# Parallelizing Natural Language Techniques for Knowledge Extraction from Cloud Service Level Agreements.

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

*Abstract*—To efficiently utilize their cloud based services, consumers have to continuously monitor and manage the Service Level Agreements (SLA) that define the service performance measures. Currently this is still a time and labor intensive process since the SLAs are primarily stored as text documents. We have significantly automated the process of extracting, managing and monitoring cloud SLAs using natural language processing techniques and Semantic Web technologies. In this paper we describe our prototype system that uses a Hadoop cluster to extract knowledge from unstructured legal text documents. For this prototype we have considered publicly available SLA/terms of service documents of various cloud providers. We use established natural language processing techniques in parallel to speed up cloud legal knowledge base creation. Our system considerably speeds up knowledge base creation and can also be used in other domains that have unstructured data.

*Keywords*-Knowledge Extraction, Distributed Systems, Data Mining.

## I. INTRODUCTION

A knowledge base can be used to store complex information extracted from other structured and unstructured sources. We are creating a 'Legal Knowledge Base' that will store facts extracted form various legal documents. Currently, legal documents like, Service Level Agreements (SLAs), contracts, compliance and regulatory policies, privacy policies, etc. are managed as plain text files meant principally for human consumption. Creating this legal knowledge base will make these documents machine understandable. Once we create our Legal Knowledge Base, we can reason over it to find answers to specific legal questions. Creation of a Legal Knowledge Base is the first step to create a Legal question-answering system.

We begin by creating a knowledge base from various Service Level Agreement (SLA) documents. We extract various SLA Metrics and Term Definitions and store them in subject-predicate-object triples in the popular RDF [1] graph which can be queried by a user using a SPARQL[2] endpoint. However, this system needs to be highly scalable to deal with a large number of potential documents that contain information relevant to the query.

Our Legal Knowledge Base can help consumers monitor, contrast and analyze SLA documents for different cloud service providers. We envision a system which will maintain knowledge about various legal terms and clauses contained in SLAs, compliance and regulatory policies, contracts, privacy documents etc. In this paper we describe the techniques that we have developed to extract knowledge from various cloud SLA documents using a Hadoop based system. Section II covers the related work in this area. In sections III & IV we describe the prototype system that we have created as a proof of our concept. We end with the conclusions and future work.

## II. RELATED WORK

Researchers have applied Natural Language Processing (NLP) techniques to extract information from text documents. In Rusu et. al. [3] the authors suggest an approach to extract subject-predicate-object triplets. They generate Parse Trees from English sentences and extract triplets from the parse trees. Etzioni et. al. [4] developed a system to automate the process of extracting large collections of facts from the Web in an unsupervised, domain-independent, and scalable manner. Etzioni et. al used Pattern Learning to address this challenge. Various textual information extraction and retrieval systems have been proposed in [5], [6]. Another important natural language technique used for information extraction from unstructured text is 'Noun Phrase Extraction'. Rusu et. al. in [3] show how to create triplets by considering 'Noun Phrases' obtained by using various part-of-speech taggers. Similar techniques have also been suggested in [4]. Niu et. al. [7] [8] suggest various machine learning and language based methods for knowledge base creation.

## III. PARALLEL KNOWLEDGE EXTRACTION SYSTEM

In this section we describe our system to extract in parallel, subject-predicate-object triples from cloud SLA documents. We used a configurable Apache Hadoop [9] cluster of 2 to 4 nodes with Natural Language Took Kit (NLTK) [10], Stanford PoS Tagger[11] and CMU Link Parser [12] installed. Our system is divided into two sub systems 'Extractor' and 'Assessor' (Figure 1).

### A. Extractor

Extractor takes a SLA document as input and then extracts all SLA metrics and term definitions found in that document.
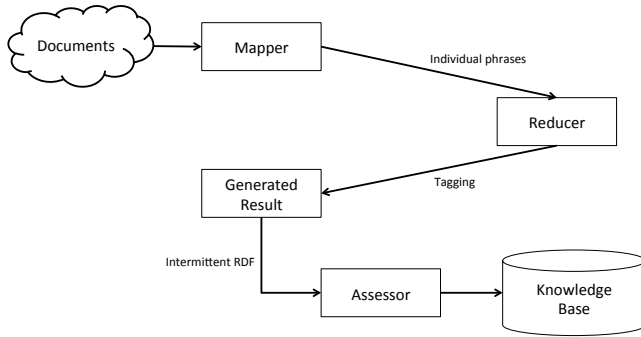
Figure 1. Architecture Diagram.

It is the core module of our system and is based on domain-specific English patterns and rules. Our extraction technique is based on generally followed rules for writing legal documents [13]. It first splits a document into individual sentences and then uses natural language techniques like 'Pattern Learning' and 'Noun Phrase Extraction' to extract knowledge.

Pattern Learning involves learning a few extraction patterns which are then used to filter out term definitions from unstructured SLA documents. A similar technique was used by Etzioni et. al. [4] to extract city names from unstructured data and Sipos et. al. [14] to extract triples from text. Rules used in our system are listed in Table I.

In order to create triples we use the the technique of 'Noun Phrase Extraction' [3], [15], [4]. In this technique we look at the 'Noun Phrase' part of the sentence found in the treebank structure generated by using the Stanford PoS Tagger[11] and CMU Link Parser [12]. We then use these noun phrases to create triplets.

| Patterns |
|---|
| X is defined Y |
| X means Y |
| X is calculated Y |
| X is Y |
| Keywords |
| 'is defined' |
| 'means' |
| 'is calculated' |
| 'is' |
| Constraint |
| X is a quoted, bold, underlined or italicised text. |

Table I
PATTERN BASED RULES FOR OUR EXTRACTOR

In our implementation of a Hadoop cluster the map function emits individual sentences and the reduce function creates RDF statements by matching statements to patterns and parsing treebank structures. An example iteration can be found in Figure 2.

### B. Assessor

We assess the quality of our extractions using a simple data type comparison with the Cloud SLA Ontology [16]. We verified the quality for a few documents manually and found that our data type comparison gave us good results. We can in the future develop an independent machine learning nodule to asses the system's output.

### C. Knowledge Base Creation

The main aim of our system is to create a knowledge base for various documents like cloud SLAs, legal documents, contracts, agreements etc. We aim to store the knowledge base as an RDF [1] graph and provide a SPARQL[2] endpoint for the user to query.

## IV. PERFORMANCE GAIN

To compare the performance of our parallel system we compare the time required to create our legal knowledge base from different number of identical documents with a single threaded system. As per our experiments, the single threaded system performs better when the number of documents is small. However as we keep on increasing the number of documents, the Hadoop-based parallel system outperforms the single threaded system. For our experiment we use 4 identically configured machines with 8 Gigabytes of RAM, and a quad-core 3.2 GHz processor.
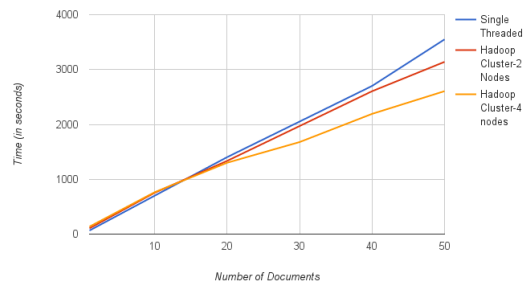


Figure 3. Time required to create knowledge base with different number of documents for the single threaded, 2 node cluster and 4 node cluster.

## V. CONCLUSIONS & FUTURE WORK

In this paper, we describe our prototype system which parallelly extracts knowledge from legal documents and create a knowledge base for the same. We envision a system which will have knowledge about various legal documents, contracts, agreements etc. and will be able to automatically suggest a service based on one's needs. We achieve a considerable speed up by parallelizing our system. We believe that parallel knowledge base population is faster and can be also used in other domains to extract knowledge. In the
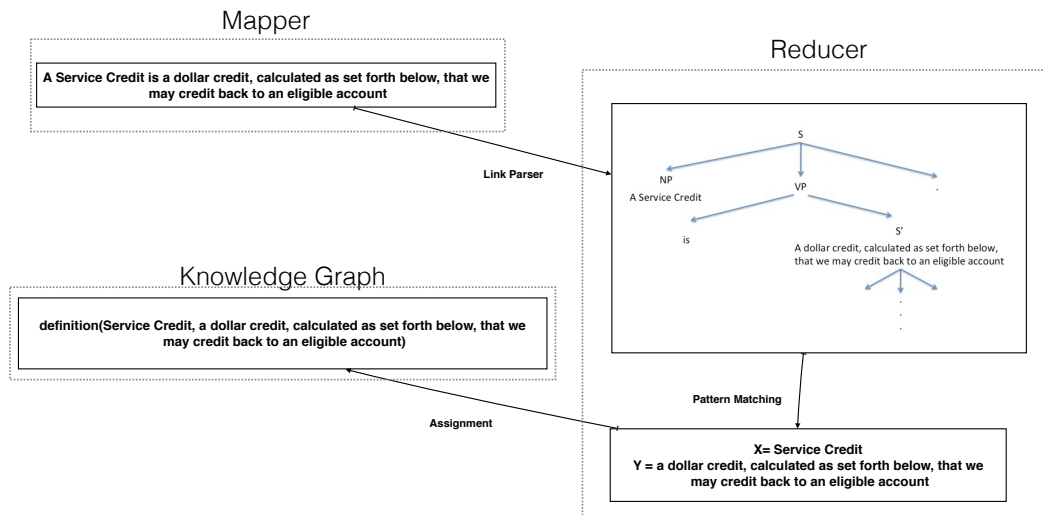
Figure 2. An example iteration where a sentence is converted to a tree structure and then to a RDF statement using Pattern Matching.

future we would like to extend our system to include various other legal documents and agreements.

## REFERENCES

[1] "Resource description framework (rdf)." [Online]. Available: http://www.w3.org/RDF/

[2] "Sparql protocol and rdf query language 1.1 overview." [Online]. Available: http://www.w3.org/TR/sparql11-overview/

[3] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," in *Proceedings of the 10th International Multiconference" Information Society-IS*, 2007, pp. 8–12.

[4] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[5] F. Ciravegna, "2, an adaptive algorithm for information extraction from web-related texts," in *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*. Citeseer, 2001.

[6] P. Cimiano, S. Staab, and J. Tane, "Automatic acquisition of taxonomies from text: Fca meets nlp," in *Proceedings of the International Workshop & Tutorial on Adaptive Text Extraction and Mining held in conjunction with the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2003.

[7] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik, "Deepdive: Web-scale knowledge-base construction using statistical learning and inference." 2012.

[8] F. Niu, C. Zhang, C. Ré, and J. Shavlik, "Elementary: Large-scale knowledge-base construction via machine learning and statistical inference," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 8, no. 3, pp. 42–73, 2012.

[9] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[10] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[11] "The stanford parser: A statistical parser." [Online]. Available: http://nlp.stanford.edu/software/lex-parser.shtml

[12] "Carnegie mellon university link grammar." [Online]. Available: http://www.link.cs.cmu.edu/link/

[13] "Drafting legal documents." [Online]. Available: http://www.archives.gov/federal-register/write/legal-docs/definitions.html

[14] R. Sipoš, D. Mladenić, M. Grobelnik, and J. Brank, "Modeling common real-word relations using triples extracted from n-grams," in *The Semantic Web*. Springer, 2009, pp. 16–30.

[15] K. Barker and N. Cornacchia, "Using noun phrase heads to extract document keyphrases," in *Advances in Artificial Intelligence*. Springer, 2000, pp. 40–52.

[16] K. Joshi and T. Finin, "Ontology for cloud services sla." [Online]. Available: http://ebiquity.umbc.edu/resource/html/id/344/Ontology-for-Cloud-Services-SLA-Service-Level-Agreement