CLOUD ONTOLOGY

ONTOLOGY-BASED SYSTEM TO DISCOVER CLOUD INFRASTRUCTURE SERVICES

Abstract

The Cloud infrastructure services landscape advances steadily leaving users in the agony of choice. As a result, Cloud service identification and discovery remains a hard problem due to different service descriptions, non-standardized naming conventions and heterogeneous types and features of Cloud services. With the increase in the number of services, there has also been an increase in demand and adoption of cloud services making cloud service identification and discovery a challenging task. Thus, selecting an appropriate cloud service according to consumer requirements is a daunting task, especially for applications that use a composition of different cloud services.

In this paper, we present an ontology, the Cloud Computing Ontology (CoCoOn) that defines functional and non-functional concepts, attributes and relations of infrastructure services. In this paper, we have designed an ontology-based cloud infrastructure service discovery and selection system that defines functional and non-functional concepts, attributes and relations of infrastructure services.

Research summary

In recent years, significant attention has been spent on mining Online Social Network (OSN) in real-time. Decision analytic methods, from marketing to emergency management, from politics to business and management benefit of real-time or near real-time event processing. Event detection is for example crucial in traffic management, fire control, TV show hosting and smart-city management systems. In combination with other data sources, OSN can boost complex decision making and risk management methodologies. For instance, Twitter has been effectively exploited in many real-world incidents to communicate disaster warnings and disseminate information, capture the evolving trends, control resource consumption, or discover effective mitigation strategies bottom-up.

The reliability of the social network-based event analysis depends on several factors. Social media are general-purpose communication platforms, for this reason filtering the activities that are related to the domain of analysis is crucial to avoid introducing selection bias.

Twitter, for example, gives researchers a gateway providing them with billions of information about users' links, written contents, and community circles, giving analytics a gateway for improving their algorithms primarily in Natural Language Processing, Link Prediction, Community Detection and Sentiment Analysis. However, such methods require a relevant amount of data to be processed that impacts on the timeliness of the result provided as well as the resources needed.

Objectives

The objective is how to determine a good ontology for a recommender system for a social network.

Overview

Experiment outline

In this section, we present the framework. Accordingly, we discuss the system hierarchy organizing our architecture with its abstraction layer, we detail the workflow guiding the mining procedures, the network space exploring algorithms and the different filtering strategies that make the system effective.

Relevant literature

Selecting an appropriate cloud service according to consumer requirements is a daunting task, especially for applications that use a composition of different cloud services.

Online Social Network (OSN) is considered a key source of information for real-time decision making. However, several constraints lead to decreasing the amount of information that a researcher can have while increasing the time of social network mining procedures.

Social media are general-purpose communication platforms, for this reason filtering the activities that are related to the domain of analysis is crucial to avoid introducing selection bias.

To do so, a new class of agile and cost-effective methods and tools has been proposed to support operators in analyzing at a deeper level and closer to real-time the huge amount of data generated by OSN is paramount. Several approaches have been proposed for mining OSNs while limiting the time and the budget required for mining. We are seeking machine learning to find people's entity from their shared contents, and ontologies to build a knowledge graph that maps the relations between the accounts and their environment. Additionally, we consider the platform for building ontology enhanced knowledge graph and use it for recommendation purposes.

Proposed analysis.

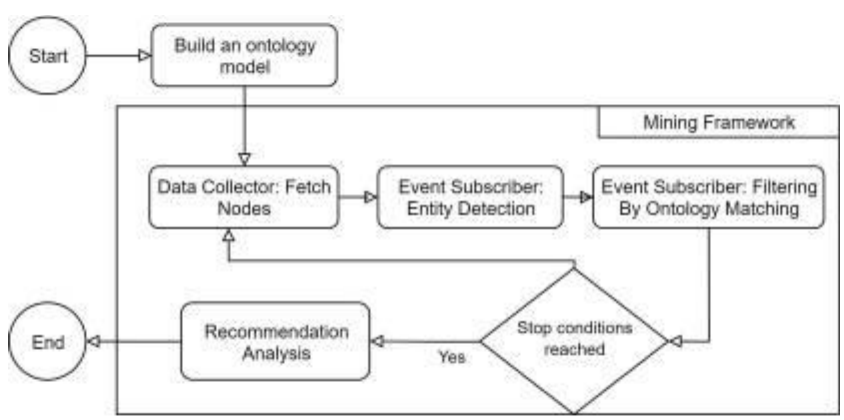
This paper proposes a framework for sampling Online Social Network (OSN). Domain knowledge is used to define tailored strategies that can decrease the budget and time required for mining while increasing the recall. An ontology supports our filtering layer in evaluating the relatedness of nodes. Our approach demonstrates that the same mechanism can be advanced to prompt recommendations to users**.** We propose a mining platform to help researchers and data collector to mine and directly analyze social networks, defining API-specific and budget-constrained strategies able to filter data collection based on concurrent sampling and ontology-enhanced filtering algorithms. To test the approach, we exploited it in creating a content-based recommender system. In our proposed architecture, recommendations are the results of the graph projection of social network nodes with their relationship and roles.

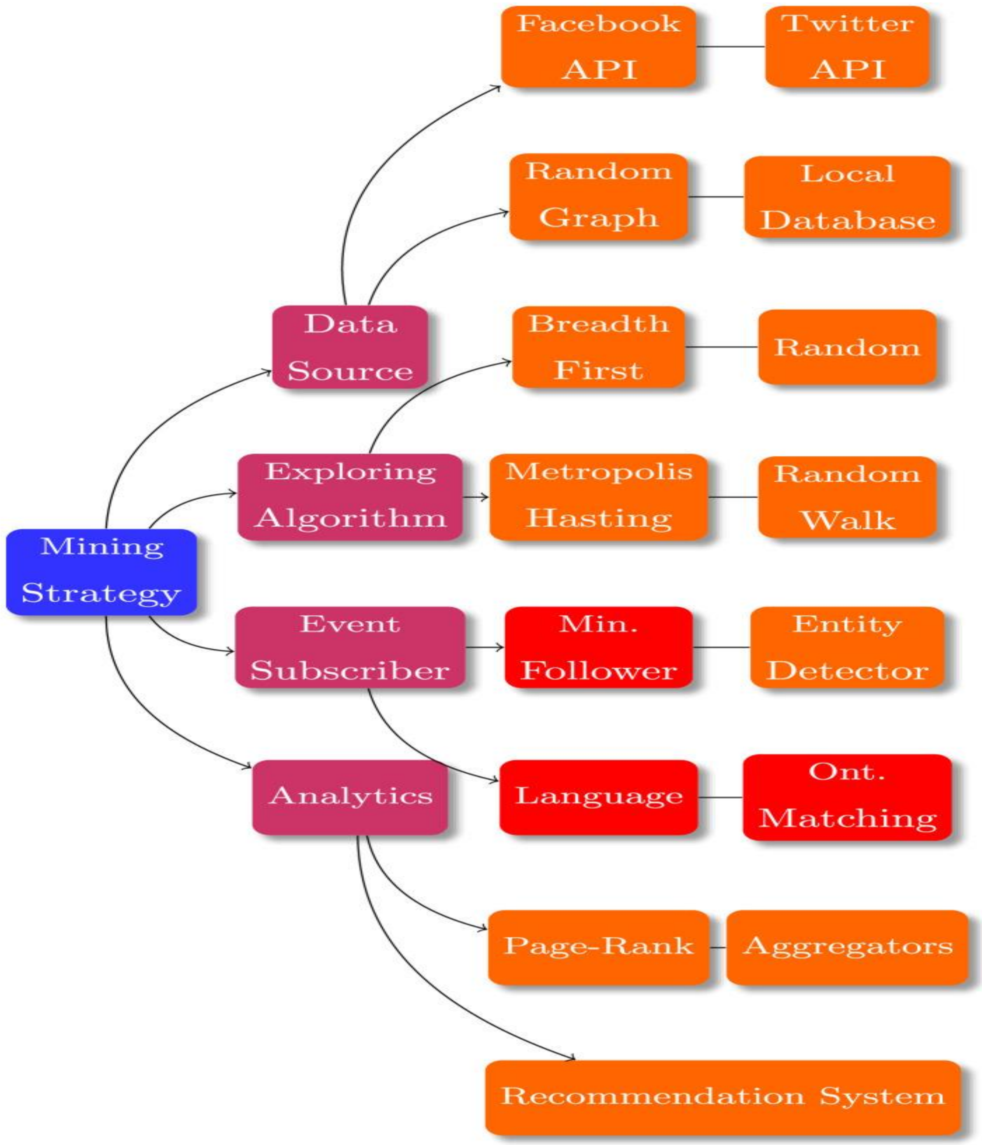
Experiment

Systems setup

A Social Network is an ever-expanding data source. For this reason, an effective mining procedure must rely on real-time data collection. Also, an evolving domain may require to extend the computational capacity of the system. In order to address this issue, we used different technologies that helped to achieve maximum scalability in our architecture, by interfacing separated components using abstract classes. It comprises of the hierarchy of our architecture listing the abstract classes that compose it.

It describes in detail the data processing workflow of our system. The process starts when the network space exploring algorithm navigates the social network graph choosing the first node. The results differ based on the algorithm selected while defining the strategy. The next step is scanning the selected node, thus allowing further routes of the social graph to explore. Moreover, more detail on the nodes demands additional specialized requests to fetch them.





Benchmarking

An important part of the framework depends on the space exploring algorithms as they represent a crucial component when defining a strategy. With such algorithms, we can navigate through social networks and embed them as a graph in our data storing system. Graph Databases are used to store OSN accounts and posted-contents as nodes, while the link between them is captured by edges showing a graphical representation of the stored data. Edges can be labeled to define the type of relationship interconnecting two nodes, i.e. friends, follower, co-authors, etc. A post and its originator can be presented as two nodes connected by an edge labeled as posted-it. A post and a reader can be connected by like-it, hate-it. Addressing the limits posed by OSN providers, navigation algorithms combined with filters are used as data samplers working on a subset of data selected to be representative of the whole dataset. Sampling social network reduces the time and budget required to collect the minimum information needed. The algorithms that are available in the platform can be divided into two groups: deterministic like Breadth-First, probabilistic like Forest Fire, Random Walker, and Metropolis Hasting. Additional focus has been given to probabilistic approaches, which can be supplied with hyper-parameters that are capable of changing the shape of the mined data, thus fitting more for data sampling work, while further widening the traversed space of the network. Frequent hyper-parameter used in the platform are forward weight and iterations. Forward weight controls the onward and backward jump rate of the random walker, the number range between 0 and 1, and the higher the number to deeper the level explored by the algorithm. This parameter significantly affects the accuracy of the results.

Simulations.

Data collection is handled by our mining framework using a mining strategy. For example, the level breakdown of a space exploring algorithm allows us to focus on a specific area of the network rather than the whole, thus reducing a large amount of data processing that may lead to no or few results. In our case, the focus was on the lower level. Therefore, as a network space exploring algorithm, Breadth-First could be a relevant algorithm for this task. However, other algorithms can be tuned to focus also on the lower levels. For example, with a small forward probability, Random Walker and Metropolis Hasting emphasize the backward moving rather than moving forward, causing the lower-level nodes more significant.

	S1	S2	S3	S4
Exploring algorithm	BF	BF	RW	MH
Account fetched	–	50	50	50
Max depth	8	3	–	–
Iterations	–	–	500	500
Forward weight	–	–	0.8	0.8
Distribution	–	–	–	Normal

Results

Measurements/Metrics tables

Table 1

Set 1 data overview.

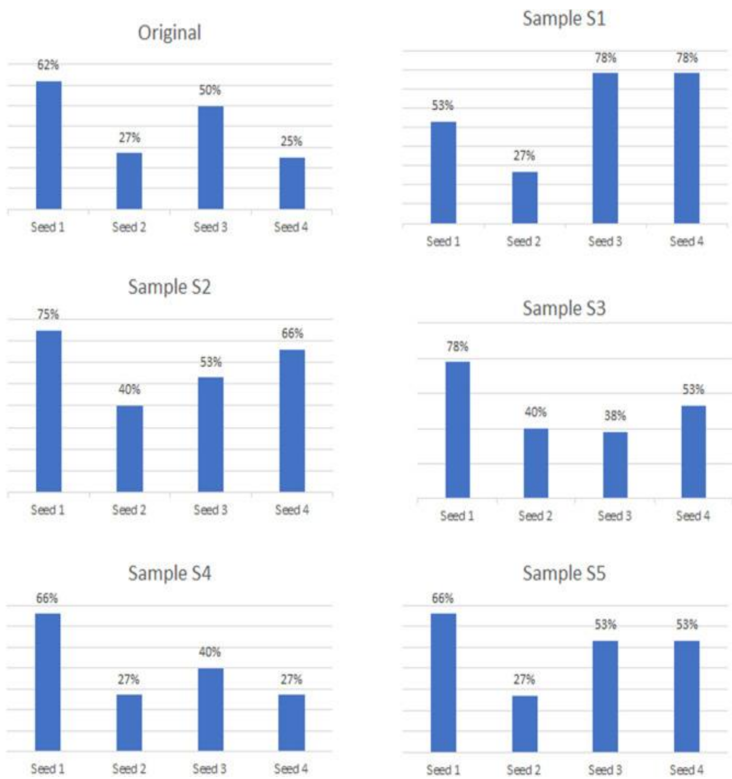
Seed	1	2	3	4
Italy	62%	27%	50%	25%

Table 2

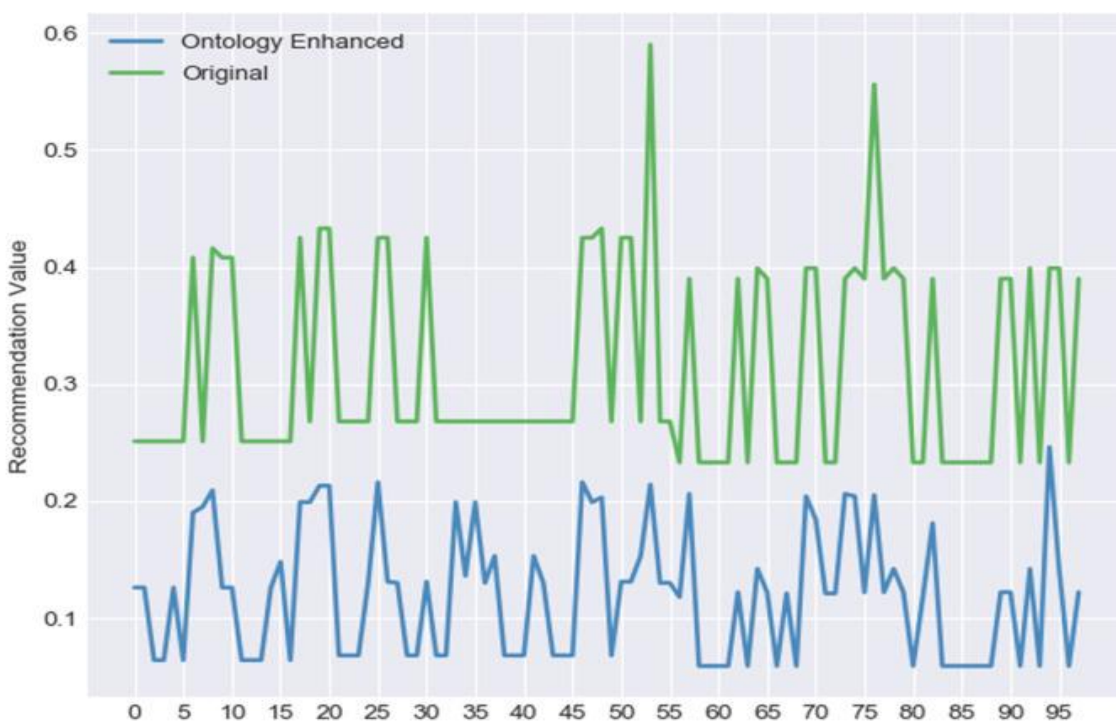
Set 1 mining strategies.

	S1	S2	S3	S4	S5
Exploring algorithm	BF	RW	MH	RW	MH
Account fetched	10%	10%	10%	5%	5%
Location filter	Italy	Italy	Italy	Italy	Italy
Iterations	–	500	500	500	500
Forward weight	–	0.2	0.2	0.2	0.2
Distribution	–	–	Normal	–	Normal

Visualizations



Ontology Enhanced
Original
Recommendation Value
0.1
0.2
0.3
0.4
0.5
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95



Ontology Enhanced
Original
Recommendation Value
0.1
0.2
0.3
0.4
0.5
0.6
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95

Evaluation.

Our test cases and experimental results emphasize the importance of the strategy definition step in our social miner and the application of ontologies on the knowledge graph in the domain of recommendation analysis.

Conclusion - Success/Failure results, outputs, caveats/cautions.

We have shown how the system enables one to discover appropriate services optimally as requested by consumers.

OSNs can be considered as the main source of information for any Big Data analysis study.

We have introduced domain-specific sampling strategies that serve as input for platform miners. Moreover, we have demonstrated the capabilities of our platform by employing ontologies to reinforce our graphical representation with stronger relations and used them as part of the proposed recommendation system. In our experiments, we have explored the importance of the strategy definition step as well as its impact on the quality of the results. Additionally, we have illustrated the implication of the ontologies on the graph and have used it on a real word dataset for a recommendation system.

Bibliography - Reference citations (Chicago style - AMS/AIP or ACM/IEEE).

1. Nsouli A., Mourad A., Azar D. *Proceedings of the IEEE International Wireless Communications and Mobile Computing Conference(IWCMC 2018)* IEEE; 2018. Towards proactive social learning approach for traffic event detection based on arabic tweets. [Google Scholar]

2. Abel F., Hauff C., Houben G.J., Stronkman R., Tao K. *Proceedings of the 21st Annual Conference on World Wide Web Companion.* 2012. Twitcident: Fighting fire with information from social web streams. [CrossRef] [Google Scholar]

3. Crisci A., Grasso V., Nesi P., Pantaleo G., Paoli I., Zaza I. Multimedia Tools and Applications, Springer.2017; 2017. Predicting TV programme Audience by Using Twitter Based Metrics. [CrossRef] [Google Scholar]

4. Lytras M.D., Raghavan V., Damiani E. Big data and data analytics research. *Int. J. Semant. Web Inf. Syst.* 2017;13(1):1–10. doi: 10.4018/ijswis.2017010101. [CrossRef] [Google Scholar]

5. Lytras M.D., Visvizi A., Jussila J. Social media mining for smart cities and smart villages research. *Soft Comput.* 2020;24(15):10983–10987. doi: 10.1007/s00500-020-05084-3. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

6. Rezvanian A., Moradabadi B., Ghavipour M., Daliri Khomami M.M., Meybodi M.R. *Learning Automata Approach for Social Networks.* Springer International Publishing; 2019. (Studies in Computational Intelligence). [CrossRef] [Google Scholar]

7. M.B. Habib, P. Apers, M. van Keulen, Neogeography: The challenge of channelling large and ill-behaved data streams, in: In the Proceeding of the 27th IEEE International Conference on Data Engineering Workshops, 2011, pp. 284–287.

8. Hossmann T., Legendre F., Carta P., Gunningberg P., Rohner C. *In the Proceedings of the 3rd Extreme Conference on Communication the Amazon Expedition - ExtremeCom '11.* ACM Press; 2011. Twitter in disaster mode. [CrossRef] [Google Scholar]

9. A B., And J., MacEachren A., Robinson A., Jaiswal A., Pezanowski S., Savelyev A., Blanford J., Mitra P. 2011. Geo-Twitter analytics: Applications in crisis management. [Google Scholar]

Code & Data

Reference Documentation - Software commands, inputs, installation.

Data Sources - Links, downloads, access information.

Data fetched from live data used the necessary APIs

Source Code - Listings, documentation, dependencies (open-source).

```
Algorithm 1: Mining Process
Data: strategy
let root denote the starting node of the graph;
extract navigator from strategy;
extract dataSource from strategy;
extract events from strategy;
observe(navigator, root);
analyse(root);
Function observe(navigator: Navigator, root: Graph) : void is
    scan(root);
    foreach seed ∈ root do
        navigate(navigator, seed);
    end
end
Function navigate(navigator: Navigator, node: Node) : void is
    do
        broadcast(node, "PreScan");
        if node == dead then
            broadcast(node, "Failed");
        else if node ≠ scannable then
            broadcast(node, "Failed");
        else
            scan(node);
            broadcast(node, "PostScan");
            if node == dead then
                broadcast(node, "Failed");
            else
                fetch(node);
                broadcast(node, "PostFetch");
            cache();
        node = navigator.next(node);
    while node ≠ null||stopped;
end
```

Algorithm 2: RandomWalk Navigator

Function *next(node: Node) : Node* **is**
  let R be a random generated number;
  extract $strategy.weight$ into W;
  extract $strategy.maxDepth$ into M;
  initialise $f := forward$ if $node == root$;
  initialise $f := backgward$ if $node.level == M$;
  initialise $f := backgward$ if $node$ is Leaf;
  initialise $f := forward$ if $R < W$ else $backward$;
  if $f == forward$ **then**
    return a *child* selected randomly from *node*;
  else
    if $parent(parent(node)) \neq null$ **then**
      initialise $ancestor := parent(parent(node))$;
    else
      initialise $ancestor := parent(node)$;
    return a *child* selected randomly from *ancestor*;
end