Word Sense Disambiguation: Speed up and Filtering through Part-of-Speech Tagging

Alok Ranjan Pal, Anupam Munshi
Dept. of Computer Science and Engineering
College of Engineering and Management, Kolaghat
West Bengal, India.
{chhaandasik, anupam.munshi.1991}@gmail.com

Diganta Saha
Dept. of Computer Science and Engineering
Jadavpur University
Kolkata, India
neruda0101@yahoo.com

Abstract— **In this paper, we are going to focus on speed up of the Word Sense Disambiguation procedure by filtering the relevant senses of an ambiguous word through Part-of-Speech Tagging procedure. This proposed approach performs the Part-of-Speech Tagging operation before the disambiguation procedure. As a result, the disambiguation process becomes faster and the filtering of relevant glosses increases the accuracy of the disambiguation result.**

In the training phase, we have used Brown Corpus for Part-of-Speech Tagging and the proposed approach reduces the execution time upto half (approximately) of the normal execution time for a text, containing around 200 sentences. Not only that, we have found several instances, where the correct sense of an ambiguous word is found for using the Part-of-Speech Tagging before the Disambiguation procedure.

Key words: Word Sense Disambiguation (WSD), Part-of-Speech Tagging (POS), WordNet, Lesk Algorithm, Brown Corpus.

I. INTRODUCTION

In human languages all over the world, there are a lot of words having different meanings depending on the contexts. Word Sense Disambiguation (WSD) [1-8] is the process for identification of actual meaning of an ambiguous word based on distinct situations. As for example, the word “Bank” has several meanings, such as “place for monitory transaction”, “reservoir”, “turning point of a river”, and so on. Such words with multiple meanings are ambiguous in nature. The process to decide the appropriate meaning of an ambiguous word for a particular context is known as Word Sense Disambiguation. People have inborn ability to sense the actual meaning of an ambiguous word in a particular context. But the machines do this job by some pre-defined rules or statistical methods.

Two types of learning procedures are commonly used for Word Sense Disambiguation procedure. First, Supervised Learning, where a learning set is considered for the system to predict the actual meaning of an ambiguous word using a few sentences, having a specific meaning for that particular word. A system finds the actual meaning of an ambiguous word for a particular context based on that defined learning set. In this method, learning set is created manually. As a result, it is unable to generate fixed rules for all the systems. Therefore, the actual meaning of an ambiguous word in a given context can’t be detected always. Supervised learning derives partially correct result, if the learning set does not contain sufficient information for all possible senses of the ambiguous word. Even, it fails to show the result, if there is no information in the predefined database.

In Unsupervised learning, an online dictionary is taken as learning set avoiding the inefficiency of Supervised learning. “WordNet”[9-15] is the most widely used online dictionary maintaining “words and related meanings” as well as “relations among different words”.

But in Unsupervised Learning procedures, commonly used for Sense Disambiguation, all the dictionary definitions (glosses) of the ambiguous word are considered from the Dictionary. These glosses are of different types of Part-of-Speech (POS), such as noun, verb, adjective and adverb. In case of commonly used Unsupervised Learning procedures like Lesk Algorithm [16, 17], all the glosses of different Part-of-Speech are considered, which takes some unnecessary additional execution time. As an ambiguous word carries a specific Part-of-Speech in a particular context, we have gone through Part-of-Speech Tagging [18-31] before the WSD procedure. As a result, only the glosses of the related Part-of-Speech are considered.

Using this approach, we have observed two types of betterment in the output. First, the execution of the disambiguation procedure becomes faster and second, as the relevant glosses are filtered, accuracy in disambiguated sense is increased.

Organization of rest of the paper is as follows: Section 2 is about the Theoretical Background of the proposed approach; Section 3 describes the

ISBN: 978-81-923777-9-7

Implementation Background; Section 4 describes the Proposed Approach in detail; Section 5 depicts the experimental results along with comparison; Section 6 represents Conclusion of the paper.

II. Theoretical Background

The most common Unsupervised WSD algorithm is Lesk Algorithm, which uses WordNet as an online dictionary. The algorithm is described below in brief:

A. Preliminaries of Lesk Algorithm

Typical Lesk Algorithm selects a short phrase from the sentence containing an ambiguous word. Then, dictionary definition (gloss) of each of the senses of the ambiguous word is compared with glosses of the other words in that particular phrase. An ambiguous word is being assigned with the particular sense, whose gloss has the highest number of overlaps (number of common words) with the glosses of the other words of the phrase.

Example 1: "Ram and Sita everyday go to bank for withdrawal of money."

Here, the phrase is taken depending on window size (number of consecutive words). If window size is 3, then the phrase would be "go bank withdrawal". All other words are being discarded as "stop words".

Consider the glosses of all words presented in that particular phrase are as follows:

Suppose, the number of senses of "Bank" is 2 such as 'X' and 'Y' (refer Table 1).

The number of senses of "Go" is 2 such as 'A' and 'B' (refer Table 2).

And the number of senses of "Withdrawal" is 2 such as 'M' and 'N' (refer Table 3).

TABLE I. PROBABLE SENSES OF "BANK"

Keyword	Probable sense
Bank	X
	Y

TABLE II. PROBABLE SENSES OF "GO"

Word	Probable sense
Go	A
	B

TABLE III. PROBABLE SENSES OF "WITHDRAWAL"

Word	Probable sense
Withdrawal	M
	N

Consider the word "Bank" as a keyword. Number of common words is measured in between a pair of sentences.

TABLE IV. COMPARISON CHART BETWEEN PAIR OF SENTENCES AND COMMON NUMBER OF WORDS WITHIN A PARTICULAR PAIR

Pair of Sentences	Common number of Words
X and A	A'
X and B	B'
Y and A	A''
Y and B	B''
X and M	M'
X and N	N'
Y and M	M''
Y and N	N''

Table 4 shows all possibilities using sentences from Table 1, Table 2, Table 3, and number of words common in each possible pair.

Finally, two senses of the keyword "Bank" have their counter readings (refer Table 4) as follows:

X counter, $X_C = A' + B' + M' + N'$.

Y counter, $Y_C = A'' + B'' + M'' + N''$.

Therefore, higher counter value would be assigned as the sense of the keyword "Bank" in the particular sentence. This strategy believes that surrounding words have the same senses as of the keyword.

B. Simple (unsmoothed) N-gram and Bigram Model

N-gram [32] is used to compute the probability of a complete string of words (which can be represented either as $w_1 \ldots . w_n$ or w_1^n). If each word, occurring in its correct location, is considered as an independent event, the probability is represented as:

ISBN: 978-81-923777-9-7

$P\ (W_1, W_2, \ldots\ldots, W_{n-1}, W_n)$.

The chain rule to decompose the probability would be:

$P\ (W_1^n)=P(W_1)P(W_2|W_1)P(W_3|W_1^2)\ldots.P(W_n|W_1^{n-1})$.

But, computing the probability like $P(W_n|W_1^{n-1})$ is not easy for a long sequence of preceding words. The Bigram model approximates the probability of a word with respect to all the previous words $P(W_n|W_1^{n-1})$ by the conditional probability of the just preceding word $P(W_n \mid W_{n-1})$.

For example, instead of computing the probability P(rabbit | Just the other day I saw a), the probability is approximated by P(rabbit | a).

III. Implementation Background

This paper adopts the basic ideas from typical Lesk algorithm by introducing some modifications.

A. Simplified Lesk Approach

In this approach, the glosses of only the keyword are considered for a specific sentence instead of all words. Number of common words is being calculated between the specific sentence and each dictionary definition of the particular keyword.

- Consider, earlier mentioned sentence of "Example 1" as follows: "Ram and Sita everyday go to bank for withdrawal of money."
- The instance sentence would be "Ram Sita everyday go bank withdrawal money" after discarding the "stop words" like "to", "for" and so on.
- If "Bank" is considered as the keyword and its two senses are X and Y (refer Table 1). Then, number of common words should be calculated between the instance sentence and each probable senses of "Bank" (refer Table 1).
- Number of common words found would be assigned to the counter of that sense of "Bank". Consider X-counter has the value I' and Y-counter has the value I".
- Finally, the higher counter value would be assigned as the sense of the keyword for the particular instance sentence.
- The dictionary definition (gloss) of the keyword would be taken from "WordNet".
- This approach also believes that entire sentence represents a particular sense of the keyword.

IV. Proposed approach

The proposed approach derives the actual sense of an ambiguous word in two steps. First, the input text is passed through the POS Tagging module, where the POS of the ambiguous word is derived. Second, the input sentence, containing the ambiguous word with derived POS is passed to WSD module, where the disambiguation operation is performed using Simplified Lesk Algorithm.

As the POS of the ambiguous word is derived before WSD operation, the selected dictionary definitions (glosses) are filtered from all the instances present in WordNet (as Noun, Verb, Adjective and Adverb instances). As a result, the disambiguation procedure becomes faster.

Not only that, as the POS of the ambiguous word is derived before the WSD operation, the disambiguation algorithm is applied on only the relevant glosses. As a result, the accuracy of the disambiguation algorithm is increased. The detail explanation of the proposed approach is given below:

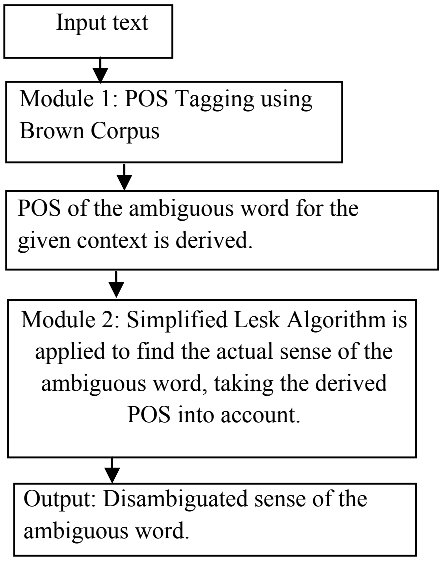

Input text
Module 1: POS Tagging using Brown Corpus
POS of the ambiguous word for the given context is derived.
Module 2: Simplified Lesk Algorithm is applied to find the actual sense of the ambiguous word, taking the derived POS into account.
Output: Disambiguated sense of the ambiguous word.

Figure 1. Modular representation of the overall approach.

Algorithm 1: This algorithm (refer Figure 1) describes the overall approach. The first module is responsible for POS Tagging and the second module is responsible for WSD task.

Input: Input text, containing the ambiguous word.

Output: Disambiguated sense of the ambiguous word.

Step 1: Input text, containing the ambiguous word is passed to Module 1 for finding the POS of the ambiguous word.

Step 2: Simplified Lesk Algorithm is applied to find the actual sense of the ambiguous word, taking the derived POS into account.

Step 3: Stop.

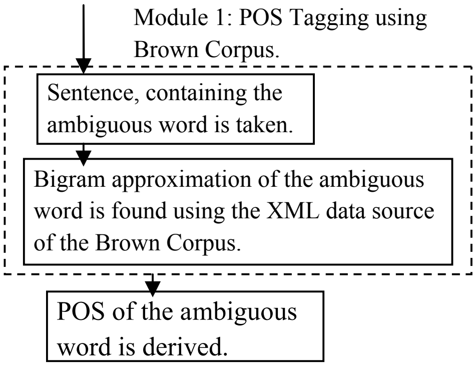

Module 1: POS Tagging using Brown Corpus.
Sentence, containing the ambiguous word is taken.
Bigram approximation of the ambiguous word is found using the XML data source of the Brown Corpus.
POS of the ambiguous word is derived.

Figure 2. Implementation detail of Module 1 for POS Tagging.

Module 1: Algorithm 2: This algorithm (refer Figure 2) finds the POS of the ambiguous word using Brown Corpus. The maximum Time Complexity of the algorithm is O(n^2), which is evaluated at step 2.

Input: Sentence, containing the ambiguous word.

Output: POS of the ambiguous word.

Step 1: Input sentence, containing the ambiguous word is taken.

Step 2: Bigram approximation of the ambiguous word is found using the XML data source of the Brown Corpus.

Step 3: POS of the ambiguous word is derived from the largest approximation value.

Step 4: Stop.

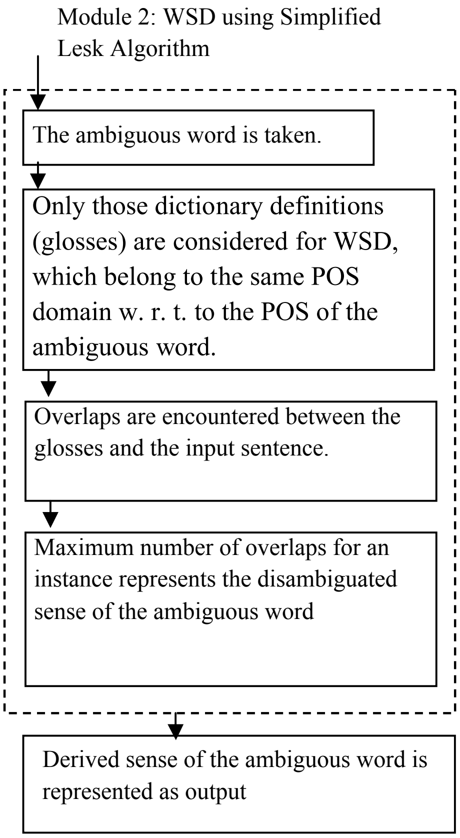

Module 2: WSD using Simplified Lesk Algorithm
The ambiguous word is taken.
Only those dictionary definitions (glosses) are considered for WSD, which belong to the same POS domain w. r. t. to the POS of the ambiguous word.
Overlaps are encountered between the glosses and the input sentence.
Maximum number of overlaps for an instance represents the disambiguated sense of the ambiguous word
Derived sense of the ambiguous word is represented as output

Figure 3. Implementation detail of Module 2 for WSD procedure.

Module 2: Algorithm3: This algorithm (refer Figure 3) derives the actual sense of an ambiguous word using the Simplified Lesk Algorithm. Time Complexity of the algorithm is O(n^3), as finding the total number of overlaps between a particular gloss and the input sentence is of O(n^2) complexity and this procedure is performed for all the n number of glosses.

Input: Ambiguous word with derived POS.

Output: Disambiguated sense of the ambiguous word.

Step 1: The ambiguous word is taken.

Step 2: Only those dictionary definitions (glosses) are considered from WordNet, which belong to the same POS domain w. r. t. to the POS of the ambiguous word.

Step 3: Overlaps are encountered between the glosses and the input sentence itself.

Step 4: The actual sense of the ambiguous word is derived from the maximum number of overlaps for an instance.

Step 5: Stop.

The proposed approach gives better result regarding the execution time and the accuracy of the result, which is described in the next section.

V. OUTPUT AND DISCUSSION

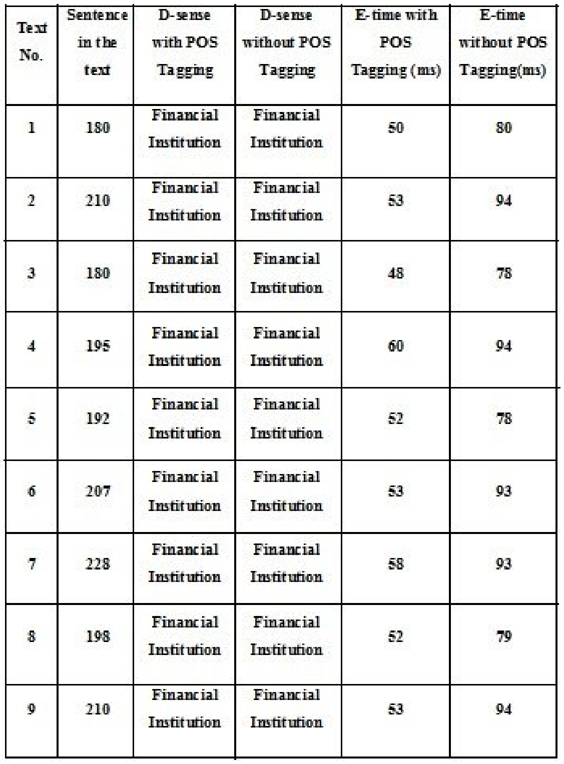
The algorithm is tested on more than 100 texts of different lengths and categories. Average length of the texts is of 200 sentences and two ambiguous words are selected for testing, "Bank" and "Plant".

Next, the Simplified Lesk Algorithm is applied on the input text, containing the POS-tagged ambiguous word. As the POS of the ambiguous word is derived earlier, only those dictionary definitions are selected from WordNet for WSD process, which belong to the same POS domain w. r. t. the POS of the ambiguous word. As a result, the execution time of the WSD process becomes less (refer Table 5).

It is also observed that, as the relevant glosses are considered for the WSD process, accuracy of the disambiguated sense is increased (refer Text no. 10).

Some of the results for target word "Bank" are given in Table 5. All the sample texts are taken from "www.wikipedia.com".

TABLE V. SPEED UP ANALYSIS OF WSD PROCEDURE FOR TARGET WORD "BANK"



Text No.	Sentence in the text	D-sense with POS Tagging	D-sense without POS Tagging	E-time with POS Tagging (ms)	E-time without POS Tagging(ms)
1	180	Financial Institution	Financial Institution	50	80
2	210	Financial Institution	Financial Institution	53	94
3	180	Financial Institution	Financial Institution	48	78
4	195	Financial Institution	Financial Institution	60	94
5	192	Financial Institution	Financial Institution	52	78
6	207	Financial Institution	Financial Institution	53	93
7	228	Financial Institution	Financial Institution	58	93
8	198	Financial Institution	Financial Institution	52	79
9	210	Financial Institution	Financial Institution	53	94

D-sense means Disambiguated sense,

E-time means Execution time,

MS means Mille Second.

In the following sample test (Text no. 10), it is depicted that, in the given input text the ambiguous word "Plant" carries the actual sense (decided by human) as "Living Organism", which is in noun sense, but when the algorithm runs without POS Tagging, it derives the sense "Contact", which is in verb sense and obviously it is a wrong sense for this context according to human decision. This text is also taken from "www.wikipedia.com".

Accuracy measurement of WSD procedure using the proposed approach is described below with a sample text.

Text no. 10:

Plants, also called green plants, are living organisms of the kingdom Plantae including such multi cellular groups as flowering plants, conifers, ferns and mosses, as well as, depending on definition, the green algae, but not red or brown seaweeds like kelp, nor fungi or bacteria.

Green plants have cell walls with cellulose and characteristically obtain most of their energy from sunlight via photosynthesis using chlorophyll contained in chloroplasts, which gives them their green color. Some plants are parasitic and may not produce normal amounts of chlorophyll or photosynthesize. Plants are also characterized by sexual reproduction, modular and indeterminate growth, and an alternation of generations, although asexual reproduction is common, and some plants bloom only once while others bears only one bloom.

Precise numbers are difficult to determine, but as of 2010, there are thought to be 300–315 thousand species of plants, of which the great majority, some 260–290 thousand, are seed plants. Green plants provide most of the world's molecular oxygen and are the basis of most of the earth's ecologies, especially on land. Plants described as grains, fruits and vegetables form mankind's basic foodstuffs, and have been domesticated for millennia. Plants enrich our lives as flowers and ornaments. Until recently and in great variety they have served as the source of most of our medicines and drugs. Their scientific study is known as botany.

Output:

Target word: Plant.

Actual sense: Living Organism (Noun).

Derived sense with POS Tagging: Living Organism (Noun).

Derived sense without POS Tagging: Contact (Verb).

VI. Conclusion and Future Work

The proposed approach speeds up the WSD procedure by filtering the only relevant glosses and increases the accuracy of the WSD procedure as well. The execution time differences between the two cases (with and without POS Tagging procedure) might be increased, if few system calls and other system related tasks are handled properly. The obvious operations (loop, memory allocation, condition check, function call etc.) for POS Tagging took some time, which is included in the cited result. Otherwise, the actual time difference could have been better.

References

1. R. S. Cucerzan, C. Schafer, D. Yarowsky, “Combining classifiers for word sense disambiguation, Natural Language Engineering,” Vol. 8, No. 4, 2002, Cambridge University Press, pp. 327-341.
2. M. S. Nameh , M. Fakhrahmad, M. Z. Jahromi, “A New Approach to Word Sense Disambiguation Based on Context Similarity,” Proceedings of the World Congress on Engineering, Vol. I, 2011.
3. Gaizauskas, “Gold Standard Datasets for Evaluating Word Sense Disambiguation Programs,” Computer Speech and Language, Vol. 12, No. 3, Special Issue on Evaluation of Speech and Language Technology, pp. 453-472.
4. R. Navigli, “Word Sense Disambiguation: a Survey,” ACM Computing Surveys, Vol. 41, No. 2, 2009, ACM Press, pp. 1-69
5. N. Ide , J. Véronis, “ Word Sense Disambiguation: The State of the Art,” Computational Linguistics, Vol. 24, No. 1, 1998, pp. 1-40.
6. W. Xiaojie, Y. Matsumoto, “Chinese word sense disambiguation by combining pseudo training data,” Proceedings of The International Conference on Natural Language Processing and Knowledge Engineering, 2003, pp. 138-143.
7. C. Santamaria, J Gonzalo, F. Verdejo,”Automatic Association of WWW Directories to Word Senses,” Computational Linguistics, Vol. 3, Issue 3, Special Issue on the Web as Corpus, 2003, pp. 485-502.
8. J. Heflin, J. Hendler, “A Portrait of the Semantic Web in Action,” IEEE Intelligent Systems, Vol. 16, No. 2, 2001, pp. 54-59.
9. S. G. Kolte, S. G. Bhirud,” Word Sense Disambiguation Using WordNet Domains,” First International Conference on Digital Object Identifier, 2008, pp. 1187-1191.
10. G. Miller: WordNet: An on-line lexical database, International Journal of Lexicography, Vol. 3, No. 4, 1991
11. Y. Liu, P. Scheuermann, X. Li, X. Zhu, “Using WordNet to Disambiguate Word Senses for Text Classification,” Proceedings of the 7th International Conference on Computational Science, Springer-Verlag, 2007, pp. 781 – 789.
12. A. J. Cañas, A. Valerio, J. Lalinde-Pulido, M. Carvalho, M. Arguedas,”Using WordNet for Word Sense Disambiguation to Support Concept Map Construction,” String Processing and Information Retrieval, 2003, pp. 350-359.
13. G. A. Miller, “WordNet: A Lexical Database,” Comm. ACM, Vol. 38, No. 11, 1993, pp. 39-41.
14. H. Seo, H. Chung, H. Rim, H. Myaeng, S. Kim, “Unsupervised word sense disambiguation using WordNet relatives,” Computer Speech and Language, Vol. 18, No. 3, 2004, pp. 253-273.
15. G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. J. Miller, “WordNet An on-line lexical database,” International Journal of Lexicography, Vol. 3, No. 4, 1990, pp. 235-244.
16. S. Banerjee, T. Pedersen, “An adapted Lesk algorithm for word sense disambiguation using WordNet,” In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February, 2002.
17. M. Lesk, “Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone,” Proceedings of SIGDOC, 1986.
18. S. Dandapat, “Part Of Specch Tagging and Chunking with Maximum Entropy Model,” in Proceedings of the IJCAI Workshop on Shallow Parsing for South Asian Languages, (Hyderabad, India), pp. 29–32, 2007.
19. A. Ekbal, R. Haque, and S. Bandyopadhyay, ”Maximum Entropy based Bengali Part of Speech Tagging,” in A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal, vol. 33, pp. 67–78.
20. A. Ekbal, R. Haque, and S. Bandyopadhyay, “Bengali Part of Speech Tagging using Conditional Random Field,” in Proceedings of the seventh International Symposium on Natural Language Processing, SNLP-2007, 2007.
21. A. Ratnaparkhi, “A maximum entropy part-of -speech tagger,” in Proc. of EMNLP’96., 1996.
22. PVS. Avinesh, G. Karthik, “Part Of Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning,” In Proc. of SPSAL 2007, IJCAI, India, 21-24.
23. T. Brants, “TnT-A Statistical Part of Speech Tagger,” In Proc. of the 6th ANLP Conference, 224-231.
24. D. Cutting, J. Kupiec, J. Pederson and P. Sibun,”A Practical Part of Speech Tagger,” In Proc. of the 3rd ANLP Conference, 133-140, 1992.
25. A. Ekbal, and S. Bandyopadhyay, “Lexicon Development and POS tagging using a Tagged Bengali News Corpus. In Proc. of FLAIRS-2007, Florida, 261-263.
26. M. Mcteer, R. Schwartz and R. Weischedel, “Empirical studies in part-of-speech labeling. Proceedings Of the 4th DARPA Workshop on Speech and Natural Language,” pp. 331-336, 1991.
27. B. Merialdo, “Tagging English text with a probabilistic model,” Computational Linguistics, 20(2):155-171, 1994.
28. M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, “The infinite hidden markov model,” Advances in Neural Information Processing Systems, 14:577 –584, 2002.
29. Chris Biemann, Claudio Giuliano, and Alfio Gliozzo,”Unsupervised part-of-speech tagging supporting supervised methods,” In Proceedings of RANLP. Eric Brill. 1994. Some advances in transformationbased part of speech tagging. In National Conference on Artificial Intelligence, pages 722–727, 2007.
30. J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani, “Beam sampling for the infinite hidden markov model,” In Proceedings of the 25th international conference on Machine learning, volume 25, Helsinki, 2007.
31. Jianfeng Gao and Mark Johnson, “A comparison of bayesian estimators for unsupervised hidden markov model pos taggers,” In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 344–352.
32. Daniel Jurafsky and James H. Martin. *Speech and Language Processing*, Prentice Hall, 2000.