Comparative Study of POS Taggers

Aastha Gupta
Bhagwan Parshuram Institute of Technology
GGSIPU, Delhi

Rachna Rajput
Bhagwan Parshuram Institute of Technology
GGSIPU, Delhi

Richa Gupta
Bhagwan Parshuram Institute of Technology
GGSIPU, Delhi

Monika Arora
Bhagwan Parshuram Institute of Technology
GGSIPU, Delhi

ABSTRACT
POS Tagging provides important grammatical as well as contextual information for each word in the corpus. POS Tagging enables various companies to be able to track user reviews and can even be used for Speech Synthesis. In this paper, different POS Tagging Algorithms, namely, Memory-Based Learning Algorithm, Multi-Domain Web Based Algorithm and the Hybrid Model, will be compared on the basis of their execution time as well as efficiency. In Memory-Based Learning algorithm, the word to be tagged is searched in the lexicon using weighted similarity matrix, if an exact match is found, its lexical representation is retrieved, but, if it is not found, the lexical representation of its nearest neighbor is retrieved. Thus, the algorithm will not work efficiently for sparse data. On the other hand, Multi-Domain Web Based Algorithm is used to tag unknown words. The word is searched over the web for its possible tags. Due to the web search, runtime overhead is induced for each word. The tag with highest occurring probability is assigned to the word. The Hybrid Model executes Memory-Based Learning algorithm for known words and Multi-Domain Web Based Algorithm for unknown words.

Keywords
POS Tagging, Multi-Domain Web based Algorithm, Memory Based Learning Algorithm, Hybrid Model

1. INTRODUCTION
With the advent as well as the popularization of the internet, large amount of data is present on review sites, blogs, forums, social networking sites such as Facebook, Twitter, etc. All this data is in English language and will continue to be a major medium [16] for communication, knowledge accumulation and information distribution. Because of increasing competition, users require information extraction from the text documents, quickly but at a low cost. While processing texts written in natural language, the most critical problem is ambiguity and uncertainty issues. To deal with this effectively, the processing includes an important task called POS Tagging or Parts-of-speech Tagging. POS Tagging is the assigning of a word with its syntactic role in the sentence. Parts-of-speech may include Noun, Verb, adjective, Pronoun, Adverb and various other categories. For Example:

Unlabeled Text

When he handed in his homework, he forgot to give the teacher the last page.

Labeled Text

When/WRB he/PRP handed/VBD in/IN his/PRP$ homework/NN ,/, he/PRP forgot/VBD to/TO give/VB the/DT teacher/NN the/DT last/JJ page/NN./.

Words are often ambiguous in their behavior depending on their usage in the sentence. For Example,

The word 'run' can be used as

A Verb: Hurry! Run for it.

A Noun: She usually goes for a run before breakfast.

POS taggers are broadly classified into two categories, namely, rule based and Stochastic based taggers. The rule based POS taggers use contextual information and apply a set of hand written rules to find an appropriate tag for the word. For instance, Contextual rules may say something like " *Eliminate VBN if VBD is an option when the sentence starts with PRP"*.

The Stochastic based POS taggers use its past experiences to handle the new situation, i.e. from the previously tagged data, it finds out, most frequently used tags for a specific word and on the basis of this information, the tagger finds an appropriate tag. For each word, he tagger calculates frequency and probability of occurrence for each word in the corpus. Stochastic based POS tagger can be further divided into two broad categories, they are, Supervised POS Taggers and Unsupervised POS Taggers. Supervised POS taggers [4] require a pre-tagged corpus, i.e. it requires information about the tag set, word-tag frequencies, rule sets etc. to perform POS Tagging. The performance of this tagger depends on the corpus size, it increases with the increase in size of the corpus. Unsupervised POS Tagger automatically induces the tag sets and the transformational rules, using Baum-Welch algorithm. On the basis of this information, they either compute the probabilistic information or the contextual rules.

Parts-of-speech tells us how the word is used in the context of the corpus. The word '*content*', for example, can be a noun or an adjective. Thus, knowing the part-of-speech can aid well speech synthesis system. Parts-of-speech can also be used for informational retrieval (IR), They can also enhance an IR application by selecting out nouns or other important words from a document. To do this efficiently, various POS Tagging algorithms have been designed. This paper focusses on discussing and comparing two widely used POS Tagging algorithms, Memory-Based learning and Multi-Domain Web Based Algorithm, and the Hybrid Model.

2. MEMORY-BASED LEARNING ALGORITHM
Memory-Based Learning (MBL) is a supervised learning algorithm based on its classification. A Memory-Based Learning algorithm has two main components: A learning component, which does memory based classification, i.e. words with its exemplified usage into the lexicon as the training set, and a performance component, which does similarity-based classification, i.e. The performance of the algorithm can be evaluated on the basis of how efficiently the similarity matrix will be able to map the new situation to the earlier experiences.

Over other traditional POS taggers, Memory-Based Learning algorithm has a number of advantages. First, While computing similarity metrics, the weights for sparse data, i.e. the data for which the lexicon lacks any similar examples, are set as ∞. Therefore, does not require any additional smoothing. Second, once exceptional or rare patterns are stored in the training set, can contribute to generalization. Third, with the help of weighted similarity metrics, this approach provides an effortless integration of various sources of information.

During the process of tagging, when a word is to be tagged, a weighted similarity matrix is formed i.e. weights are applied to the neighbors on the basis of its similarity to the word to be tagged. If there is an exact match, then a weight of 'Zero' is denoted in the similarity matrix. Least weights are applied to the nearest neighbor. For the exact match, lexical representation is retrieved and the appropriate tag is determined. On the other hand, if the word is not found in the lexicon, the lexical representation of the neighbor, having the least weight, is retrieved and the appropriate tag is determined. The output [1] is a best guess of the category for the word in its current context.

Memory-Based Learning has two variants. First, in IBI-IG [1], during learning, a database of instance is built. For each match, the algorithm calculates the distance between the new instance A and the memory instance B. When both instances are equal, the distance is zero and one otherwise. Second, IGTREE [1] uses compressed decision tree structure and contains the same information as IBI-IG. The search is restricted from the feature with the same weight to the feature with the highest weight.

The researchers devised other algorithms to overcome various limitations of Memory-Based Learning. Some of the limitations are: First, It is practically impossible to prepare such a large training set. Second, for sparse data, the algorithm proved to be inefficient. Third, computing similarity matrix for every word, takes a larger amount of time than searching it over the web.

3. MULTI-DOMAIN WEB BASED ALGORITHM

Multi-Domain Web Based Algorithm is applied to find tag for an unknown word, i.e. the word whose tag cannot be computed with data present in the lexicon. The algorithm follows supervised POS tagging approach, meaning, it requires a pre-tagged corpus for POS Tagging. It does not require any preprocessing of the corpus, but considers 'Multi Domain', i.e. not only the domain of the sentence; it takes into account, all possible domains of the word. As the name suggests, it is a 'web based algorithm', the tags for the unknown word are searched over the web.

Using Multi-Domain Web Based Algorithm, when a word is to be tagged, the algorithm executes a web query on the web server. The web server, then, retrieves its usage in all possible domains along with the frequency of occurrence of each domain. On the basis of the results retrieved by the web server, the algorithm will be able to compute the probability of occurrence of each domain, according to the context of the sentence. The tag or the domain having the highest probability of occurrence will be assigned to the word under consideration. Every time the algorithm executes, it creates a connection with the web server. The time lost in the transfer of control from the application to the web server, connecting to the web server and transfer of control from the web server to the application, creates a runtime overhead of 0.5 seconds. This runtime overhead adds substantially to the total execution time.

The two common features in a web query, supported by most of the popular search engines, they are, wild-card search, denoted using the '*' character (Search Engine retrieves an alternative for '*'), and exact sentence search, expressed by quoting characters (Search Engine searches for '*' based on the given context). The retrieved sentences contain the parts enclosed in quotes in the exact same place they appear in the query, while an asterisk can be replaced by any single word. For each unknown word 'U', the algorithm executes three queries, to retrieve all possible domains [5].

1. **Replacement**: "$U_{I-2}U_{I-1}*U_{I+1}U_{I+2}$". This retrieves words that appear in the same context as U_I.
2. **Left-side**: "$**U_IU_{I+1}U_{I+2}$". This retrieves alternative left-side contexts for the word U and its original right-side context.
3. **Right-side**: "$U_{I-2}U_{I-1}U_I**$". This retrieves alternative right-side contexts for U and its original left-side context.

Final Tagging is based on the conditional probability of the tag t_r in various domains[5]

$$p(t_r|h) = \frac{p(h, t_r)}{\sum t_r' \epsilon T \; p(h, t_r')}$$

Where, T is the tag set, and p(h) is the 'history' set for the given context.

Although, the algorithm tags unknown words accurately, but lacks efficiency. Since, the algorithm [2], has a runtime overhead of 0.5 seconds per unknown word, even if the word was previously searched using the algorithm. To deal with this and the problem of sparse data in Memory-Based Learning Algorithm, The Hybrid Model has been developed.

4. THE HYBRID MODEL

The Proposed model efficiently integrates the required features of Memory-Based Learning Algorithm and Multi-Domain Web Based Algorithm. When, Unstructured data is fed as input to the POS Tagger, it performs a series of steps to find the POS tags of the given input. The first step is tokenization, i.e. each word is separated in the corpus, so that, they can be processed individually. The second step is the word selection, i.e. each word is sequentially selected for tagging. In the third step, the selected word is searched in the lexicon. If an exact match is found, Memory-Based Learning Algorithm executes, but if not, Multi-Domain Web Based Algorithm executes. The algorithm repeats recursively, till the tagger encounters the last word in the corpus. At the end, results, i.e. words along with its tags, are displayed.

If the word is not found in the lexicon, i.e. it is an 'unknown word', as explained above the word is searched over the web and collects all possible tags. The tagger, then computes probability of occurrence of various collected tags, going by the frequency count of the occurrence. The tag with maximum probability is assigned. Along with temporarily storing the final result into the memory using linked list structure, it also stores the results of web query into the lexicon.

If the word is found in the lexicon, i.e. it is a 'Known Word', the lexical representation of the neighbor having the least weight in the similarity matrix is retrieved and POS tagging rules are applied to disambiguate various possible POS tags. The final result is then temporarily stored into the memory using linked list structure (to be displayed on the screen).

By saving the results of Multi-Domain Web Based Algorithm into the lexicon, this model not only improves the overall execution time of the Multi-Domain Web Based Algorithm, but also, improves the efficiency of Memory-Based Learning Algorithm. Since, the results of web query are stored into the lexicon, the word will be treated as a 'Known Word', if it appears again in the corpus. There will be no web connection overhead. Also, after every execution, the training set of the algorithms becomes more efficient.

This model exhibits a tradeoff between space and time complexity, i.e. the time utilized in the computation can be reduced at the cost of increased memory in usage. To manage the disk space efficiently, two different concepts, i.e. Least Recently Used (LRU) Page Replacement Technique in combination with Least Frequently Used (LFU) Page Replacement Technique is implemented in this model. According to these techniques, Whenever the assigned storage space [1] gets full and a new word is to be stored into the lexicon, the record of the word will be removed from the lexicon whose occurrence frequency is low with a high occurrence timestamp period (in comparison with others).

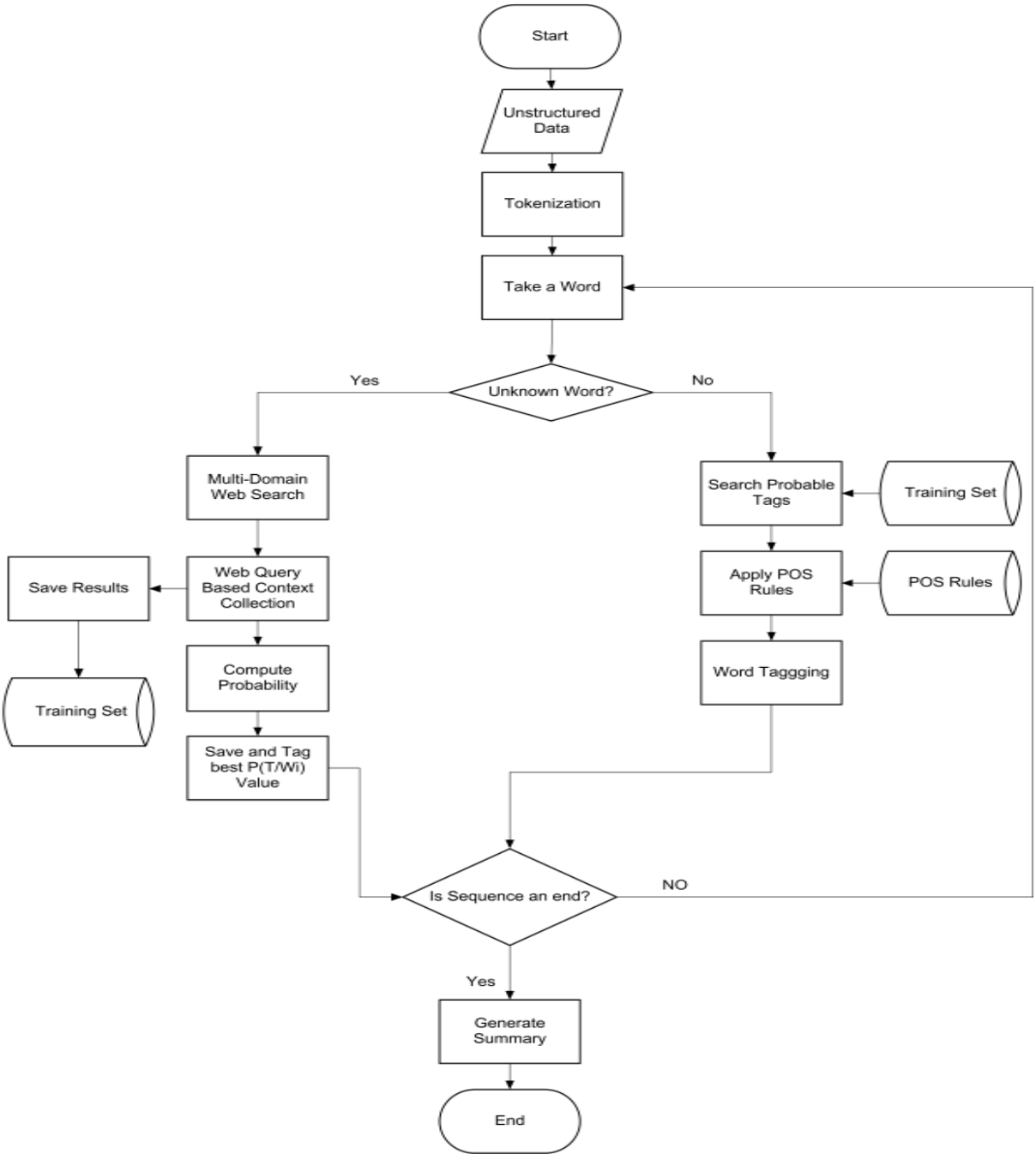

Figure 1: Flow Chart of the Hybrid Model [1]

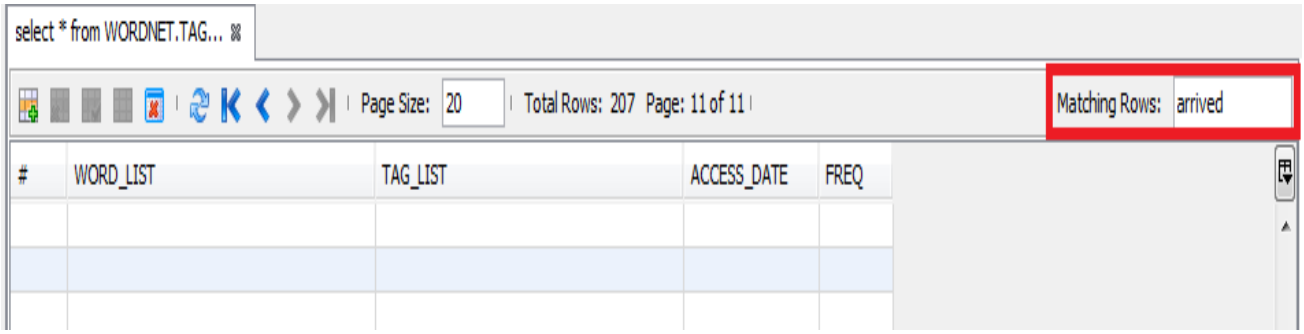

Figure 2: The word 'arrived' in the lexicon is missing before execution of the Hybrid Model

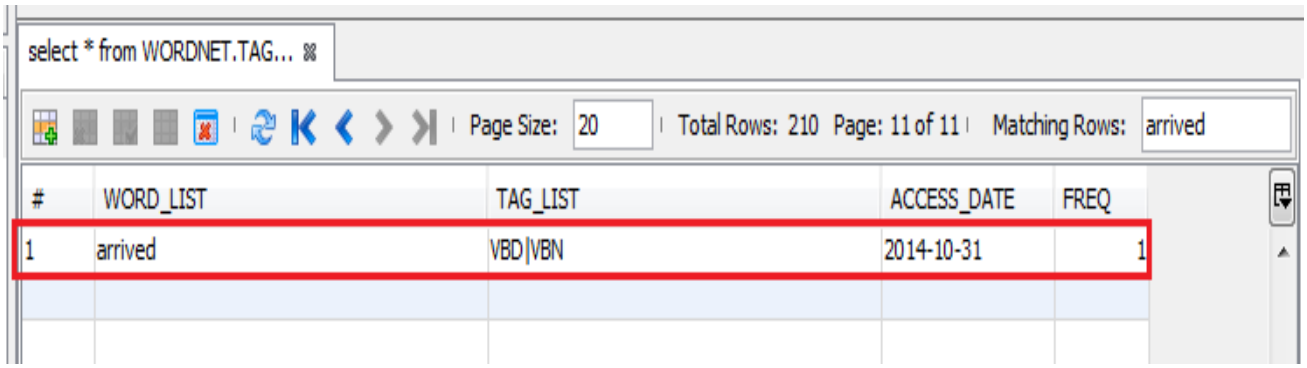

Figure 3: The word 'arrived' is stored into the lexicon after the execution of the Hybrid Model

5. RESULT AND DISCUSSION

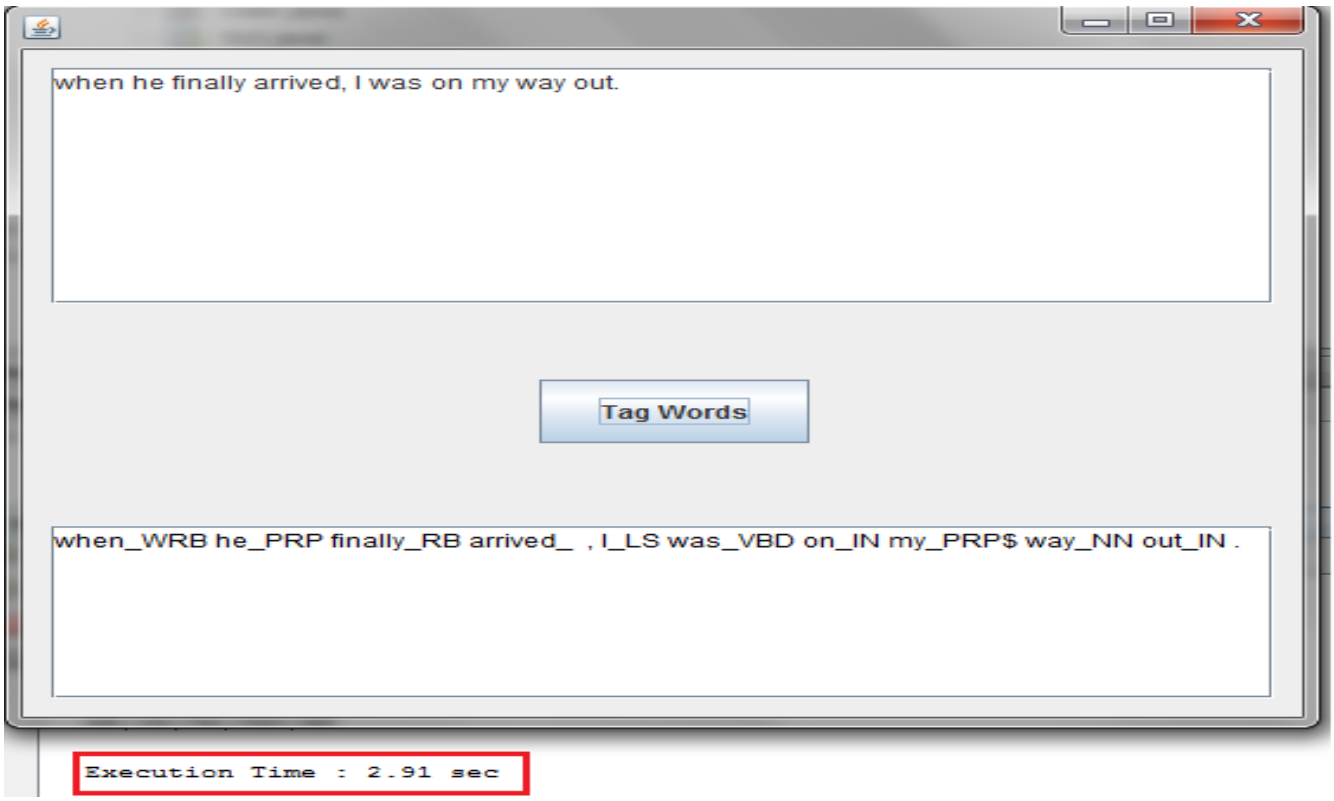

Figure 4: Execution Time for Memory-Based Learning Algorithm

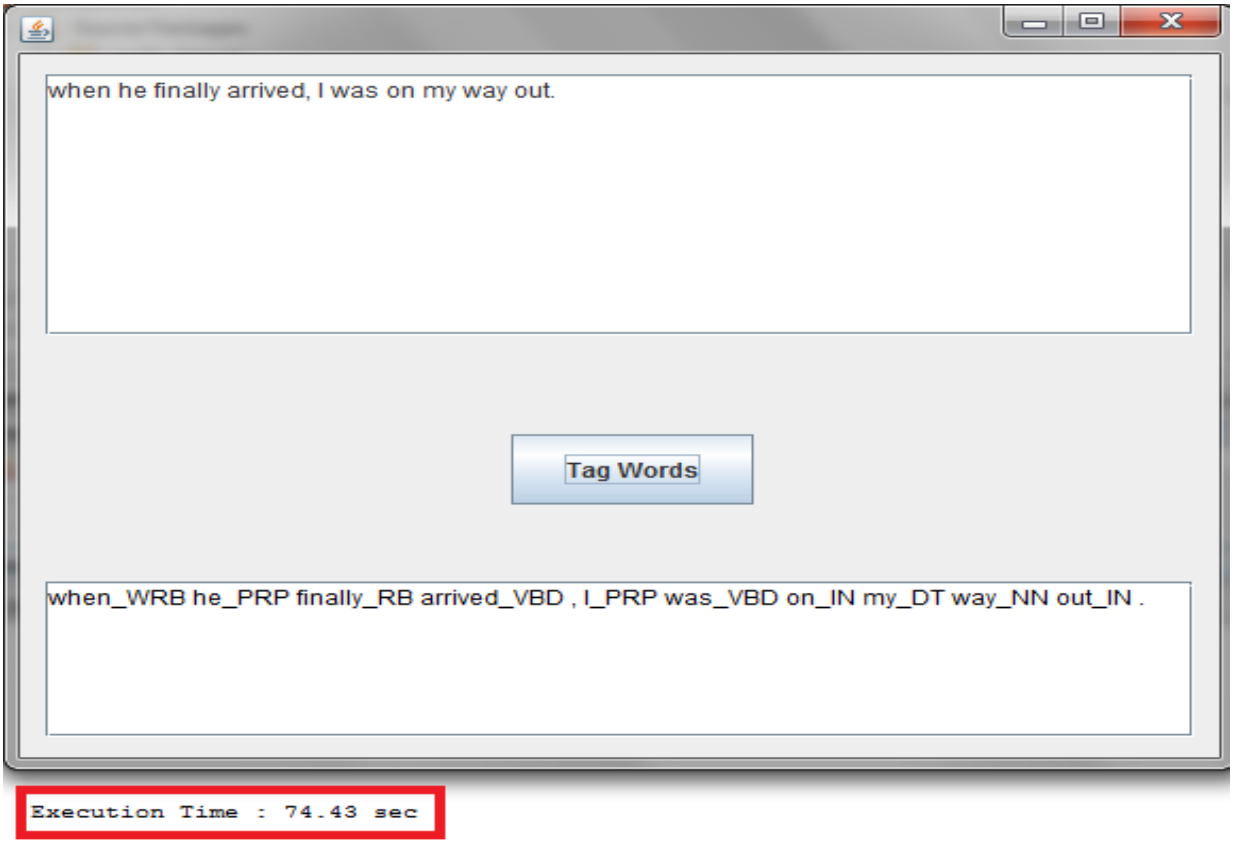

Figure 5: Execution Time for Multi-Domain Web Based Learning Algorithm

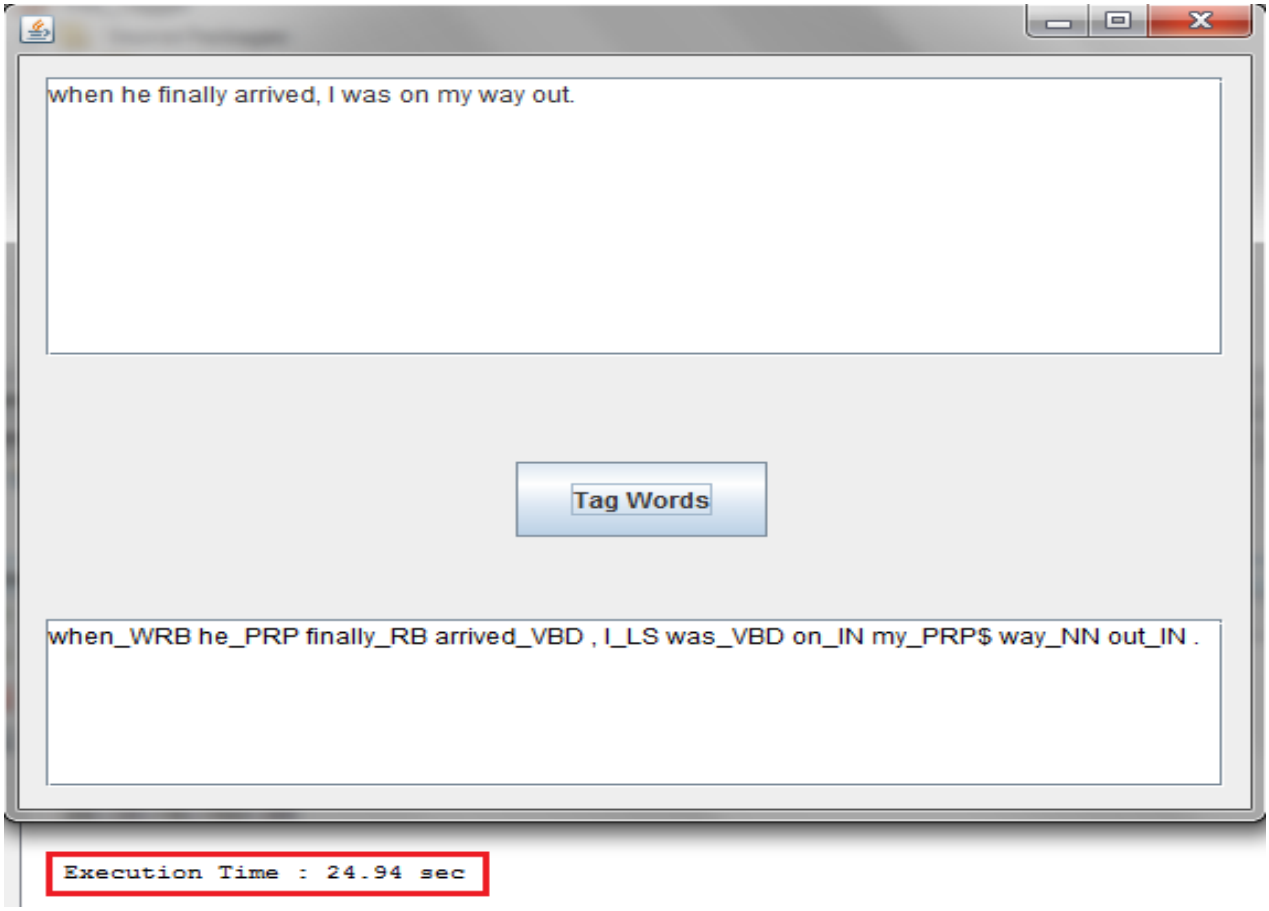

Figure 6: Execution Time for the Hybrid Model (during first execution of the text)

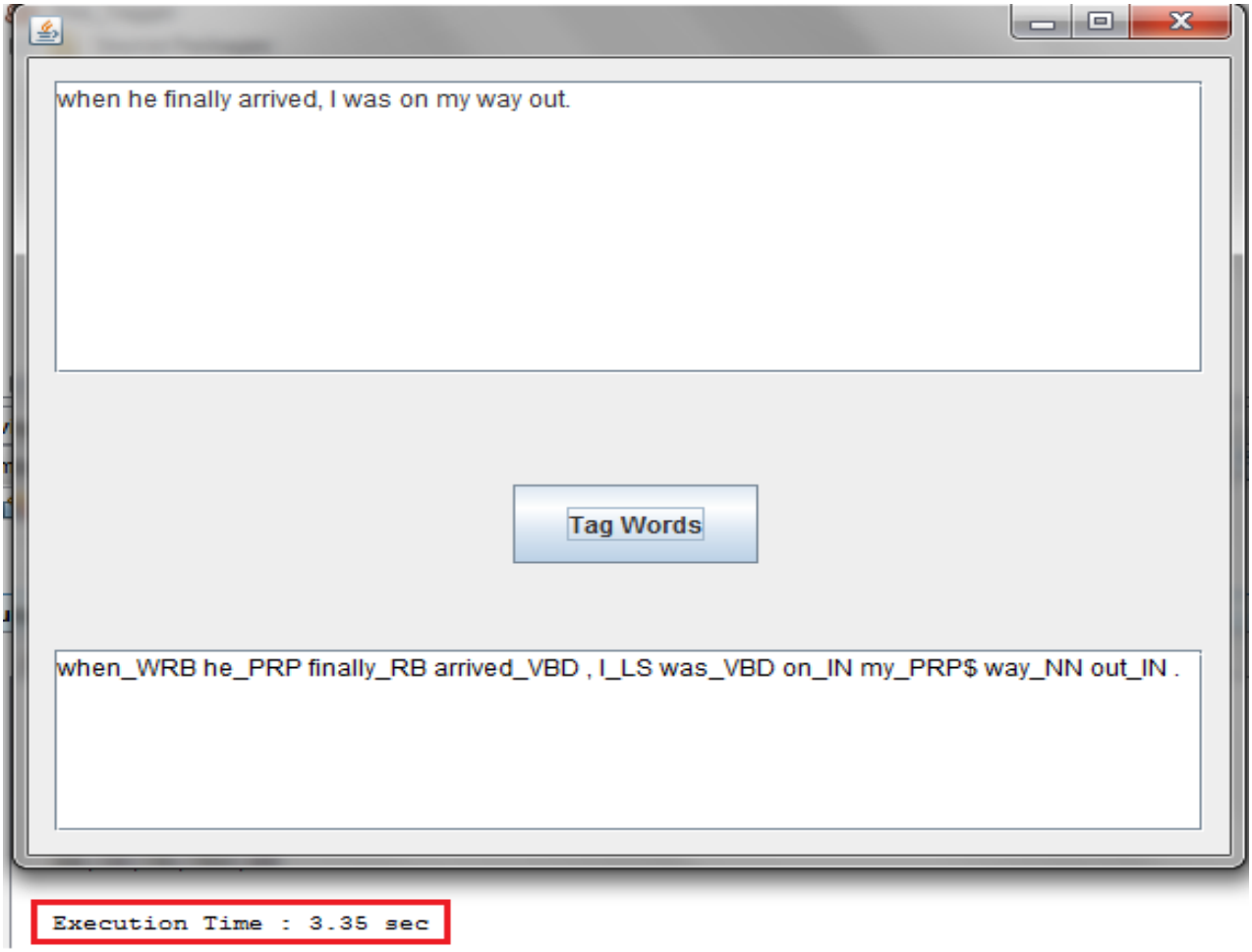

Figure 7: Execution Time for the Hybrid Model (during second execution of the text)

Table 1: Execution Time for various algorithms to tag a sentence having 10 words

Execution time for Memory-Based Learning Algorithm (sec)	Execution time for Multi-Domain Web Based Algorithm (sec)	Execution time for the Hybrid Model when 'arrived' was unknown (sec)	Execution time for the Hybrid Model when all words are known (sec)
2.91	74.43	24.94	3.35

As shown in the results, Memory-Based Learning algorithm failed to find an appropriate tag for the word 'arrived', which was not stored in the lexicon. Thus, the algorithm is not effective while tagging sparse data. The results also show that, running Multi-Domain Web Based Algorithm can be one of the solution for the sparse data problem. But running this algorithm, solely, can be hazardous for the execution time, since there is a runtime overhead of 0.5 seconds per word. Therefore, running a hybrid can deal with both the problem, effectively and efficiently. A corpus may contribute to the worst case in the first execution, i.e. most words are searched over the web, but, will be the best case in the second and future execution for the same corpus i.e. all the words will be found in the lexicon.

6. CONCLUSION

POS Tagging plays a vital role in extracting information from the data. Considering its importance, researchers have devised various algorithms to improve its efficiency. The paper intentions are delivering a comprehensive discussion and comparison between Memory-Based Learning Algorithm, Multi-Domain Web Based Algorithm and The Hybrid Model. Memory-Based Learning Algorithm, although, being efficient in finding POS tags for the words whose exemplified usage is stored in the lexicon, lacks ability to tag sparse data. Multi-Domain Web Based Algorithm requests for web connection, every time it executes, thus creates a runtime overhead of 0.5 seconds. The execution time increases significantly, if the corpus size is large enough. The Hybrid Model is a combination of both the above mentioned algorithms, thus, it is able to deal with the limitations, effectively. Adding the results of web query to the lexicon, eliminates runtime overhead for the word, whenever it appears in corpus next time. Also, Multi-Domain Web Based Algorithm is apt in finding the POS tag for unknown words, thus, is efficient in sparse data problem.

7. REFERENCES

[1] Aastha Gupta, Rachna Rajput, Richa Gupta and Monika Arora "Improved POS Tagging for Unknown Words ", International Journal of Soft Computing and Engineering, ISSN:2231-2307 Vol. 4,Issue-ICCIN-2k14, March 2014.

[2] Aastha Gupta, Rachna Rajput, Richa Gupta & Monika Arora, 2014,'Hybrid Model to Improve Time Complexity of Words Search in POS Tagging'.Paper presented at

International Conference on Data Mining and Intelligent Computing,IEEE, Delhi, India

[3] Amit S. Chavan, Kartik R. Nayak, Keval D. Vora, Manish D. Purohit and Pramila M. Chawan, "A Comparison of Page Replacement Algorithms", ACSIT International Journal of Engineering and Technology, Vol.3, No.2, April 2011

[4] Antony P J and Dr. Soman K P, "Parts Of Speech Tagging for Indian Languages: A Literature Survey", International Journal of Computer Applications (0975 – 8887) Vol.3, No.8, November 2011.

[5] Ari Rappoport, Roi Reichart and Shulamit Umansky-Pesin, "A Multi DomainWeb-Based Algorithm for POS Tagging of Unknown Words", Coling 2010: Poster Volume, pages 1274–1282,Beijing, August 2010

[6] Debnath Bhattacharyya, Susmita Biswas and Tai-hoon Kim, "A Review on Natural Language Processing in Opinion Mining", International Journal of Smart Home Vol.4, No.2, April, 2010.

[7] Erik Cambria, Robert Speer, Catherine Havasi and Amir Hussain," SenticNet: A Publicly Available Semantic Resource for Opinion Mining", Commonsense Knowledge: Papers from the AAAI Fall Symposium (FS-10-02).

[8] Guido Minnen, Francis Bond and Ann Copestake, "Memory-Based Learning for Article Generation", in Proceedings Of CoNLL-2000 and LLL-2000, pages 43-48, Lisbon, Portugal, 2000.

[9] Hejab M. Alfawareh and Shaidah Jusoh, "Resolving Ambiguous Entity through Context Knowledge and Fuzzy Approach", International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397, Vol. 3 No. 1 Jan 2011

[10] Jakub Zavrel & walter Daelemans,"Recent Advances in Memory Based Part of Speech Tagging.", VI Simposio Internacional de Comunicacion Social, Santiago de Cuba pp. 590-597, 1999

[11] Lars Bungum, Bjorn Gamback, "Evolutionary Algorithms in Natural Language Processing", Norwegian Artificial Intelligence Symposium, Gjøvik, 22 November 2010

[12] Mahesh T R, Suresh M B, M Vinayababu. "Text Mining: Advancements, Challenges And Future directions", International Journal of Reviews in Computing, ISSN: 2076-332, © 2009-2010 IJRIC& LLS.

[13] Omae Al-Harbi, Shaidah Jusoh and Norita Md Norwawi, "Lexical Disambiguation in Natural Language Questions (NLQs), IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011 ISSN (Online): 1694-0814

[14] Parag Bhalchandra, Nilesh Deshmukh, Sakharam Lokhande, and Santosh Phulari, "A Comprehensive Note on Complexity Issues in Sorting Algorithms", Advances in Computational Research, ISSN: 0975–3273, Volume 1, Issue 2, 2009, pp-1-09

[15] Parmar Mitixa R, Prof.Arpit Rana, "A Survey on Opinion and Sentiment Analysis With Applications and Issues", International Journal of Computational Linguistics and Natural Language Processing ISSN 2279 – 0756, Vol. 2, Issue 1, January 2013

[16] Shaidah Jusoh and Hejab M. Alfawareh, "Techniques, Applications and Challenging Issue in Text Mining", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012 ISSN: 1694-0814

[17] Shaidah Jusoh and Hejab M. Al Fawareh, "Semantic Extraction from Texts", 2009 International Conference on Computer Engineering and Applications IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

[18] Yin Shaohong and Fan Guidan, "Research of POS Tagging Rules Mining Algorithm", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).