IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. VIII (Nov – Dec. 2014), PP 61-68
www.iosrjournals.org

Enhanced Model to Improve Memory Based Learning Algorithm

Aastha Gupta
Computer Science and Engineering, Bhagwan Parshuram Institute of Technology/ IP University, India

Abstract: *Opinion mining/Sentiment analysis is a field of research which focuses on tracking human opinions written in natural language. Many companies, Now-a-days, extract opinions from various internet sources (such as review sites, blogs, twitter) and try to predict customer's reviews on their upcoming products. Parts-Of-Speech Tagging or POS Tagging, being a crucial step in the process of opinion mining, is the assigning of each word with its POS tag, according to its definition and context in the corpus. In this paper, we will first have detailed study of a POS tagging algorithm i.e. Memory Based Learning Algorithm, along with other POS tagging techniques and then, dealing with its limitations, we will propose an enhanced model in order to significantly improve the efficiency of Memory-Based Learning Algorithm. The Memory Based Learning Algorithm, maintains a word-tag-feature repository, storing word with all its possible tags, along with its fixed width context called feature. During tagging, the algorithm retrieves/computes an appropriate tag, deduced by analyzing feature value of the word. Thus, this algorithm is ineffective, when dealing with sparse data. To be able to deal with this limitation, we have devised an enhanced Memory Based Learning model. In addition to this, we have compared the proposed model with other POS tagging algorithm, in terms of its performance.*

Keywords: *Memory-Based Learning Algorithm, Multi Domain Web Based Algorithm, Maximum Entropy POS Tagging Model , Opinion Mining, POS Tagging*

I. Introduction

Opinion Mining or Sentiment Analysis is a type of natural language processing which aims at designing an automated system, enabling users to extract human's opinion from the corpus written in natural language (complex language). 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., companies use various opinion mining techniques which can aid them in strategic decision making. Considering a case when an organization tracks, reviews about their products and services, use them in their strategic planning. Not only organizations but users, who plans to buy a product, to have a better idea, he/she looks over the web, searches for various brands, compare product reviews and based on their analysis, takes a decision.

Based on user requirements, opinion mining can be done at three levels namely, at the sentence level, at the document level and at the feature level. At the sentence level, the sentence is classified as being subjective, i.e. an opinion, or being objective i.e. a fact. Going by the behavior of a subjective sentence, it can be further classified as positive or negative. At the document level, assuming that every document contains the opinions/reviews of a single user/reviewer and focuses on a single objective, a document can be classified on the basis of overall sentiments expressed by the user/reviewer. At the feature level, opinions are classified according to different features of the product. Let us follow this with an example [1]:

The review of a Motorola Xt875 mobile user " I bought the phone a few years back and it was worth buying. It has a very large touch screen having an excellent voice quality. Although, the battery life is not competitive enough, I was really impressed with other features". In reviews like this, during review analysis, particular product features are extracted.

Various challenges are faced while mining an opinion. First, since English is a dynamic language, the inclination of an opinionated word may change according to its occurrence in various situations. Consider the two cases:

Case 1: The news is good and true.

Case 2: The news is too good to be true.

Here, the word 'good' is indicating an opinion in the sentence. In the first case, it is having a positive inclination, i.e. it is providing a positive tone to the sentence. Whereas in the second case, it is having a negative inclination, i.e. it is providing a negative tone to the sentence

Second, users use different sentence structures to express their opinions (similar in meaning). They may use complex sentence structures to express a simple opinion while reviewing. For example [2]:

Consider a review "Perhaps this be a great innovation, but I fail to see why". The reviewer used contrary words, such as 'great' and 'fail', together in his/her opinion. The review is written in natural language and can be easily understood by the human mind, but is a bit confusing for the machine to interpret, which lacks real intelligence.

The reviews, written in natural language, need to be processed to determine its opinion. The processing initiates with a critical task called Parts-of-speech Tagging or POS Tagging. POS Tagging is the assignment of each word with its morphological meaning in the sentence. Various elemental Parts-of-speech classes may include Adverb, Adjective, Determiner, Noun, Pronoun, Verb, and a lot more.

For Example: Unlabeled Text
You should come jam with us sometime.
Labeled Text
You/PRP should/MD come/VB jam/VB with/IN us/PRP sometime/RB.

Depending on how a word is used in the sentence, it may behave differently from its prior occurrences. For Instance, The word 'jam' can behave as
A Verb: You should come, jam with us sometime.
OR
A Noun: It caused a jam in the printer.

POS Taggers can be categorically divided into two POS tagging approaches [3], namely, supervised POS tagging approach and unsupervised POS tagging approach. Supervised POS tagging approach requires information about word-tag frequencies, tag set, rule set etc., for POS tagging. As the size of the corpus increases, the performance of the tagger increases. Unsupervised POS tagging approach does not require pre-defined tag sets and rule sets, on the other hand, it automatically induces the tag sets as well as the transformational rules. Based on this information, the tagger will be able to compute contextual rules or probabilistic information.

With the help of POS Tagging, 'adjectives' with their inclination can be identified, thus, enabling the machine to be able to conclude the final opinion. Determining 'Noun' can aid well in Information Retrieval systems, language translation, speech synthesis and can also be used in various other interdisciplinary applications. In this paper, we will discuss various POS tagging algorithms. In addition to these, we will also discuss our proposed model, which is another enhancement of Memory Based Learning Algorithm. To prove the effectiveness of our proposed model, we will compare each of the POS tagging algorithm to our proposed model.

II. Literature Survey

Considering the importance of Parts-of-Speech Tagging in various fields, various POS Tagging algorithm have been designed and are being utilized for information extraction. We shall discuss them one-by-one.

2.1 Memory-Based Learning Algorithm

Memory-Based Learning Algorithm [4] is a 'Lazy Learning' algorithm for POS Tagging. Going by its classification as a POS Tagger, the algorithm is based on supervised learning approach. An Memory-Based Learning system, a machine learning procedure, is the integration of two major components, they are, a learning component and a performance component. The learning component, performs memory based classification, i.e. words with its exemplified usage is stored in the memory. The performance component, performs similarity based classification, i.e. forms a weighted similarity metrics for each word, to be tagged. In weighted similarity metrics, neighbors are assigned with minimum weights. If the untagged word is found in the lexicon, then in the similarity metrics, it is denoted with a 'Zero' weight. During tagging, its lexicon representation is retrieved and its context is determined. But, when the untagged word is not found in the lexicon, on the basis of its similarity, weights are assigned to each instance of the lexicon. Lexicon representation of the word having minimum weight is retrieved. The resulting pattern is disambiguated using deductions from nearest neighbors and then the context is determined. The output, in this case, is the best guess of the morphological category, the untagged word is used in. The performance of the Memory-Based Learning Algorithm is evaluated, based on the ability to identify similarity between the new situations and the past experience.

Memory-Based Learning Algorithm has two variants [4], namely, IBI-IG and IGTREE. In IBI-IG, the test instance can be classified by calculating the distance between the test instance and the memory instance. The distance is 'Zero', if the two instances are equal and 'One' elsewhere. The information gain or the weight for each feature is assigned, according to its contribution to our knowledge of the morphological class. IGTREE uses a compressed decision tree structure, storing the same information as IBI-IG. While testing, the search can be restricted to the memory instance, having the same feature value as the test instance with the highest weight.

2.2 Multi-Domain Web Based Algorithm

Multi-Domain Web Based Algorithm being based on supervised learning approach, requires a pre-tagged corpus. As the name suggests, Multi-Domain Web Based Algorithm searches for all possible domains over the web. The algorithm does not require any pre-processing of the corpus, since the search is not restricted to the domain of the sentence in the corpus. Whenever Multi-Domain Web Based Algorithm executes, it creates three different forms of web queries [5], specific to each word, which are compatible to most possible search engines.

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.

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**: query " $U_{i-2}U_{i-1}U_i**$". This retrieves alternative right-side contexts for U and its original left-side context.

The web queries are executed over the web server, returning all possible domains with their frequency of occurrence. The algorithm will then compute the conditional probability [5] of occurrence for each domain, based on the context of the sentence.

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

where T is the tag set, and p(h) is the 'history' set for the domain.

The domain or the tag, having maximum probability is assigned to the word under consideration. For each word search, the algorithm creates connection with the web server, transfer of control from the application to the web server and back to the application, which creates a runtime overhead of 0.5 seconds. Thus, the algorithm is efficient only when executed for unknown word, else the execution time for this algorithm will grow substantially high .

2.3 Maximum Entropy POS Tagging Model

Maximum Entropy Model is a combination of statistical and a rule based technique. The MaxEnt Model aims at maximizing the probability distribution, based on the given constraints. The Constraints [6] forces the model to match its feature expectations with those observed in the training data. During tagging, the MxEnt Model creates a sample space (set of all possible outcomes) of the features, which match features in the test data, from the training set. Using the heuristics about the feature occurrence in the test set, the model then computes the probability of each feature. The probability is defined over H×T, where H refers to the tag contexts or 'histories' and T refers to the set of all possible tags. Therefore, the probability [6] refers to

$$\mathbf{P(H, T) = \pi\mu \prod_{J=1}^{K} \alpha_J^{F_J(H,T)}}$$

where π is a normalization constant, {μ, α_1,........,α_k} are the model parameters, {F_1,.....,F_K} are known as features. Higher, the statistical occurrence is, higher, the probability/entropy will be. Since, it is difficult to predict, the behavior of a sparsely occurring feature, its statistics may not be reliable for the tagging. For the tagging of unknown words, generation of sample space features is hypothesized as "rare words" in the training set. A rare word is referred to those words which appears less in the training set.

III. The Proposed Model

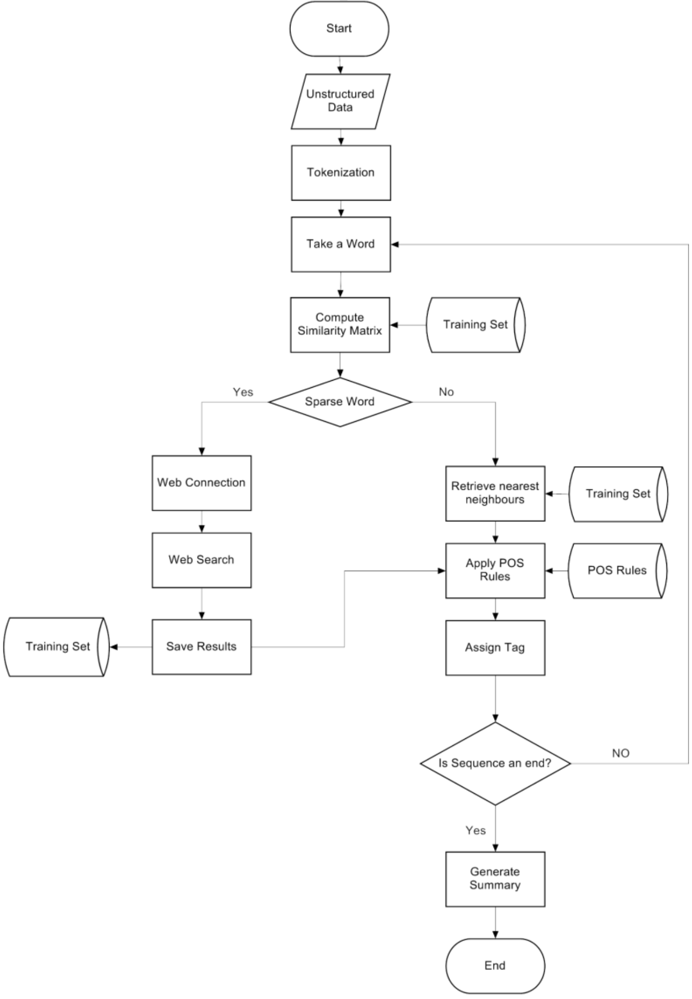
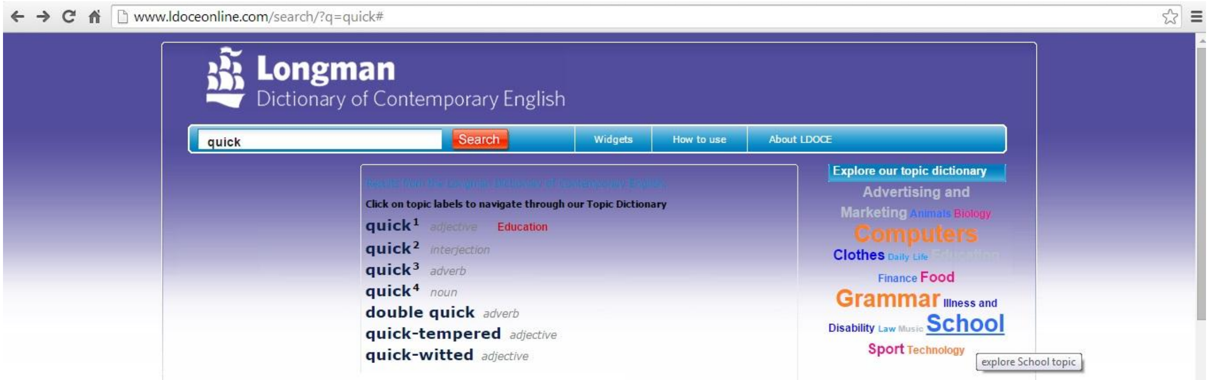


Figure 1: The Proposed Model

The proposed model can efficiently deal with the sparse data problem in Memory Based Learning Algorithm. Memory Based Learning Algorithm is ineffective in finding an appropriate tag for the sparse word i.e. those words for which similar examples are not present in the lexicon. The corpus data is mostly in form of unstructured data, which is supplied as input to the tagger. The process of POS tagging begins with the segmentation of unstructured data into sequence of tokens. Each token represents a word in the sentence. These tokens or words are selected, one at a time, and are searched over the lexicon. Weighted similarity matrix, specific to the word, is then calculated. For an exact match, 'Zero' weight is assigned, but, for a sparse data, '∞' is assigned to the memory instance. If the word under consideration is a 'sparse word', the tagger will then create a web connection and searches for all possible tags. During experimentation, we referred "Longman Dictionary of Contemporary English" for web search. The search results includes all possible domains of the word. With the help of these results, an appropriate POS tag for the word can be computed. In addition to this, the web results are also stored into training repository (for future computations). Whereas, If the word under consideration is not a 'sparse word', Memory Based Learning Algorithm executes. For tagging, the lexicon representation of the instances, having least weight, is retrieved. POS rules are then applied to disambiguate retrieved POS tags. The process recursively repeats itself till the sequence of token ends.

The proposed model, not only, deals with sparse data problem in Memory Based Learning Algorithm, but also, improves efficiency of the training set with every execution. With the addition of web results into the lexicon, the enhancement in efficiency can be measured in two aspects. First, the word, once searched over the web, will no longer be a sparse word, whenever it appears next. Second, approximation of other related words can also be done more appropriately.



Results from the Longman Dictionary of Contemporary English

Click on topic labels to navigate through our Topic Dictionary

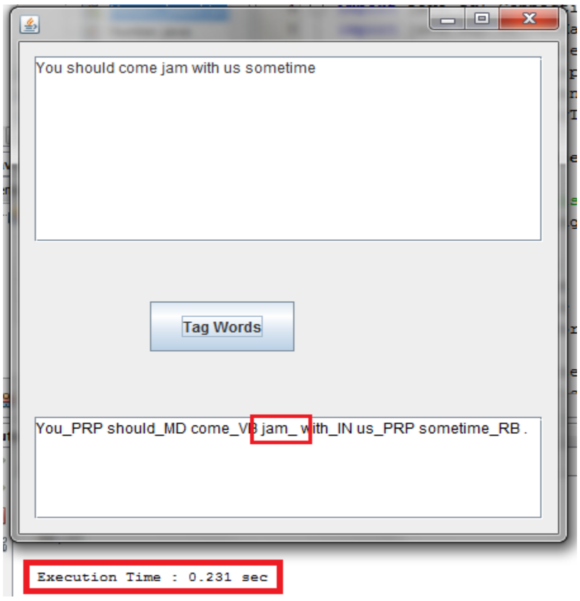
quick¹ adjective Education
quick² interjection
quick³ adverb
quick⁴ noun
double quick adverb
quick-tempered adjective
quick-witted adjective

Figure 2: Web Page showing Various POS Domains for the Word "Quick"

Searching POS tags over the web requires web connection, transfer of controls from the application to the web server and back to the application, increasing the execution time. Saving results into the repository, reduces the web search. This clearly illustrates a tradeoff between space and time complexity in the proposed model i.e. the overall time required for computation can be minimized at the expense of increased memory usage. Effective management of storage space, determines the feasibility of this model. Not all companies, can afford to have unlimited storage space, so as to manage this limited space, the proposed model uses a combination of two memory management page replacement techniques, they are, Least Recently Used(LRU) Page Replacement Technique and Least Frequently Used(LFU) Page Replacement Technique. According to these techniques, whenever some data is to be stored into the lexicon and the storage space is full, the record having low occurrence frequency and high occurrence time period, with respect to others.

IV. Results and Discussions

4.1 Snapshots

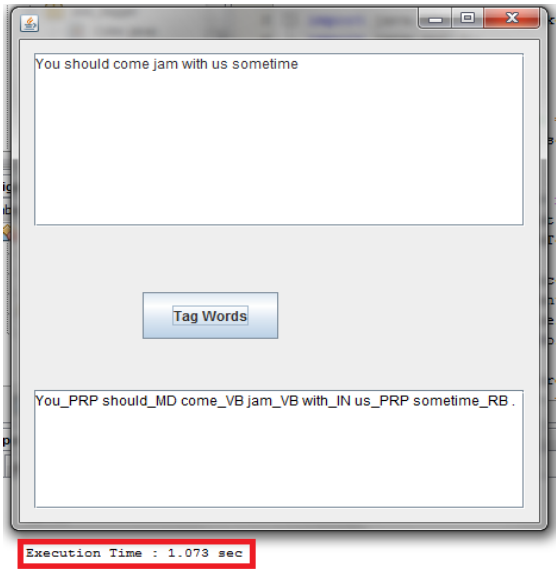


You should come jam with us sometime

You_PRP should_MD come_VB jam_ with_IN us_PRP sometime_RB .

```
Execution Time : 0.231 sec
```

Figure 3: Memory-Based Learning Algorithm when 'jam' was a sparse word

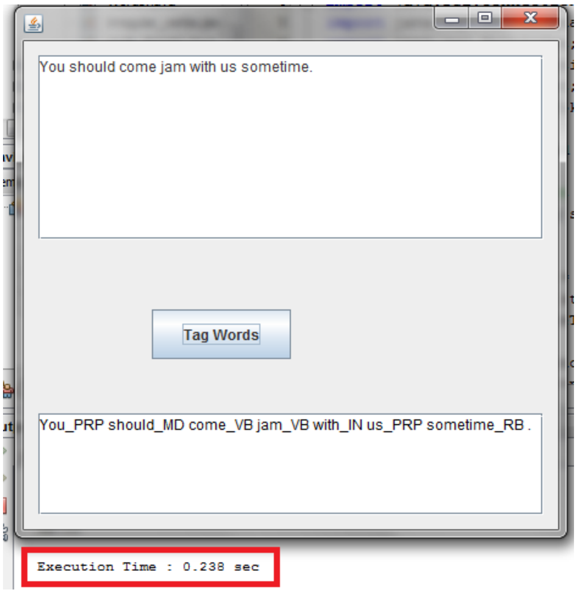


You should come jam with us sometime

You_PRP should_MD come_VB jam_VB with_IN us_PRP sometime_RB .

```
Execution Time : 1.073 sec
```

Figure 4: Proposed Model execution when 'jam' was searched over the web



You should come jam with us sometime.

You_PRP should_MD come_VB jam_VB with_IN us_PRP sometime_RB .

```
Execution Time : 0.238 sec
```

Figure 5: Proposed Model execution when all words are known

4.2 Test Results and Discussions

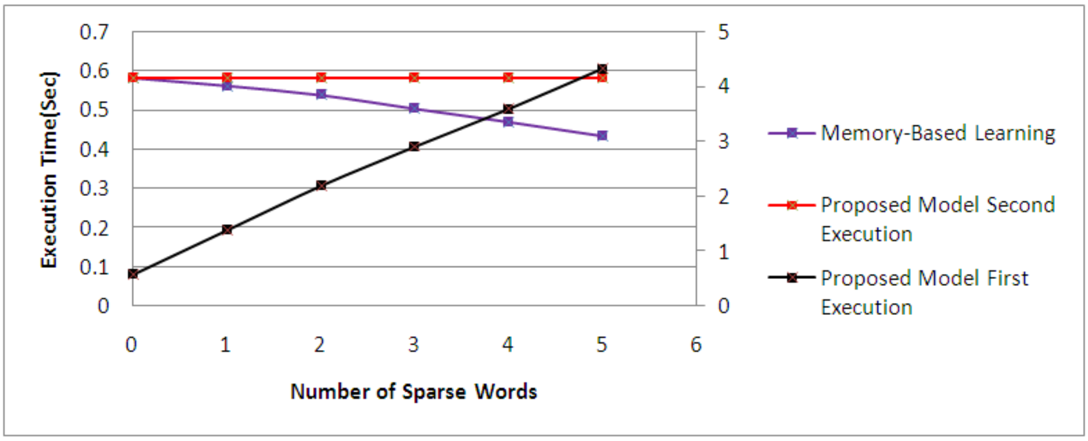


Figure 6: Graph showing varied Execution Time for different Number of Sparse Words in POS Tagging Models

Table 1: Execution Time for POS Tagging Model

Number of Sparse Word	Memory-Based Learning Execution Time (sec)	Proposed Model First Execution Time (sec)	Proposed Model Second Execution Time (sec)
0	0.582	0.582	0.582
1	0.564	1.388	0.582
2	0.538	2.192	0.582
3	0.503	2.904	0.582
4	0.469	3.588	0.582
5	0.433	4.322	0.582

As the results above indicates that the attempt for assigning an appropriate tag to the sparse word "" failed, while executing Memory-Based Learning algorithm. The results also show that, while running the Proposed Model, there is an increase in the execution time, if web search takes place, when compared to the Memory-Based Learning algorithm. This increase is due to runtime overhead of 0.5 seconds, per word, during the web search. Since, the web results are stored into the lexicon, during future executions of the same corpus, the execution time reduced to that of Memory-Based Learning algorithm. A tagger may encounter a worst case corpus during the first execution, i.e. when most words are searched over the web, but, will eventually become the best case corpus in the future executions i.e. when all the words will be found in the repository.

4.3 Comparison

The proposed model, being an enhanced version of Memory Based Learning Algorithm, can effectively solve the sparse data problem by searching over the web. Memory Based Learning Algorithm requires a training set and the performance of this algorithm is based on the efficiency of the training set. The more efficient the training set is, less will be the sparse data and will be high on performance. However, the performance of the proposed model does not solely depend on the training set. The training set is capable of accommodating web results of sparse data, thus, increasing its efficiency with every execution.

In Multi-Domain Web Based Algorithm, unknown words are searched over the web. The web query fetches all possible domains of the unknown word with their occurrence count. The tagger then computes probability of each domain based on the history set. The domain having maximum probability is assigned to the unknown word. The proposed model, on the other hand, does not calculate any probabilistic information, but uses similarity matrix and POS rules, to find the POS tags. In addition to this, the proposed model adds web results to the repository, for future computations, which the Multi-Domain Web Based Algorithm lacks. For every web search, there is a runtime overhead 0.5 sec due to time elapsed in web connection as well as in the back and forth transfer of controls from web server to the application. Saving the results into the lexicon makes the unknown word "known" and eliminates the need of a web search.

Computing POS Tags based on the statistical data of a feature, MaxEnt Model is not only inefficient in dealing with the sparse data, but also, is ineffective while tagging special case sentences. Also, the performance of the model is completely dependent on how efficient the training set is. The proposed model can successfully overcome these limitations.

V. Conclusion

POS tagging plays a vital role in the extraction and analysis of information from the data present in any form. Considering its advancing scope in various fields, researchers have devised various algorithms to improve its performance on the basis of its efficiency and execution time. In this paper, we have proposed an enhanced version of Memory-Based Learning Algorithm, which efficiently overcomes the sparse data problem in Memory-Based Learning Algorithm. In Memory-Based Learning Algorithm, for each word, the tagger will compute similarity matrix, retrieving lexical representation of its nearest neighbors. If the word under consideration is a sparse word, the tagger will not be able to retrieve lexical representation of any memory instance and thus, will not be able to assign any tag to the word. To overcome this shortcoming, in the proposed model, the tagger will search all possible tags, for the sparse word, over the web. The tagger will, then, store the web results on the memory, making sparse word known to the lexicon. Web search increases the execution time due to runtime overhead, but, storing the web results into the lexicon will eliminate the runtime overhead whenever the same word appears next in the text corpus. In addition to this, with every execution, the efficiency of the training set will be improved.

The proposed model has various applications in different fields of study, such as, speech synthesis, Human-Computer interaction, parsers for syntactic analysis, lexicographic research, extracting information from the review sites or blogs, and in many other applications.

Acknowledgements

Authors would like to thank the college authority of Bhagwan Parshuram Institute of Technology, for their support and help in the research work.

References

[1] 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

[2] G.Vinodhini, RM.Chandrasekaran, Sentiment Analysis and Opinion Mining: A Survey, International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X , Volume 2, Issue 6, June 2012.

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

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

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

[6] Adwait Ratnaparkhi, A Maximum Entropy Model For Part-Of-Speech Tagging, Conference on Empirical Methods in Natural Language Processing, Philadelphia, Pa. USA, 1996, W96-0213, 133-142