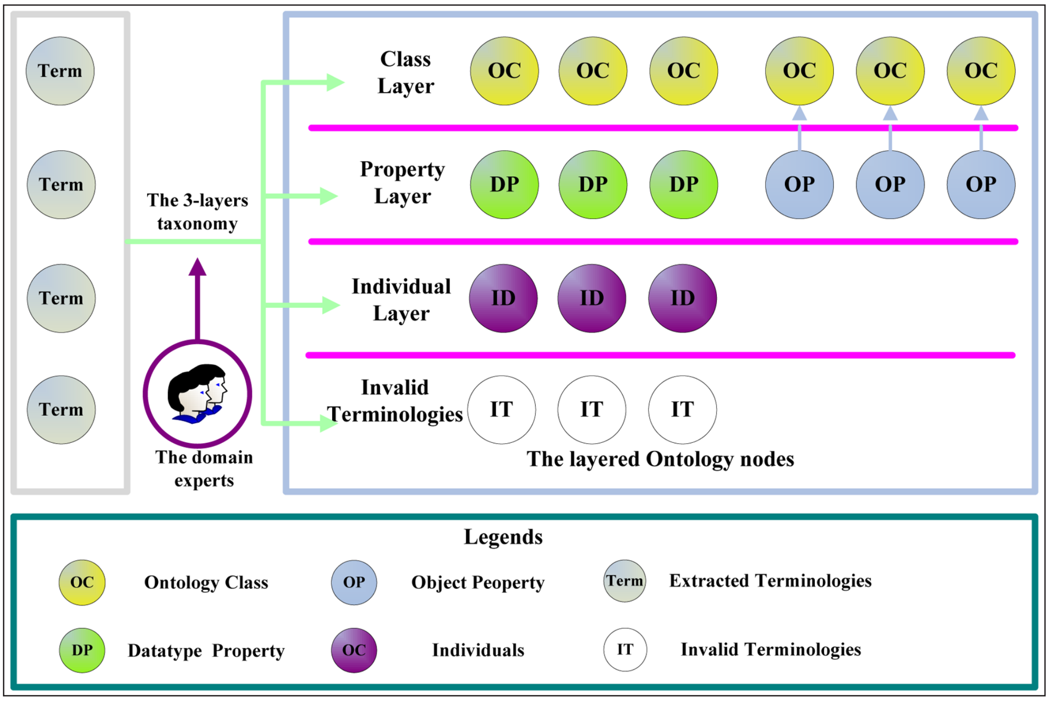
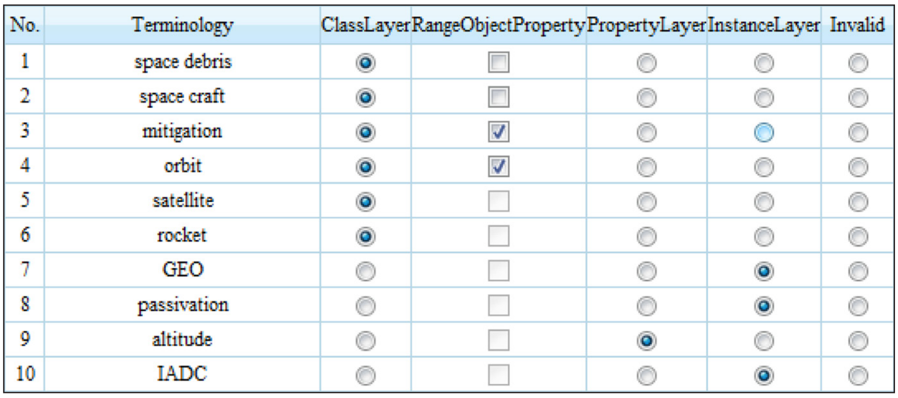


Figure 5: The 3-layers taxonomy.



No.	Terminology	ClassLayer	RangeObjectProperty	PropertyLayer	InstanceLayer	Invalid
1	space debris	●	□	○	○	○
2	space craft	●	□	○	○	○
3	mitigation	●	☑	○	○	○
4	orbit	●	☑	○	○	○
5	satellite	●	□	○	○	○
6	rocket	●	□	○	○	○
7	GEO	○	□	○	●	○
8	passivation	○	□	○	●	○
9	altitude	○	□	●	○	○
10	IADC	○	□	○	●	○

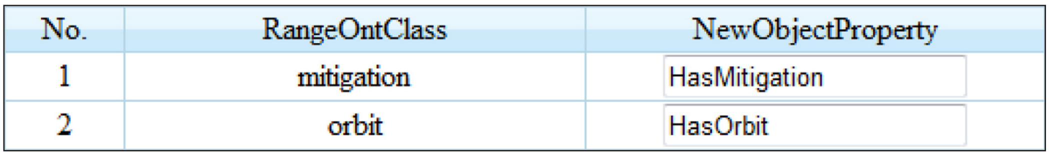
Figure 6: A part of the selection for domain experts to achieve 3-layers taxonomy.

After the 3-layers taxonomy, ROCP will create a temporary ontology model by Apache Jena to save the results of the taxonomy. In the ontology model, ontology nodes will be created according to the terminologies.

5.2. Ontology assembly

After the 3-layers taxonomy, ontology nodes must be assembled by a series of relationships to form a complete ontology model. As shown in **Figure 8**, these relationships consist of the hyponymy of ontology classes, the domain of datatype properties and object properties, and the types of individuals. In particular, the hyponymy of ontology nodes is the most important and tricky.

The algorithm for ontology class hyponymy construction is shown in **Table 4**. The input NodesPool stands for a list which contains all the ontology class nodes. The output OntTree is a 2 dimensional list which can save the nodes hierarchically. Firstly the domain experts select the root nodes from the NodesPool (Line 1). Subsequently, the root nodes will be added in the OntTree as the first layer (Line 2). Meanwhile,



No.	RangeOntClass	NewObjectProperty
1	mitigation	HasMitigation
2	orbit	HasOrbit

Figure 7: Object property creation.

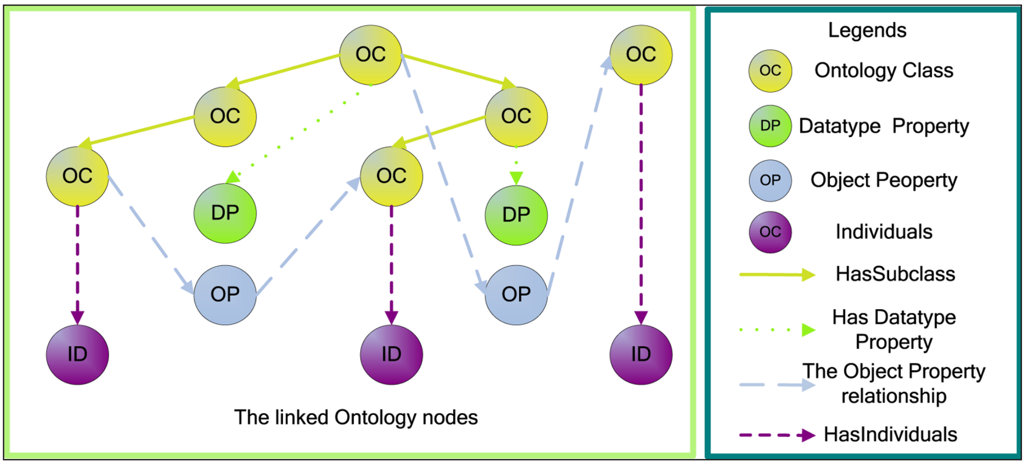


Figure 8: Ontology assembly.

Table 4: The algorithm for ontology classes hyponymy construction.

Algorithm 2 The construction of ontology classes hyponymy

Input: list NodesPool;
OutPut: list OntTree;

```
1.  List rootNodes=SelectRootNodesByExperts(NodesPool);
2.  OntTree$_0$=rootNodes;
3.  NodesPool.remove(rootNodes);
4.  int n=1;
5.  while(NodesPool.hasElement())
6.      tempnodes=SelectNodesByExperts (NodesPool);
7.      OntTree$_n$.addsubnodes(tempnodes);
8.      OntTree$_{n+1}$.add(tempnodes);
9.      NodesPool.remove(tempnodes);
10.     n++;
11. end while
```

the NodesPool will remove the selected nodes (Line 3). As long as there are nodes exist in the NodesPool, domain experts can select sub nodes of the current layer to construct next layer. Meanwhile, the NodesPool will remove the selected nodes (Line 4–11).

However, selecting a layer of nodes from all nodes will cost a lot of time. Therefore, a new parameter terminology hyponymy height (THH) is proposed to sort the nodes. Generally, the upper layer of terminologies has a lot of occurrences in the domain document, and it also has some occurrences in domain-independent documents. On the contrary, the lower layer of terminologies has less occurrences in the domain document, and it has nearly no occurrences in domain-independent documents. Therefore, according to the definition of TF and IDF, the hyponymy height ***THH*** is defined in formula 20.

$$THH = \frac{TF}{IDF} \tag{20}$$

Thus, the upper layer of terminologies will be sorted in the front of all nodes. Domain experts can conveniently build each layer. ROCP will save all the selection of domain experts in the ontology model by Apache Jena. After the ontology assembly, a complete OWL file can be generated according to the ontology model.

6. Experimental data analysis

6.1. A case study in space debris mitigation domain

Currently, ROCP has been applied in space debris mitigation domain, which is secret-related. Domain experts can firstly extract terminologies from domain documents by ROCP. In **Figure 9**, the tag cloud of the extracted terminologies is generated to give a brief result for the domain experts. Afterwards, the domain experts can discard unnecessary terminologies and construct the ontology under the guide of ROCP. A part of the terminologies are renamed for more accurate definition (e.g. GEO is turned to Geostationary Orbit; SSO is turned to Sun Synchronous Orbit, etc.). The main structure of the ontology in space debris mitigation domain is shown in **Figure 10**.

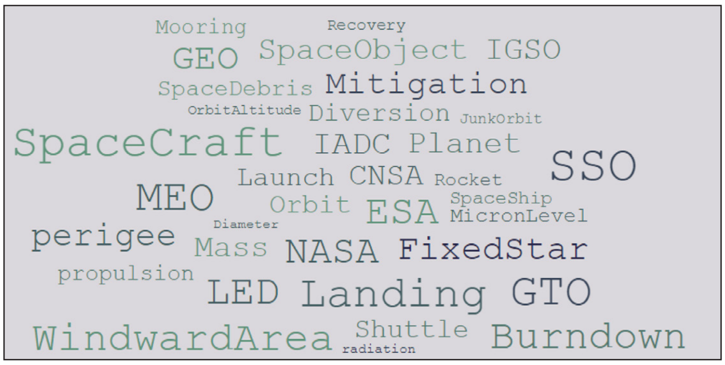


Figure 9: The tag cloud in space debris mitigation domain.

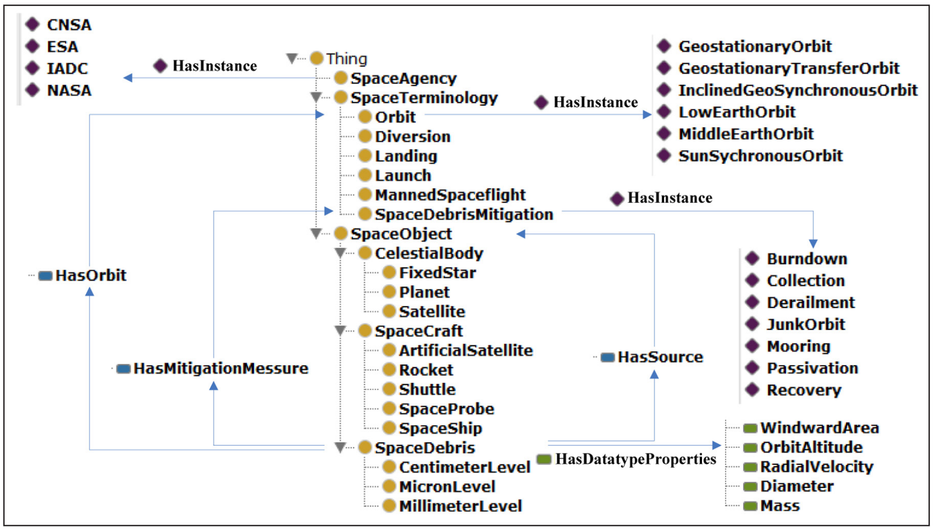


Figure 10: The main part of the ontology in space debris mitigation domain.

6.2. The accuracy comparison of algorithm MPVW and TF-IDF

An experiment is designed to evaluate the feasibility of the MPVW algorithm. Different domains may have different characteristics. Therefore, we collect two sets of domain documents. One set is from a small and specific domain, the other is from a big and wide domain. MPVW and classical TF-IDF algorithms compete for higher accuracy.

The Corpus is extracted from China Daily English Edition. Each edition of this journal is regarded as a document. The domain documents set 1(DS1) is extracted from a small domain named space debris mitigation. The domain documents set 2(DS2) is extracted from a big domain named astronautics fundamentals. The detailed information of the corpus and experimental data sets are shown in **Table 5**.

ROCP firstly performs the stemming and stopping to make a statistics of non-repeat words. These words will be regarded as total valid words (TW). Domain experts manually search for terminologies from TW, the result is regarded as total terminologies (TT). Afterwards, TW will be sorted by the MPVW score or TF-IDF score. The words which have highest score will be extracted as terminologies. The number of the extraction (NE) is 120% of TT. The number of correct words in NE is expressed as NC. The related statistics is shown in **Table 6**.

$$recall = \frac{NC}{TT} \tag{21}$$

$$precision = \frac{NC}{NE} \tag{22}$$

$$f1 - measure = \frac{2^{*} recall * precision}{recall + precision} = \frac{2NC}{NE + TT} \tag{23}$$

In this paper, three parameters recall, precision and F1-measure are used to evaluate the feasibility of the algorithms. The recall is defined in formula 21. The precision is defined in formula 22. The F1-Messure is defined in formula 23. The related experimental data is shown in **Table 7**. The corresponding histogram is shown in **Figure 11**.

Table 5: The detailed information of the corpus and experimental data.

Documents	**The Corpus**	**Domain documents set 1**	**Domain documents set 2**
Source	**China Daily**	**Space debris mitigation**	**Astronautics fundamentals**
Number of documents	1000	20	50
Total number of words	1777763	54619	145628
Average number of words	1778	2731	2513

Table 6: The statistics of the extracted terminologies.

	Total Valid words(TW)	**Total Terminologies(TT)**	**Number of Extraction(NE)**	**Number of Correct words(NC)**
DS1-MPVW	2617	129	155	123
DS1-TF-IDF	2617	129	155	81
DS2-MPVW	4126	288	346	254
DS2-TF-IDF	4126	288	346	209

Table 7: The result of the recall, precision and F1-Measure.

	Recall	**Precision**	**F1 Measure**
DS1-MPVW	95.3%	79.4%	86.6%
DS1-TF-IDF	62.8%	52.3%	57.1%
DS2-MPVW	88.1%	73.4%	80.1%
DS2-TF-IDF	72.6%	60.4%	65.9%

The Experimental data shows that MPVW algorithm has obvious advantage in recall, precision and F1-measure. However, this advantage will decrease when the documents comes from a big domain. The reason is that the terminologies in a big domain have more opportunity to exist in the corpus. On the contrary, the terminologies in a small domain may be more specialized and have little opportunity to exist in the corpus. Therefore, MPVW is more suitable for the terminologies extraction in a small domain.

6.3. The time test of the semi-automatic ontology construction

The ontology construction from terminologies needs more manual work of domain experts. Therefore, we make a statistics of each period of the manual operation. Four data sets DS3, DS4, DS5 and DS6 which have different number of terminologies are used in this test. At last the pure manual ontology construction time by Protégé is shown as a comparison. The detailed experimental data is shown in **Table 8**. The corresponding histogram is shown in **Figure 12**.

The manual ontology construction costs a lot of time. Especially, domain experts will be more confused when the numbers of terminologies are very large. The experimental data indicates that ROCP can save 42% time when the number of terminology is 85 but 56% time when the number of terminology is 254. Moreover, ROCP can save more time when the result domain ontology is larger. The reason is that the nodes classification and sorting by ROCP is more important to deal with big data.

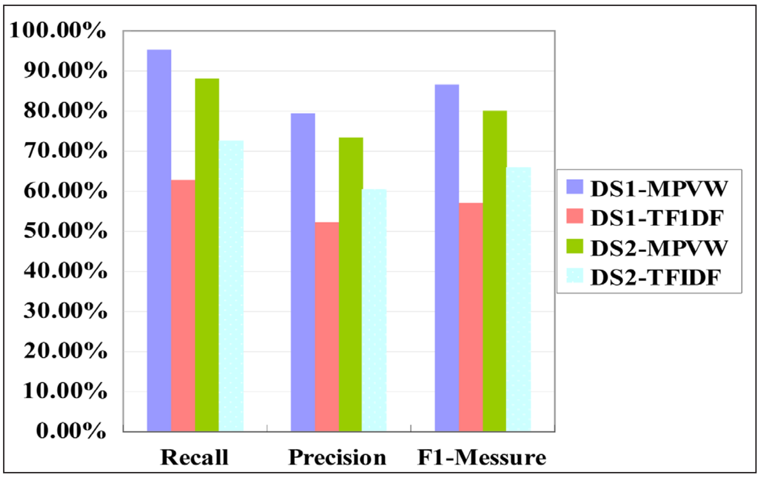


Figure 11: The accuracy comparison of algorithm MPVW and TF-IDF.

Table 8: The time cost of each period of the manual operation.

Data sets	DS3	DS4	DS5	DS6
Number of Terminologies	85	123	171	254
3-layers taxonomy	382s	579s	856s	1366s
Hyponymy construction	415s	695s	1056s	1690s
Properties and instances link	236s	346s	491s	747s
ROCP Total time	1033s 12.15 s/word	1620s 13.17 s/word	2403s 14.05 s/word	3803s 14.97 s/word
Protégé Total Time	1787s 21.02 s/word	2867s 23.31 s/word	4602s 26.91 s/word	8708s 30.28 s/word

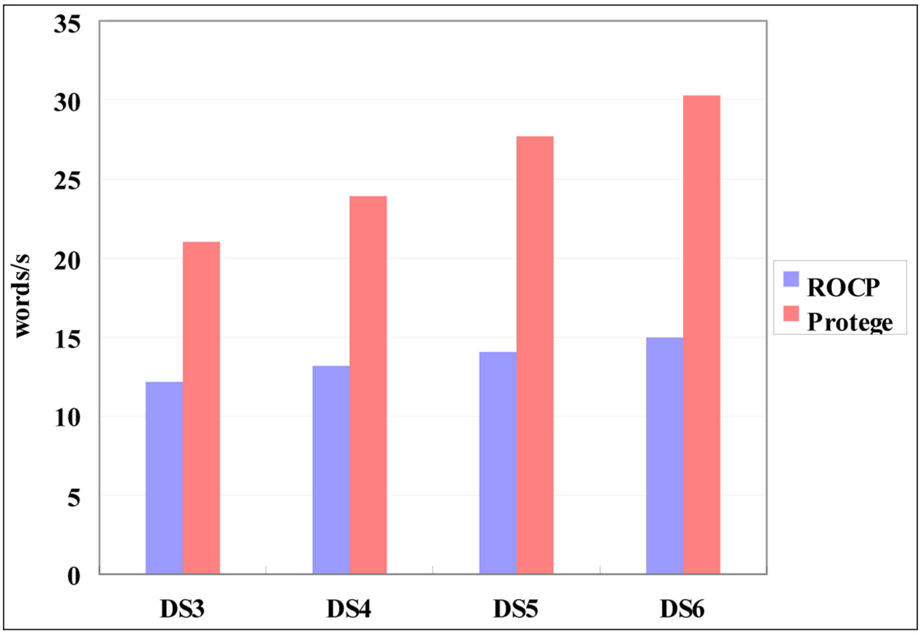


Figure 12: The time test of ontology construction by ROCP and manual work by Protégé.

7. Conclusions

In this paper, we have proposed an approach to establish a rapid domain ontology construction platform ROCP. ROCP uses a QA mechanism to enable domain experts to achieve the ontology construction from unstructured data, which consists of two main steps as follows. One step is the extraction from unstructured data. ROCP firstly performs the text pre-processing to construct a Vector Space Model from the domain documents. Afterwards, the average cosine-similarity algorithm is used to achieve the document validation. Subsequently, a new algorithm MVPW, which extends from the classical algorithm TF-IDF, is proposed to implement the terminology extraction. The other step is the ontology construction from terminologies, which needs more manual work of the domain experts. After the 3-layers taxonomy of the terminologies, a temporary ontology model with separated ontology nodes is constructed. Subsequently, a new parameter terminology hyponymy height (THH) and corresponding algorithm are proposed to make it convenient for domain experts to construct the hyponymy of ontology classes. In the experiments, we firstly compare the recall and precision of MVPW and TF-IDF algorithm. The experimental results indicate that the MPVW algorithm has obvious advantage in terminology extraction. Afterwards, we make statistics of the manual operation time to verify the efficiency of the semi-automatic ontology construction.

ROCP has been used in space debris mitigation domain as a part of a decision support system. With the help of ROCP, domain experts can (1) rapidly construct a domain ontology which can provide decision support to deal with new problems and (2) reduce communication barriers with information experts. Additionally, other users can (3) have better understanding of space debris mitigation domain through a lot of related knowledge in the ontology.

We have discussed that ROCP can provide a great convenience for domain experts to rapidly construct a domain ontology. However, ROCP still needs further improvement. The limitation is that non-taxonomy relationships can not be perfectly extracted. Besides, ROCP faces a challenge of ensuring the accuracy of big domain ontology construction.

In the future, (1) we can use statistics-based methods to achieve automatic ontology relationships extraction. For example, the Bayesian-network may be used to derive ontology relationships from XML formats of Word or PDF documents. (2) We can derive ontology from big open source knowledge base such as DBPedia and Yago. (3) We can apply the methodology of ROCP to other domains. (4) In addition, more applications about domain data integration, such as a domain micro-encyclopedia, can be achieved with the help of ROCP.

Acknowledgements

This work is supported in National Natural Science Foundation of China under Grant No.51271033, Hebei Natural Science Foundation under Grant No. F2013208107, Science Fund Project of Hebei Education Department under Grant No. QN20131138.

Competing Interests

The authors have no competing interests to declare.

References

Ashino, T. 2010. Materials ontology: an infrastructure for exchanging materials information and knowledge. *DataScience Journal*, 9(1): 54–61. DOI: https://doi.org/10.2481/dsj.008-041

Astrakhantsev, NA and **Turdakov, DY.** 2013. Automatic construction and enrichment of informal ontologies: a survey. *Programming and Computer Software*, 39(1): 34–42. DOI: https://doi.org/10.1134/S0361768813010039

Bernth, A, Mccord, M and **Warburton, K.** 2003. Terminology extraction for global content management. *Terminology International Journal of Theoretical & Applied Issues in Specialized Communication*, 9(1): 51–70. DOI: https://doi.org/10.1075/term.9.1.04ber

Choi, SP and **Myaeng, SH.** 2012. Terminological paraphrase extraction from scientific literature based on predicate argument tuples. *Journal of Information Science*, 38(6): 593–611. DOI: https://doi.org/10.1177/0165551512459920

Chung, TM. 2002. A corpus comparison approach for terminology extraction. *Terminology*, 9(2): 221–246. DOI: https://doi.org/10.1075/term.9.2.05chu

Dahab, MY, Hassan, HA and **Rafea, A.** 2008. TextOntoEx: Automatic ontology construction from natural English text. *Expert Systems with Applications*, 34(2): 1474–1480. DOI: https://doi.org/10.1016/j.eswa.2007.01.043

Dutta, B. 2015. Yamo: yet another methodology for large-scale faceted ontology construction. *Journal of Knowledge Management,* 19(1): 6–24. DOI: https://doi.org/10.1108/JKM-10-2014-0439

Erdmann, M, Nakayama, K, Hara, T and **Nishio, S.** 2009. Improving the extraction of bilingual terminology from wikipedia. *Acm Transactions on Multimedia Computing Communications & Applications*, 5(4): 1729–1739. DOI: https://doi.org/10.1145/1596990.1596995

Faure, D, Nédellec, C and **Rouveirol, C.** 1998. Acquisition of semantic knowledge using machine learning methods. *The system ASIUM technical report number ICS-TR-88-16.*

Gruber, TR. 1993. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2): 199–220. DOI: https://doi.org/10.1006/knac.1993.1008

Kara, D and **David, B.** 2013. Automating the construction of gene ontologies. *Nature Biotechnology*, 31(1): 34–35. DOI: https://doi.org/10.1038/nbt.2476

Küçük, D and **Arslan, Y.** 2014. Semi-automatic construction of a domain ontology for wind energy using wikipedia articles. *Renewable Energy*, 62(3): 484–489. DOI: https://doi.org/10.1016/j.renene.2013.08.002

Lee, CS, Kao, YF, Kuo, YH and **Wang, MH.** 2007. Automated ontology construction for unstructured text documents. *Data and Knowledge Engineering*, 60(3): 547–566. DOI: https://doi.org/10.1016/j.datak.2006.04.001

Liu, Y, Chen, XF, Sui, Z, Wang, H and **Zhou, Y.** 2008. On automatic construction of based-NLP Chinese medicine ontology concept's description architecture. In: *2008 International Conference on Audio, Language and Image Processing*, 50–55.

Macken, L, Lefever, E and **Hoste, V.** 2013. Texsis: bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology*, 19(1): 1–30. DOI: https://doi.org/10.1075/term.19.1.01mac

Maedche, A and **Staab, S.** 2000. Semi-automatic engineering of ontologies from text. In: *Proceedings of the 12th international conference on software engineering and knowledge engineering*, 231–239.

Maedche, A and **Staab, S.** 2001. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2): 72–79. DOI: https://doi.org/10.1109/5254.920602

Manganello, F, Falsetti, C, Spalazzi, L and **Leo, T.** 2013. Pks: an ontology-based learning construct for lifelong learners. *Educational Technology & Society*, 16(1): 104–117.

Marciniak, M and **Mykowiecka, A.** 2014. Terminology extraction from medical texts in Polish. *Journals of Biomed Semantics*, 5(1): 1–14. DOI: https://doi.org/10.1186/2041-1480-5-24

Navigli, R, Velardi, P and **Gangemi, A.** 2003. Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(1): 22–31. DOI: https://doi.org/10.1109/MIS.2003.1179190

Niu, J and **Issa, RRA.** 2015. Developing taxonomy for the domain ontology of construction contractual semantics: a case study on the AIA A201 document. *Advanced Engineering Informatics*, 29(3): 472–482. DOI: https://doi.org/10.1109/MIS.2003.1179190

Pascal, H, Markus, K, Parsia, B, Peter, F and **Rudolph, S.** 2007. OWL Primer. Available at: www.w3.org/2007/OWL/wiki/Primer (accessed November 18, 2015).

Rios-Alvarado, AB, Lopez-Arevalo, I and **Sosa-Sosa, VJ.** 2013. Learning concept hierarchies from textual resources for ontologies construction. *Expert Systems with Applications*, 40(15): 5907–5915. DOI: https://doi.org/10.1016/j.eswa.2013.05.005

Sánchez, D and **Moreno, A.** 2008. Learning non-taxonomic relationships from web documents for domain ontology construction. *Data & Knowledge Engineering*, 63(3): 600–623. DOI: https://doi.org/10.1016/j.datak.2007.10.001

Sellami, Z and **Camps, V.** 2012. DYNAMO-MAS: A Multi-Agent System for Building and Evolving Ontologies from Text. *Advances on Practical Applications of Agents and Multi-Agent Systems*. Springer: Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-642-28786-2_38

Shamsfard, M and **Barforoush, AA.** 2003. The state of the art in ontology learning: a framework for comparison. *Knowledge Engineering Review*, 18(4): 293–316. DOI: https://doi.org/10.1016/j.datak.2007.10.001

Wei, Y, Wang, R, Hu, Y and **Wang, X.** 2012. From Web Resources to Agricultural Ontology: a Method for Semi-Automatic Construction. *Journal of Integrative Agriculture*, 11(5): 775–783. DOI: https://doi.org/10.1016/S2095-3119(12)60067-7

Weng, S, Tsai, H, Liu, S and **Hsu, C.** 2006. Ontology construction for information classification. *Expert Systems with Applications*, 31(1): 1–12. DOI: https://doi.org/10.1016/j.eswa.2005.09.007

Zhang, C and **Wu, D.** 2012. Bilingual terminology extraction using multi-level termhood. *The Electronic Library*, 30(2): 295–309. DOI: https://doi.org/10.1108/02640471211221395

Zhao, CC, Dong, C, Zhang, XM. 2016. EM3B2 – a semantic integration engine for materials science. *Program*, 50(2): 58–82. DOI: https://doi.org/10.1108/PROG-01-2015-0004

How to cite this article: Zhao, C, Dong, C and Zhang, X. 2018. ROCP: A Rapid Ontology Construction Platform from Unstructured Data. *Data Science Journal,* 17: 23, pp. 1–16. DOI: https://doi.org/10.5334/dsj-2018-023

Submitted: 14 February 2017 **Accepted:** 02 May 2017 **Published:** 25 September 2018

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS