A Question and Answering System for Management of Cloud Service Level Agreements

Sudip Mittal, Aditi Gupta, Karuna P. Joshi, Claudia Pearce and Anupam Joshi
University of Maryland, Baltimore County, Baltimore, MD 21250, USA
Email: {smittal1,adigupta,kjoshi1,cpearce,joshi}@umbc.edu

***Abstract*—One of the key challenges faced by consumers is to efficiently manage and monitor the quality of cloud services. To manage service performance, consumers have to validate rules embedded in cloud legal contracts, such as Service Level Agreements (SLA) and Privacy Policies, that are available as text documents. Currently this analysis requires significant time and manual labor and is thus inefficient. We propose a cognitive assistant that can be used to manage cloud legal documents by automatically extracting knowledge (terms, rules, constraints) from them and reasoning over it to validate service performance. In this paper, we present this Question and Answering (Q&A) system that can be used to analyze and obtain information from the SLA documents. We have created a knowledgebase of Cloud SLAs from various providers which forms the underlying repository of our Q&A system. We utilized techniques from natural language processing and semantic web (RDF, SPARQL and Fuseki server) to build our framework. We also present sample queries on how a consumer can compute metrics such as service credit.**

I. Introduction

Cloud based solutions are increasingly being adopted by organizations to store and manage their enterprise data [1]. Amazon Web Services (AWS) and Microsoft's Azure are a couple of the leading service providers in cloud computing[1]. On one hand, cloud computing has multiple benefits such as no up-front investment, low operating cost and being highly scalable, while on the other hand like any other new technology, it brings new challenges [2]. Some typical issues in cloud computing arise in management, governance, maintenance, trust and privacy of cloud computing services. National Institute of Standards and Technology (NIST), one of the organizations leading the initiative to build standardized metrics and policy for cloud computing, identified comparing and choosing between different cloud service providers and defining / enforcing appropriate Service Level Agreements as one of the challenges faced by consumers [3]. We have developed a semantically-rich framework to automatically analyze and reason over cloud legal documents, like SLAs, Privacy policies, terms of service documents etc. In this paper, we present our Legal Question and Answering system that can be used by cloud consumers to automatically analyze various cloud SLA documents using simple natural language questions.

Traditionally Q&A systems have been of two types, *Closed-domain* Q&A systems and *Open-domain* Q&A systems. Closed-domain systems deal with questions regarding a specific domain, they are able to exploit domain specific knowledge, and hence are easier to train than Open-domain systems. Open-domain systems rely on general ontologies and aim to answer questions about nearly everything.

A Question Answering (Q&A) system provides a user with a direct answer computed by consulting its knowledge base. We believe a Q&A system will enable the user to manage various cloud services better as it will provide a method to retrieve answers to various questions like: 'What is Google's uptime?', 'How is Amazon's service credit computed?', and 'How much service credit on Amazon will I get if my uptime was 98%?'

The first two questions above retrieve relevant information from various knowledge stores. The third one, requires computation and reasoning over the information obtained from the knowledge graph to answer the user query.

In our previous work we have created an integrated methodology to significantly automate the cloud service lifecycle using Semantic Web technologies and Text mining techniques [4], [5], [6], [7]. We have developed a semantically rich ontology to capture key elements of cloud SLAs [4]. Since a majority of the cloud legal documents are created as text documents for human consumption, we have applied existing text mining techniques to extract performance rules, measures and metrics [5], [6]. Next, we extracted rules and constraints in the form of deontic rules such as permissions and obligations from the SLA documents [7].

In this paper, our goal is to describe our cognitive agent based on a Question and Answer (Q&A) system to analyze legal documents associated with cloud services using various techniques such as machine learning and text mining.

II. Related Work

For our cloud services lifecycle framework [8], we have used semantic web technologies like OWL instead of WS-Agreement and WS-Negotiation protocol as we were able to more richly define the cloud SLA ontologies, thereby allowing us to incorporate different descriptions of the same SLA measure. We have divided the IT service lifecycle into five phases of: requirements; discovery; negotiation; composition; and consumption. We detail each phase and describe the ontologies that we have developed to represent the concepts and relationships for each phase [4].

[1] http://www.forbes.com/sites/benkepes/2015/03/04/new-stats-from-the-state-of-cloud-report/

Developing systems that can answer natural language queries from information in a database or collection of documents has been a goal for more than more than 45 years. The Text REtrieval Conference (TREC)[2], co-sponsored by the National Institute of Standards and Technology and U.S. Department of Defense runs a question answering track since 1999. The track mainly focused on obtaining answers to factual questions based on a corpus of news articles. Some sample questions for which the systems are evaluated were *"When was 3M founded?"* or *"Where is Merrill Lynch headquartered?"*. Mollá et al. [9] focus on the use of restricted domains for automated question answering. More recent question answering systems that have been developed are IBM Watson, AskMSR, and Wolfram Alpha[3] [10], [11], [12], [13]. These systems are based on knowledge base computational models where answers are computed by reasoning over an underlying knowledge base. Early question answering systems include MURAX [14], which was developed to answer general-knowledge questions using an on-line encyclopedia, and LUNAR [15], which answered questions about moon rocks and soil gathered by the Apollo 11 mission. WOLFIE [16] was a closed domain question answering system which used inductive logic programming methods to map natural language queries into executable logical form.

III. Methodology

Our research aims at building a Question and Answer (Q&A) system to analyze legal documents associated with cloud services based on knowledge extracted using various techniques such as machine learning and text mining. Using the Q&A system a user would be able to input her query regarding the cloud service and we would parse the information, link entities and extract the information from the corresponding knowledge graph. The information extracted would then be displayed to the user in form of text, graphs or tables. For example, a user may have a query like "What is the uptime provided by a particular service provider?". Figure 1 explains the architecture of the proposed system. In this paper we have used Amazon and Google's SLA to provide various examples [17], [18].

The first step in building a backend for the Q&A system is to create the ontology and populate the knowledge graphs for legal documents associated with cloud services. We developed a detailed ontology in OWL to represent cloud SLAs which is available in the public domain. We have developed programs to extract cloud metrics from publicly available cloud SLAs and have published our results in [4], [5], [6], [7].

A. Question and Answering System

We built a closed domain Question and Answering System by using various Semantic Web technologies including OWL [19], RDF [20] and SPARQL [21]. We use Stanford PoS Tagger [4] to create a tree-bank structure for the input query and then used it to query our knowledge base. We used the Jena Apache Fuseki [22] server graph store as a store for the cloud SLAs. A user can select the service provider for which she wants to analyze the SLA using a form in our framework. The framework will also extract the metrics and deontic expressions stored in the knowledge graph for that particular SLA and display them to the user. The system also allows a user to enter a natural language query to display various details and compute her service credit value based on her monthly uptime availability values. The purpose of this framework is to present to the user, main components of an SLA such as definitions, metrics, permissions and obligations which are stored in the knowledge graph and provides user the ability to query and reason over them.

1) Question Processing Module: A Question and Answering (Q&A) system provides a user with a direct answer computed by consulting its knowledge base. It is primarily a natural language user interface, where the user enters various question types including: facts, definitions, 'How', 'Why', hypothetical constrained questions. In the recent times, Q&A systems have gained widespread application as they provide a break from keyword-based query systems to a more Natural Language question-based system.

In our question processing module the user's query input is first passed to the Stanford PoS Tagger [23] to create a tree-bank for the question's phrasal structure. We use the question's phrasal structure to create their corresponding template based SPARQL queries. A similar template-based methodology was used by Unger et al [24].

2) Answering Module: In this section we describe the mapping between the question extracted in the previous module and then create a SPARQL query to extract data from our knowledge graph. Now, we demonstrate a use-case of how a reasoner can be built using first-order predicate logic to perform reasoning over the policies extracted from SLA documents. We then convert them into SPARQL queries to extract the relevant data from the Knowledge graph populated in previous sections.

We use the template based SPARQL Creation method by Unger et al. [24] and the tagged output of the question processing module to query our knowledge graph. We use the word marked as 'NNP'' (Proper Singular noun) to identify the service provider mentioned in the query. So for the tree-bank structure obtained for the query "What is Google's Uptime?" and "How is Amazon's Service Credit computed?", we select the service provider by looking at the tags created for the input question. Google and Amazon are tagged as 'NNP'' which helps us query the respective instances in the knowledge graph. An alternate to 'NNP'' is '$NNPS'$' which stands for proper plural noun. The second noun in the input query ('NN'' or 'NNS'') can be used to locate various properties like 'uptime' and 'service credit'. These 2 nouns present in the input query are then used to populate template based SPARQL queries.

An important task for the current system is to determine the type of the input question. As we mentioned earlier there are 4 typical types of questions: 'What', 'Where', 'How',

[2]http://trec.nist.gov/
[3]http://www.wolframalpha.com/about.html
[4]http://nlp.stanford.edu/software/tagger.shtml

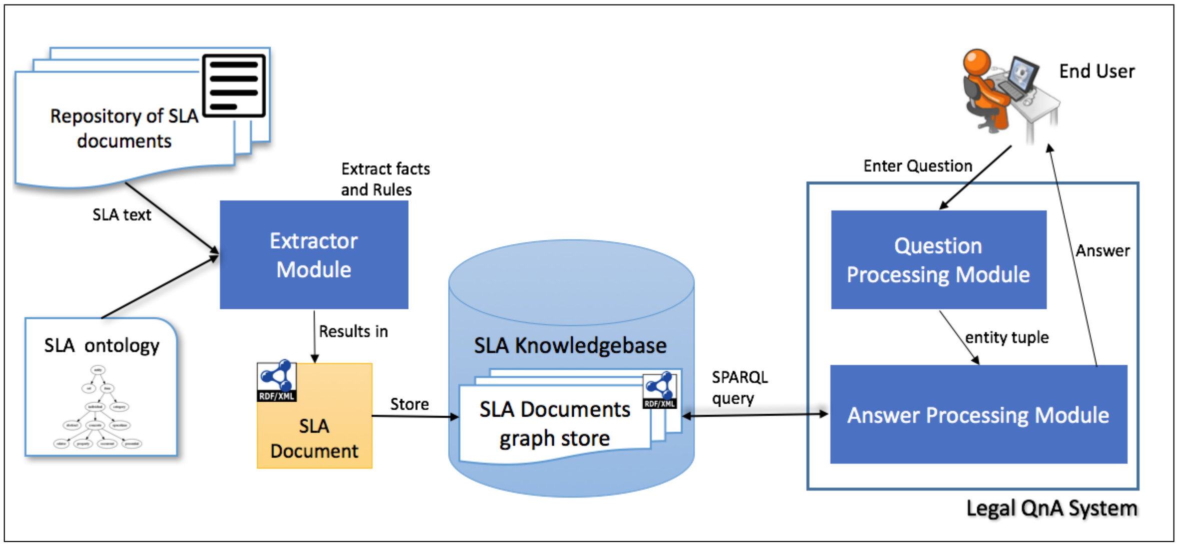

Fig. 1. System architecture for the Question and Answering system for cloud computing services.

and ‘When’. *What* deals with definitions, *where* deals with location, *when* deals with the temporal aspects and *how* pertains to description. Presence of these words help us to identify the type of question, which is then used to formulate the query to the knowledge base.

Using the tagged values and presence of question keywords we convert the input natural language question to its SPARQL query. For example, let the user’s question be “What is Google’s Uptime?” then the SPARQL query generated is:

```
SELECT ?y WHERE {
?Google <Uptime> ?y .
}
```

For the above question, our module extracts the service provider as ‘Google’, the property as ‘Uptime’ and the type of question as definition. Another example is: “How is Amazon’s service credit computed?”, for which we generate the query:

```
SELECT ?y WHERE {
?Amazon <hasServiceCredit> ?Service_Credit
?y ?x ?Service_Credit .
}
```

SPARQL (SPARQL Protocol and RDF Query Language) is a semantic query language with which we can query the data stored in the knowledge graph in RDF format [21] [20]. We used the Apache Jena Framework for running the SPARQL queries. Apache Jena is a open source Java framework used for building semantic web and linked data applications. [5] Figure 2 shows a sample SPARQL query output for extracting the Monthly uptime Percentage for a cloud service provider stored in the knowledge graph.

[5]https://jena.apache.org/index.html

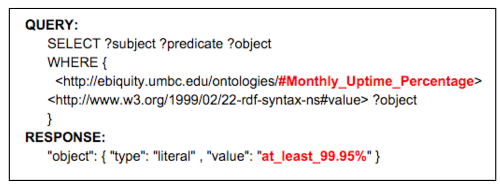

Fig. 2. A simple SPARQL query to extract the Monthly uptime Percentage for a cloud service provider stored in the knowledge graph.

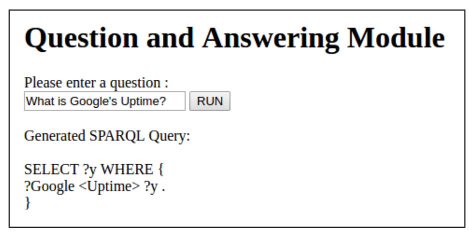

Fig. 3. A snapshot of the system user interface to input a natural language question as input and the corresponding SPARQL query generated.

B. Service Credit Computation Module

Now we describe how we can compute metrics such as service credit defined in SLA documents for cloud services. For example, if the ’availability of a cloud service’ provided by a service provider is less than the value in its SLA, the service provider may owe a service credit to its customers computed based on various factors. The details of service credit computation constitutes one of the key components in a SLA document. The amount of service credit a customer receives is dependent upon various factors such as the service monthly uptime availability and total usage of cloud services. Also, the SLA specifies certain time constraints within which the customer must send the request to receive the service credit.

Based on various factors to compute service credit, we created a new class in our cloud ontology to represent the properties of service credit for a service provider. For an instance of ontology created for each vendor, the values for each of the data properties are extracted from the SLA and populated in the ontology. The amount of service credit received by a user can be computed based on these properties. For some of the metrics, like the relation between monthly uptime percentages and uptime, we created multiple data properties to indicate different vales and thresholds. Also, the way service credit is computed and the values may vary from one service provider to another.

Next, using SPARQL queries as shown below we can compute the service credit a customer must receive based on the terms of the SLA extracted into the knowledge graph. The query below extracts the value from the knowledge graph based on the value of monthly uptime percentage.

```
SELECT * WHERE { ?x ?y ?z .
FILTER (?Monthly_Uptime_Percentage < 99.5 &&
    ?Monthly_Uptime_Percentage > 90 ) }
```

C. Evaluation

In order to evaluate our system we asked a few users to come up with a few factual questions that need to be answered based on the information present in various SLA documents. The users came up with questions like "What is $< service\_provider\_name >$ uptime?", "How is service credit computed?", etc. Our system was able to provide the correct answer 95% of the time.

IV. Conclusion and Future Work

In this paper, we present a closed domain question and answering system for managing cloud service level agreements. In the current system, the user inputs a simple natural language query about SLA documents of cloud services and the system computes an answer based on a knowledge graph. We use various semantic web technologies including RDF, SPARQL, etc. to compute answers for the user. User's questions are tagged by the Stanford PoS tagger, then the entities extracted are used to formulate SPARQL queries, which obtain the answers from the knowledge base of cloud service documents. We also implemented a reasoning module to extract answers for the user queries. As part of future work, we will expand the dataset for our system to include all legal documents associated with cloud services like terms of service, privacy policy and service level agreements. The current work demonstrates the proposed framework for simple queries, in the future we would like it to build support for more complex queries. We would also like to include a negotiation engine in the current system which would try to find the best service given various user constraints.

V. Acknowledgement

This research was partially supported by a DoD supplement to the NSF award #1439663 : NSF I/UCRC Center for Hybrid Multicore Productivity Research (CHMPR).

References

[1] "Gartner predicts infrastructure services will accelerate cloud computing growth." [Online]. Available: http://www.forbes.com/sites/louiscolumbus/2013/02/19/gartner-predicts-infrastructure-services-will-accelerate-cloud-computing-growth/
[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, 2010.
[3] "Cloud computing service metrics description." [Online]. Available: http://www.nist.gov/itl/cloud/upload/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf
[4] K. P. Joshi, Y. Yesha, and T. Finin, "Automating cloud services life cycle through semantic technologies," *Services Computing, IEEE Transactions on*, vol. 7, no. 1, pp. 109–122, 2014.
[5] S. Mittal, K. P. Joshi, C. Pearce, and A. Joshi, "Parallelizing natural language techniques for knowledge extraction from cloud service level agreements," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2831–2833.
[6] ——, "Automatic extraction of metrics from slas for cloud service management," in *IEEE International Conference on Cloud Engineering*, 2016.
[7] A. Gupta, S. Mittal, K. P. Joshi, C. Pearce, and A. Joshi, "Streamlining Management of Multiple Cloud Services," in *IEEE International Conference on Cloud Computing*. IEEE Computer Society, June 2016, p. 8.
[8] K. P. Joshi and C. Pearce, "Automating cloud service level agreements using semantic technologies," in *CLaw Workshop, IEEE International Conference on Cloud Engineering (IC2E)*. IEEE Computer Society.
[9] D. Mollá and J. L. Vicedo, "Special section on restricted-domain question answering," *Computational Linguistics*, vol. 33, no. 1.
[10] A. Ittycheriah, M. Franz, W.-J. Zhu, A. Ratnaparkhi, and R. J. Mammone, "Ibm's statistical question answering system." in *TREC*, 2000.
[11] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.
[12] M. Banko, E. Brill, S. Dumais, and J. Lin, "Askmsr: Question answering using the worldwide web," in *Proceedings of 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, 2002, pp. 7–9.
[13] E. Brill, S. Dumais, and M. Banko, "An analysis of the askmsr question-answering system," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 257–264.
[14] J. Kupiec, "Murax: A robust linguistic approach for question answering using an on-line encyclopedia," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1993, pp. 181–190.
[15] W. A. Woods and R. Kaplan, "Lunar rocks in natural english: Explorations in natural language question answering," *Linguistic structures processing*, vol. 5, pp. 521–569, 1977.
[16] C. A. Thompson and R. J. Mooney, "Automatic construction of semantic lexicons for learning natural language interfaces," in *AAAI/IAAI*, 1999, pp. 487–493.
[17] "Amazon ec2 service level agreement." [Online]. Available: http://aws.amazon.com/ec2/sla/
[18] "Google apps business agreement." [Online]. Available: https://www.google.com/intx/en_in/work/apps/terms/2013/1/premier_terms.html
[19] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.
[20] O. Lassila, R. R. Swick *et al.*, "Resource description framework (rdf) model and syntax specification," 1998.
[21] "Sparql 1.1 overview." [Online]. Available: http://www.w3.org/TR/sparql11-overview/
[22] "Jena apache fuseki: serving rdf data over http,." [Online]. Available: http://jena.apache.org/documentation/serving_data/index.html
[23] "The stanford parser: A statistical parser." [Online]. Available: http://nlp.stanford.edu/software/lex-parser.shtml
[24] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano, "Template-based question answering over rdf data," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 639–648.