IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-03, Issue-2 E-ISSN: 2347-2693

# Fast and Effective System for Name Entity Recognition on Big Data

**Jigyasa Nigam[1*] and Sandeep Sahu[2]**

Department of CS, SRIT, Jabalpur, RGPV University
jigyasa_092006@yahoo.co.in, sandeep.sahu12@gmail.com

www.ijcaonline.org

Received: Jan /09/2014 Revised: Jan/08/2014 Accepted: Jan/20/2014 Published: Jan/31/ 2014

**Abstract**— In today scenario all data store in digital form and data size is too large. So problem is that how to manage this big data or extract information with speed and efficiency. Information extraction is a technique which using in text mining. Information extraction extract required information whose user demand from unstructured text. Information extraction use NLP (Natural Language Processing) and NER (Name entity recognition). NER systems help to machine recognize proper noun (entity), events, relationships and so on. There are several NER systems in the world. Such as GATE, CRFClassifier, OpenNLP and Stanford NLP (Natural Language Processing ). The NER system works fast for limited amount of documents but drawback of this system is that it works slows for huge/large amount of data. To overcome the drawback of NER system, this paper, report the implement of a NER which is based on Map Reduce, a distributed programming model. This improvement helps to achieve the fast extraction and reduce storage cost with better performance.

***Keywords***— Distributed computing, Big textual data, Named Entity Recognition (NER) , Natural Language Processing (NLP), MapReduce, Hadoop and Maxent Tagger.

## 1. INTRODUCTION

Information extraction is the process in which extracting data from Unstructured Data, semi-structured Data and structured Data. Unstructured Data does not have organized any pre-defined manner or model. Structured data have organize in any pre-define manner or model. Generally extract information of human language texts by uses of natural language processing (NLP). [10][8] Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as information extraction. Information Extraction is a technology that is originates from the user's point of view in the current information existing world. Rather than indicating which documents need to be read by a user, it extracts pieces of information that are relevant to the user's needs. Links between the extracted information and the original documents are maintained to allow the user to reference context. Information is in different shapes and sizes. One important form is structured data, where there is a regular and predictable organization of entities and relationships.
In information extraction NER system play a role is more and more important. It easily recognizes such as persons and organizations can be extracted with reliability. But the problem is that when we use huge amount of data then its processing speed vary slow. Its improve time complexity and space complexity also. [4] [10]
So improve to speed and reduce to space proposed work in this paper to apply NER system with Distributed MapReduce framework. Using the MapReduce framework with NER system got fast information extraction and reduced copy with accuracy.
In this paper use one of the NER system Stanford-POStagger(Part of speech tagger) [8] in which Maxent Tagger model to use extract the information in the form of Name Entity recognition.
The paper is organized as follows. Section 2 introduces related work. We report on
the design of the proposed distributed text parsing system in Section 3. Finally, in Section 4, we give the conclusion.

## 2. Related Work:

In this section, introduce the Stanford-POStagger parser[8] , MapReduce programming model[6] and Hadoop[6]. These are used by the proposed system in single system and distributed environments. First, the Stanford parser, proposed by the NLP lab of Stanford University in the 1990s, in proposed system using the maxent tagger model into part-of-speech tagger from Stanford parser.POS tagger is more than faster to other available tagger. And maxent tagger model to more faster and accurate to other existing model. It uses the best tokenize method in which each and every word create a token.thats why its increase accuracy of parsing and give the best result of parsing. And easily extract the NER.
MapReduce is a programming model for use expressing distributed computations on huge amounts of data and an execution framework for large-scale data processing on clusters of produce servers.[5] It was originally developed by Google and built on well-known principles in parallel and distributed processing which was already introduce several decades. MapReduce has since enjoyed pervasive adoption via an open-source implementation called Hadoop, whose development was led by Yahoo (now an Apache project).

ApacheTM Hadoop is an open source framework that supports distributed computing. It came into existence from Google's MapReduce and Google File Systems projects. It is a platform that can be used for intense data applications which are
processed in a distributed environment. [6][11] It follows a Map and Reduce programming paradigm where the division of data is the simple step and this split data is fed into the distributed network for processing. The processed data is then integrated as a whole. Hadoop also provides a defined file system for the organization of processed data like the Hadoop Distributed File System(HDFS).[6][11] The Hadoop framework takes into account the node failures and is automatically handled by it. This makes hadoop really flexible and a versatile platform for data intensive applications. The answer to growing volumes of data that demand fast and effective retrieval of information lies in engendering the principles of data mining over a distributed environment such as Hadoop. This not only reduces the time required for completion of the operation but also reduces the individual system requirements for computation of large volumes of data. Distributed Computing is a technique aimed at solving computational problems mainly by sharing the computation over a network of interconnected systems. Each individual system connected on the network is called a node and the collection of many nodes that form a network is called a cluster.

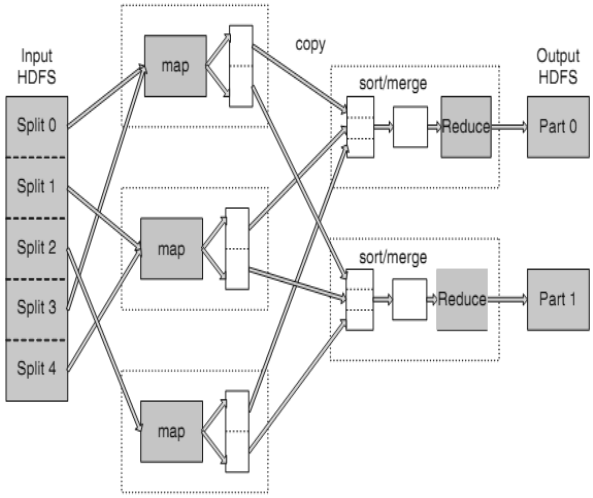



**Fig. 2.1 The MapReduce framework**

In this fig. shows the input data split equally and apply map function on these files.map function make a (key, value) pair to split data. After that they are combine ,and sort and this sorted pair reduce function are apply .reduce function reduce the size of file with maintain the accuracy and give the final MapReduce result.

### 3. Proposed Distributed Parsing System

In order to extract information from huge amount of text file. This paper propose to use of distributed environment in which mapreduce programming apply with StanfordPOS-tagger NLP system.[2][3]
In figure 1.2 show what system propose in this paper. Figure show how the huge amount of input data to access in hadoop distributed file system with the use of mapreduce programming.
Maxent tagger model of Stanford POS-tagger system used by propose system is loaded into hadoop file system because of all mapper function to share it for tokenize the sentences.
In this propose system architecture all input file store in master server and master server distributes
the files to slave servers. After distribution apply the mapper function and Stanford POS-tagger in each file. Mapper function separate the sentences into key and value form and with the help of maxent tagger model of stanfordPOS-tagger system tokenize the sentences [1] And in this tokenized sentences recognize name entity form and apply reduce method. In this paper use maxent tagger model because of this model is tokenize the each and every word of sentence that's why we retrieve the more accurate or fast desire result. Maxent tagger model use the best parsing method to all Existing model [1][9].

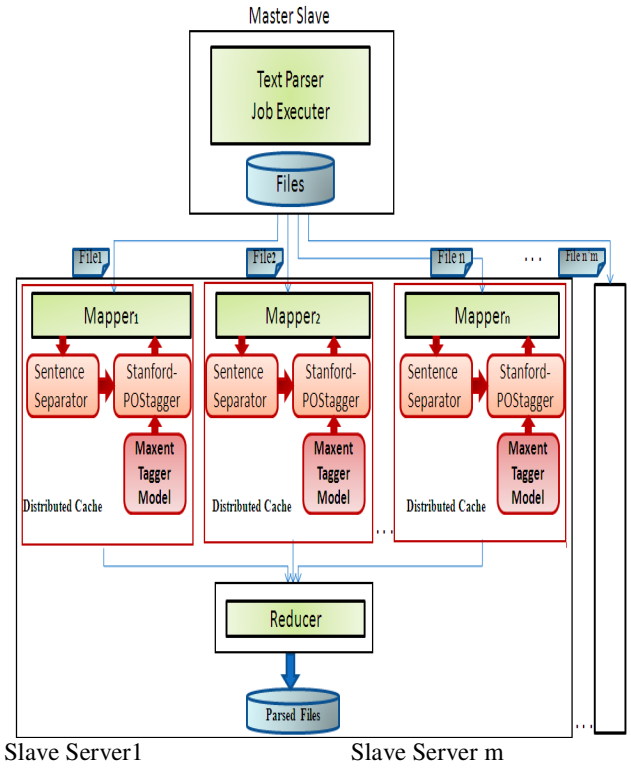
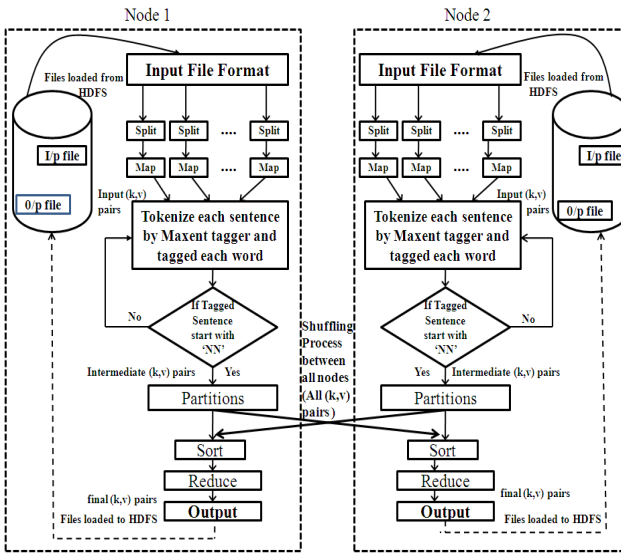



**Fig. 3.1 Proposed work architecture**

Reducer method reduces the result or length of output file. Output file is also store in master server.




**3.2 Flowchart of proposed method**

Show proposed system pseudo code in below

**Class Mapper**

1. StanfordParser.setModel(MaxentTagger model)
2. Method Map(docid a; doc d)
3. sentences = MaxentTagger.tokenizeText(d);
4. sentence1<-sentences.Startwith("NN");
5. Emit(sentence1,d.id)

**Class Reducer**

1. Method Reduce(d.id ,iterable(sentence1)
2. For each sentence1 s1
3. sum=sum+s1;
4. emit(d.id ,sum)

Proposed system have some advantages, first it reduces the time of parsing then to legacy system because in legacy system tokenize file one by one.
Second, it reduces the size of output file through which we can easily recognize how many times, the particular word uses in all files. Because it provides the count with every word [1].
Third we can easily modify it with replacing another parsing method.

## 4. Experimental Result

To calculate the performance of the proposed system, we consider four autonomous computer in these four computers make one of them Master or left are consider as a slave.
In this implementation compare the space complexity or time complexity of the legacy Stanford POS-tagger system with that of the distributed Stanford POS-tagger system applied to the MapReduce programming model in hadoop. We also evaluate the running time of the distributed Stanford POS-tagger system on four slaves. In These all autonomous computers each one consists of eight cores of Intel i3-3220, 3.30 GHz CPU speed and 2GB RAM. In this Implementation use hadoop-1.0.4 and the Stanford Pos-tagger3.30. And Java-oracle-7 programming language used. The data set consists of 1000 file of financial data. In this implementation evaluate the running time from the step of the sentence separation to the step of parsing sentences using the distributed Stanford parser.
In below table shows execution time comparison between legacy system and implemented system with different numbers of nodes. In this table execution time consider in seconds. This graph clearly shows implemented system much faster than legacy system.

| S. I | Nodes Numbers | Execution Time (In seconds) |
|---|---|---|
| 1. | Existing system | 2018 |
| 2. | Single Node MapReduce | 3067 |
| 3. | Two Node MapReduce | 1533.5 |
| 4. | Four Node MapReduce | 766.75 |

**Table 4.1 Execution Comparison time table**

In below graph shows those related to Table 4.1 i.e.
In this graph consider legacy system execution time and implemented system which consider with single node, two nodes and three nodes execution time in seconds.

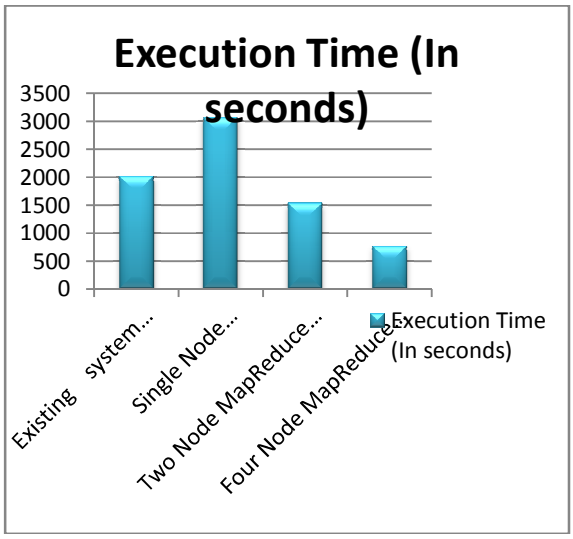



**Fig. 4.1 Execution time Comparison**

In table 5.2 shows result storage cost on legacy system as well as implemented system. In This table consider different category for Compare to result storage cost such as file size, page numbers and words count.
And in every category implemented system storage cost of output file is minimal than existing system. See in table 4.2

| S.No. | Category ⇨ System ⇩ | Documents Size (In KB) | | Page Numbers | Words count |
|---|---|---|---|---|---|
| | | In Text File (.txt) | In Word File (.do | | |
| 1. | Existing System | 191 KB | 183 KB | 519 | 22,604 |
| 2. | Implemented system | 62.5 KB | 46.3 KB | 89 | 10,324 |

**Table 4.2 Storage cost comparison table**

In fig. 4.2 shows output file size comparison in both system, and consider text file as well as word file to store same output. text file taking 191KB and word file storage cost 183KB in legacy system while in implemented system text file taking 62.5KB and word file to store same 46.3KB.

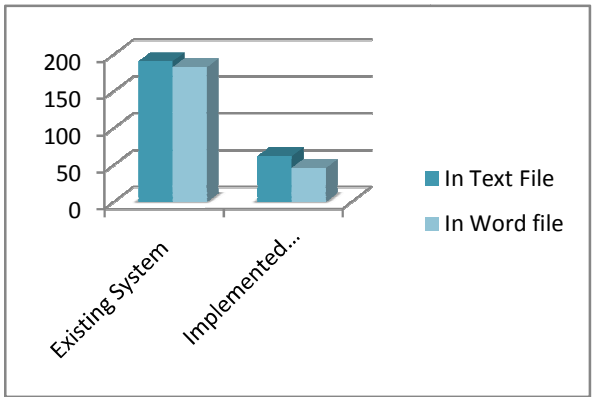



**Fig. 4.2 Output files size comparison**

In fig 4.3 shows page numbers of output files in legacy system and implemented system. Legacy system output store in 519 pages and implemented system store output in 89 pages.

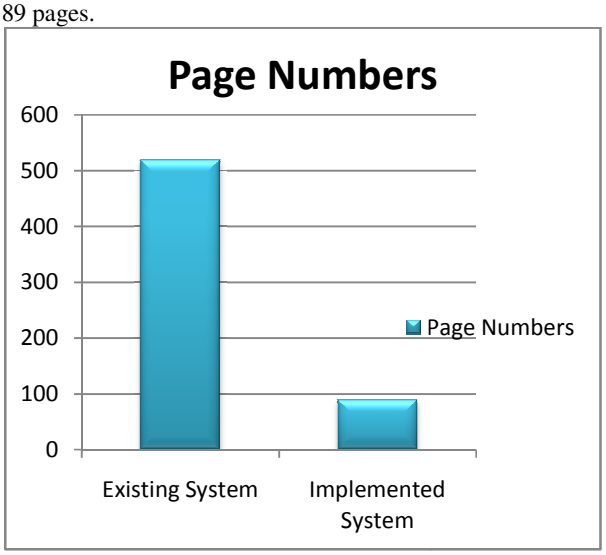



Fig. 4.3 Output file Page wise comparison

In Fig 4.4 shows words count of output file.
In legacy system 22,604 words count of output file and implemented system 10,324 words count of output file.

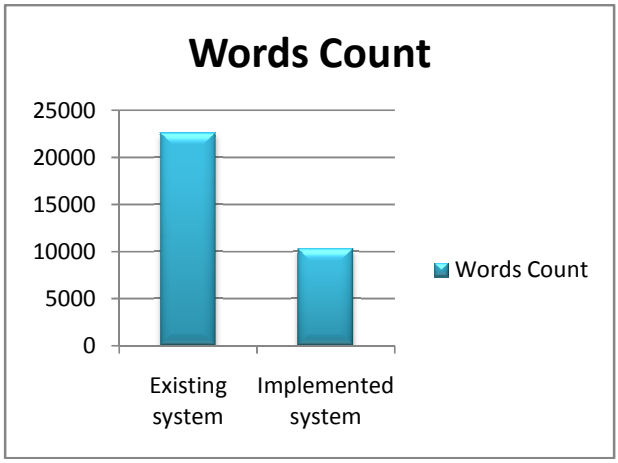



**Fig. 4.4 Output files word wise comparison**

This implemented system reduces storage cost just because it considers one word in one time whenever this word repeat in many time in data set. This system reduces repetition and considers total number occurrence count with particular word. Example word "Attention 4". In legacy system word "Attention" occur in four times in different places although in implemented system show this word like this "Attention 4".and implemented system arrange output in alphabetical order.

## 5. Conclusion:

This paper proposes the approach for processing of huge amount of text in distributed environments with MapReduce programming in which Stanford POS-tagger parser applies for name entity recognition. Advantage of propose system, it is less time consuming then to legacy system as well as reduce storage cost and arrange data in alphabetical order.
For future work this System evaluates for show the relationship between name entity and in addition optimized technique for parsing in distributed environment.

## 6. References:

[1]. Nigam, Jigyasa, and Sandeep Sahu. "An Effective Text Processing Approach With MapReduce."
[2]. James J. (Jong Hyuk) Park et al. (eds.), Mobile, Ubiquitous, and Intelligent Computing,Lecture Notes in Electrical Engineering 274,DOI: 10.1007/978-3-642-40675-1_41, © Springer-Verlag Berlin Heidelberg 2014
[3]. Kim, J., Lee, S., Jeong, D.-H., Jung, H.: Semantic Data Model and Service for Supporting Intelligent Legislation Establishment. In: The 2nd Joint

International Semantic Technology Conference (2012)

[4]. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423–430 (2003)

[5]. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: OSDI, pp. 137–150 (2004)

[6]. HDFS (hadoop distributed file system) architecture(2009),http://hadoop.apache.org/common/docs/current/hdfs-design.html

[7]. Seo, D., Hwang, M.-N., Shin, S., Choi, S.: Development of Crawler System Gathering Web Document on Science and Technology. In: The 2nd Joint International SemanticTechnology Conference (2012) Morphological features help POS tagging of unknown words across language varieties

[8]. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 425–432,Sydney, July 2006. c2006 Association for Computational Linguistics

[9]. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Databases (VLDB-94), pages 487–499, Santiago, Chile, Sept. 1994.

[10]. en.wikipedia.org/wiki/Information_extraction

[11]. Shvachko,K. Yahoo!,Sunnyvale,CA,USA Hairong Kuang ; Radia, S. ; Chansler, R.Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on E-ISBN **:**978-1-4244-7153-9