# Decision Tree based Supervised Word Sense Disambiguation for Assamese

Jumi Sarmah
Research Scholar
Dept. of IT, Gauhati University
Assam, India

Shikhar Kr. Sarma, PhD
Professor and H.O.D.
Dept. of IT, Gauhati University
Assam, India

## ABSTRACT
Word Sense Disambiguation (WSD) aims to disambiguate the words which have multiple sense in a context automatically. Sense denotes the meaning of a word and the words which have various meanings in a context are referred as ambiguous words. WSD is vital in many important Natural Language Processing tasks like MT, IR, TC, SP etc. This research paper attempts to propose a supervised Machine Learning approach- Decision Tree for Word Sense Disambiguation task in Assamese language. A Decision Tree is decision model flow-chart like tree structure where each internal node denotes a test, each branch represents result of a test and each leaf holds a sense label. J48 a Java implementation of C4.5 decision tree algorithm is taken for experimentation in our case. A few polysemous words with different real occurrences in Assamese text with manual sense annotation was collected as the training and test dataset. DT algorithm produces average F-measure of .611 when 10-fold crossvalidation evaluation was performed on 10 Assamese ambiguous words.

## General Terms
Natural Language Processing, Machine Learning, Computer Science

## Keywords
Word Sense Disambiguation, Decision Tree, Assamese, Supervised approach

## 1. INTRODUCTION
Natural language is ambiguous. It is an easy task for a human to understand and disambiguate the ambiguous words but a trivial task for a computer to do so. Ambiguity can occur at various levels of NLP- lexical, syntactic, and semantic and discourse level. The paper here concentrates on Lexical Semantic ambiguity task. Lexical Semantic ambiguity takes place when a word/lexicon or a phrase has multiple meanings associated with it. Let us consider a simple example: "*I am going to the bank to deposit some money"*. The term "bank" has two concepts- river and financial. "*বহুজনে নিজৰ মনতে উত্তৰ বিচাৰি পায়*" (bahujane nijor monote uttar bisari pai) the term "*উত্তৰ* /uttar" is ambiguous in Assamese Language as it has two concepts- "reply" and "north-sense". WSD is the task of understanding the sense of an ambiguous word in a piece of context. It basically assigns the appropriate sense to a word depending on the particular context where it occurs in an automated manner. Various survey works on WSD task are properly described by [1] and [2]. Features represent the context where the target word lies and Lexical, Syntactic and Semantic features provides clues for sense disambiguation task.

WSD task involves mainly three important steps: The first step is the sense discovery of the ambiguous word along with their sense definitions. The second step is the process or the way by which appropriate sense assignment is done. For this the context has to be represented as features. The third important step is the machine learning. The machines need to learn the disambiguation either using some statistical techniques, manual created rules and knowledge-based approaches. Knowledge-based resource like machine readable dictionaries can be used for disambiguation task.

Context plays the most integral part in WSD from Machine Learning perspective both in discovery of word senses and disambiguation task. Wittgenstein's (1953) [3] view about word sense is "the meaning of a word is its use in the language" because there is nothing else for a computer to observe. Words real usage by language communities characterizes both similarity of meanings and differences. For eg. "shoot" lexicon in the phrase "shoot a bird"- by a hunter" and "shoot a question- by a journalist" has individual meanings. In context, the words have their own base forms which are obtained by some morphological analysis. These base-forms are recorded in sense inventories with their definitions by the lexicographers. Basically, a context has three main dimensions- Context size, modality and depth processing and it has great impact in disambiguation task. Context size may be divided into three categories- zero context, local context and global size context. Zero contexts consist of the target word only; phrase and clause comprises the local context and sentences, topics forms the global context. For a human, the primary modalities for understanding language are hearing and vision. Besides these several feelings like physical, smell, taste should be given as initial values in inventories so that it help the machines to make proper sense generalizations of semantic features( like animate or inanimate etc.) of word say "shooting stars". Various phases like token analysis, morphological analysis and syntactic analysis are observed in context depth processing.

J48 is JAVA implementation of the C4.5 decision tree algorithm. C4.5, successor of ID3 decision tree algorithm and was developed by J. Ross Quinlan. A decision tree has root node, internal nodes where each node denotes a test on an attribute value, each branch of the tree denotes outcome of the test and each leaf node denotes a class label. Decision tree induction by C4.5 is used for classification task and so it is referred as statistical classifier in [4].

Assamese language belongs to the Indo-Aryan language family and is the official language of Assam and only few

works are done in Assamese NLP perspective. Works on Document Classification, Information Retrieval, Machine

Translation, and Spell checker are the few among the going works in the NLP lab of Gauhati University. Only countable number of works is done in the field of Assamese WSD.

Disambiguation of ambiguous words automatically has been a goal in computational linguistics field from long back years but there has always been state-of art accuracy as reported by the researchers in their research papers. This paper aims to explore research on WSD task for Assamese language using supervised approach- Decision Tree algorithm. The rest of the paper is organized as follows: Section2 reviews approaches used as an initiation for the WSD task in various Indian languages, Section3 describes the Decision Tree and C4.5 algorithm, Methodology used in this paper is described step-by-step in Section4, Hold-out and crossvalidation evaluation is briefly mentioned in Section5, Section6 discusses and analyses the results of test sample of ambiguous words. The paper is concluded in Section7.

## 2. LITERATURE REVIEW

History reveals that Supervised, Knowledge-based, Unsupervised, Semi-supervised approaches are used for Word Sense Disambiguation tasks. Many Indian languages like Assamese, Manipuri, Tamil, Malayalam, Hindi, Kannada, Nepali and Punjabi have done research work in this field of Natural Language Processing. Various approaches are used for the WSD task in Indian languages and is mentioned below:

Assamese belongs to the Indo-European language family. [5] Proposes a Supervised approach- Naive Bayes Classifier for word sense disambiguation task and achieved an Accuracy of 71% with Iterative Learning mechanism when trained with sample of size 2700 and tested with sample of 300. Another WSD approach in this language is reported to have a F-measure of 55.6% when Unigram Co-occurrence features (context window of two) was explored by [6]

Manipuri belongs to the Sino-Tibetan language family. [7] Proposed a Decision Tree based WSD model. Decision Tree classifier which conducts recursive partition over the instances. They proposed CART (Classification and Regression) algorithm for training the classifier and achieves an accuracy of 71.75%.

Word Sense Disambiguation task was implemented for Tamil which belongs to the Dravidian language family. A supervised approach- Support Vector Machine (SVM) was used for this task [8].

Disambiguation of ambiguous words for Malayalam language was implemented by simulating the human behavior to a computer system [9]. Malayalam also belongs to the Dravidian language family and spoken mainly in the state of Kerala.

WSD task for Hindi language is mentioned using the lexical knowledge base WordNet with overlapping approaches by [10]. Hindi is the official language of India and belongs to the Indo-European language family.

A supervised algorithm -Naïve Bayes Classifier was proposed for Kannada WSD task [11]. Kannada is a Dravidian language which is spoken mainly in the state of Kannada.

[12] Knowledge based approaches- Overlap-based, Conceptual Distance was used for Word sense Disambiguation task for Nepali Language. It is an Indo-Aryan language spoken by the Nepali, Bhutanese and some Burmese communities.

Knowledge based Approach- Walker's Algorithm was proposed for Assamese Language by [13].

WSD tasks on Decision Tree was researched by [14] on Portuguese Nouns and produces an average accuracy of .70.

[15] Uses senseval3 to evaluate the word sense disambiguation task for training and test data purpose and found out that the accuracy of decision tree is 45.14%.

## 3. DECISION TREE AND C4.5

A decision tree is a classifier that recursively partition over the data space. Basically, it is composed of a root node, branches, internal nodes and leaf nodes. Each internal node is the decision node representing test on an attribute or a set of attributes and each branch denotes a value of the input attributes. The leaf node denotes the class label say "YES" or "NO". Path starting from the root node to the leaf node forms a classification rule. Figure1 below gives a description of a decision tree. Here, circle represent decision node, square represent leaf node and there are three splitting attributes-age, student and credit and two class labels- NO and YES.

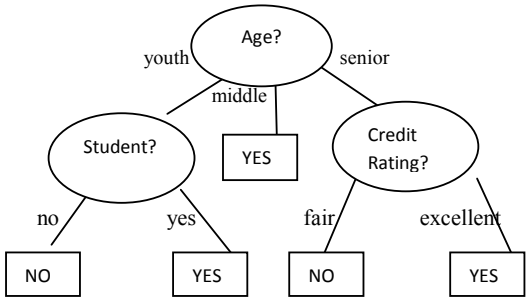


**Fig 1: A Decision Tree**

Decision Tree is a graphical rather than tabular. It is flow-chart without loops and it takes more space or room than decision lists. It shows the order of evaluation of the conditions. It may be viewed as unordered rule sets where each leaf corresponds to a single rule with a condition part consisting of the conjunction of all edges from root to leaf. The rules in the set are non-overlapping i.e., each example covers only single rule and this constraint make classification easier ( no conflicts from multiple rules) and easier to understand. Whereas decision lists follows ordered rule sets with at most k conditions per rule and are strictly more expensive than decision tree.

Principle of building a decision tree: Given a set of training sample with their sense category, we need to apply a mathematical function to get the best splitting attribute. Determining the best splitting attribute, the remaining training sample is partitioned into several parts. A recursive procedure is followed in each partition to form a decision tree. A recursive partitioning stops if all the tuples belongs to the same class label, or if there are no remaining attributes on which the tuples may be partitioned majority voting is applied else if there are no tuples for a branch a leaf is created with the majority class in the sample. The edge denotes the outcome value of each test attribute node. Each branch forms a classification rule which is used to categorize test instances. The criteria for choosing splitting attribute in C4.5 algorithm is Information Gain Ratio. The algorithm is described below in Figure 2.

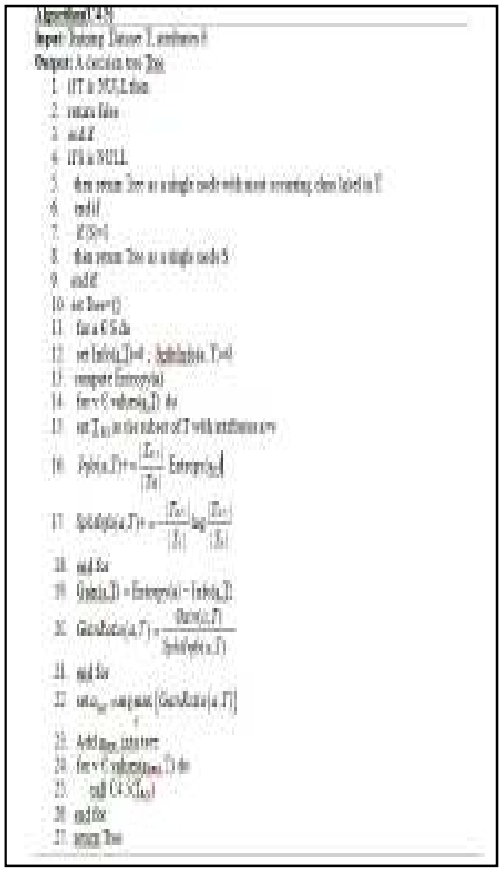

**Fig 2: C4.5 Decision Tree Algorithm**

Let the number of classes be C and P(S,j) is the proportion of instances in S that are associated to $j^{th}$ class.

$$Entropy(S) = -\sum_{j=1}^{C} P(S, j) \times \log P(S, j)$$

Information Gain by training dataset T is defined as:

$$Gain(S,T) = Entropy(S) - \sum_{v \in Val(T_s)} \frac{|T_{s,v}|}{|T_s|} Entropy(S_v)$$

Where $Val(T_s)$ is the set of values of S of T
$T_s$ is the subset of T induced by S

$T_{S,v}$ is the subset of T in which attribute S has a value of v

Information Gain Ratio of attribute S is defined as:

$$GainRatio(S,T) = \frac{Gain(S,T)}{SplitInfo(S,T)}$$

Where

$$SplitInfo = -\sum_{v \in Val(T_s)} \frac{|T_{s,v}|}{|T_s|} \times \log \frac{|T_{s,v}|}{|T_s|}$$

The algorithm [16] is described above and the Information Gain Ratio criteria computation is mentioned in Algorithm in lines (11-21) using above equations. A recursive call is made in Line 25. The algorithm with proper explanation is mentioned in Book [17]

Computational complexity of the Algorithm given training set D: O(n * |D|* log(|D|)) where n is the number of attributes describing the tuples in D and |D| number of training tuples in D.

# 4. OUR METHODOLOGY

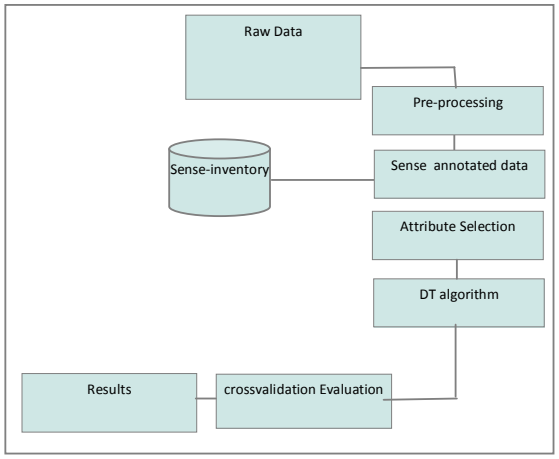


**Fig 3: Process Flow**

Section 4 describes the various system modules involved in our WSD task using C4.5 Decision Tree. A process flow diagram is shown in Figure 3 above. No work on Assamese WSD using this algorithm is available in the web. The modules proposed are systematically described step-by-step below. Starting from collection of raw data to preparing sense tagged corpus, methodology is described and results are concluded.

## 4.1 Raw Trained/Test Data

Assamese Corpus developed in NLP lab of Gauhati University by [18] consists of 1.5 million words. It is a large collection of digitalized documents with UNICODE encoding. Genre of kind literature, media, and scientific material are found in the Assamese corpus collection. Corpus is the basis of all tasks in Natural Language Processing. Various tasks like spell checker, MT, Text Processing involves the use of Corpus. 65K sentences are extracted from the corpus with the help of sentence end-mark symbol "Dari" denoted by "।".

## 4.2 Pre-processing Phases

### *Data Cleaning*

The collected 65K sentences from corpus, consist of noisy data. Noisy context are some wrong information or some unstructured data which is not understood by the machines. Various punctuation symbols like , ; : extra spaces between words, other language characters are removed from the corpus. Small length sentences of size two; three are deleted as it will play fewer clues for disambiguation task. A total of 50K sentences are found after removal of small length sized sentences.

### *Stop-word removal*

Stop-words are most frequently occurring words in a text and provide less valuable information for disambiguation task. Therefore, they are removed from the context. Some example of Assamese stop words are আৰু (aaru:and), বা (ba:or), হয় (hoi:ya) .

### *Stemming*

Morphological analysis or Stemming is the process to reduce a word in a context to its base form or nearly-root word order form. As for example: the word "উত্তৰটো" has basic root form "উত্তৰ". On performing stemming it would help us in matching the same root word form with the entries in the sense repository form.

### *Correction of Inconsistent data*

Inconsistent data means various typos error like spelling/typing mistake occurred in the context. Manual intervention in the correction of such type of data was performed.

## 4.3 Sense Inventory Preparation

With the help of Assamese WordNet and Corpus (described above), sense inventory of size 160 ambiguous are found. Assamese WordNet developed by [19] is an important lexical database. All together seventeen Indian languages Hindi, Bodo, Manipuri, Malayalam, Kannada, Punjabi, Gujarati, Tamil, Telugu, Sanskrit, Marathi, Nepali, Urdu, Oriya, Kashmiri and Konkani including Assamese WordNet was developed under the Indo-WordNet project following the structure of Princeton WordNet. WordNet consist of ID which act as a primary key for identifying any word in WordNet, CAT indicates the category of the Parts Of Speech, SYNSET lists the synonymous words in a most used frequency order and GLOSS describes the concept of any synset. It consists of Text-Definition and Example-Sentence. Text Definition contains concepts denoted by synset and Example- Sentence portrays the use of any word in the synset list. There are various semantic relation that occur between synsets in WordNet. They are Hypernymy-Hyponymy (IS-A/Kind of), Meronymy (PART-Of). An example describing the Hypernymy/Hyponymy relation is: This synset order in Assamese WordNet: {ঘৰ, গৃহ, আলয়, নিলয়} *(ghaar, grih, aalay, nilay:home)* has Hypernymy relation (IS-A):{আবাস, নিবাস, বাসস্থান}*(aabaas, nibaas, baasastaan:shelter-place)* Meronymy relation (PART-OF) {বহা_কোঠা, বৈঠকখানা} *(bahaa_kothaa, boithakkhaanaa: drawing-room)* and Hyponymy relation links to(KIND-OF) {পঁজা, জুপুৰী, খেৰীঘৰ, কুটীৰ} *(pajaa, jupurii, kheriighar, kutiir: cottage)* synsets respectively. Synsets are created for non-functional words. The non-functional words have POS- Noun, Verb, Adjective and Adverb. The total number of synsets in Assamese WordNet is 30966.

To disambiguate Assamese ambiguous words, we need to detect at first few ambiguous words in this Indo-Aryan language, Assamese. With the help of Assamese corpus, the above process flowchart (Figure 4) was followed to derive ambiguous words. The pre-processing phases are already mentioned above. Along with the previous phases mentioned, duplicate words filtration was done so that a "word-list" was formed from the 50K sentences and a distribution matrix was created to get the ambiguous words with the help of a

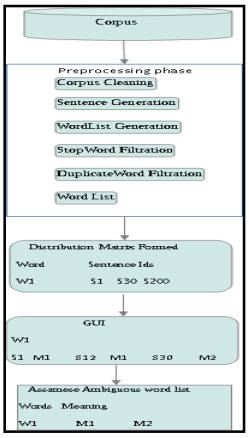


**Fig 4: Figure deriving ambiguous word from corpus**

**Fig 5: GUI**

Graphical User Interface (Figure 5 above). Those words which have multiple meanings in different sentences (s1, S30, s2002) are considered as ambiguous.

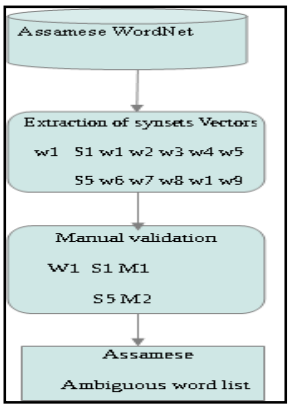


**Fig 6: Figure deriving ambiguous word from WordNet**

As all entries in WordNet are not ambiguous so a methodology is followed to derive the ambiguous words. Words which occur in different synset entries are extracted

first and later manually validated to derive ambiguous words from WordNet. A total of 100 ambiguous words are currently found from WordNet. A flowchart of deriving ambiguous words along with their definition of sense from WordNet is mentioned in Figure 6.

Preparation of sense-annotated data

The 50K sentences contain 2.7K ambiguous words based on 160 ambiguous words in the sense inventory. Sense-annotation was prepared manually and tagged with the appropriate sense accordingly. A sample of annotated data is shown below: Eg: বিশেষ্ট কলা<শিল্প-arts-culture> সমালোচক নীলমণি ফুকনে তেখেতৰ শিল্পকলা দর্শন উল্লেখ কৰিছে । In the above sentences, the word "কলা" is ambiguous and manual sense-annotators tagged the appropriate sense relevant to that context.

## 4.4 Features/Attribute Selection

Before the trained data is feed to the classifier, important features are extracted from the training data. A man or a child learns to assign meaning to a word based on its usage in a language community. This work considers local lexical features as clues to disambiguate the ambiguous words. These features appear at a certain context range with the index term. These are the left and right features of the target word with the range {-2,-1,0,+1,+2}. The more far the distance from the target word their influence gets diminishes. The target word occurring at the first and the last position gets fewer clues for the disambiguation task.

## 4.5 Feeding the DT classifier

Machine Learning involves an algorithm or a classifier that learns from some sense-tagged data. The features are fed to the classifier and the algorithm identifies pattern and infer predictions from them. The figure 7 below shows a sample of Decision Tree for Assamese WSD task. With the help of Gain Ratio the splitting attribute (Ambiguous word) is determined. The values of the splitting attribute are the outcome of the test splitting attribute. Subtracting the splitting attribute we get the remaining attributes and are partitioned accordingly. And then again the splitting attribute is derived from the each partition table and values of the test attribute helps to reach the sense class label. The values were discrete-values in our case. The process of creation of the tree is terminated when all the tuples belong to the same class label for the attribute values. The example of decision tree shown below is temporary for explanation in this paper considering the next_word, next_next_word, prev_word, prev_prev_word to the ambiguous word.

# 5. EVALUATION

The various evaluation approaches for word sense disambiguation task and metrics like Precision, Recall, F-measure are described in this section. Usage of words by different language communities arouses different sense of a word and the written or spoken context level gives clues for disambiguation. For classification task, error rate measures the performance of a model. Error rate is the percentage of incorrectly classified instance in a test set. Two evaluation procedures:

Hold-out and cross validation are mentioned briefly below:

## 5.1 Holdout Evaluation

If the size of sense-annotated data is available, then we need to simply take two independent samples- training (80%) and test (20%) data set. The more the trained data, the better is the model. The more the test data, more accurate is the error estimate. Problem: Splitting or Division of trained data and test data is a difficult job. If the trained data do not contain information regarding a particular class present in the test set, the model will wrongly classify. Therefore, this evaluation procedure can have high variance as it depend which data points ends in training set and test set. Solution: This problem can be solved using stratified hold-out method. We need to sample in such a manner that each class is represented in both sets (training and test).

Another evaluation procedure to get an accurate estimate is by k fold cross validation

## 5.2 K-fold Cross validation Evaluation

Crossvalidation improves over the holdout evaluation procedure as it divides the whole data sets into k subsets. At each time one of the k-subset is used as the test data and the remaining (k-1) as the trained data. This process is repeated k times and average error rate is calculated. The advantage of the method is that every data set gets to be in the training set and test set once. It makes less matter how the data is spitted. The variance is reduced as 'k' gets increased iteratively. Often the folds are stratified before crossvalidation is implemented. The disadvantage is that it has to re-run the trained data k times to make an evaluation. 10 fold crossvalidation is generally used but in general k remains dynamic. A simple table below will make us understand the concept in a better way:

**Table 1. Crossvalidation Evaluation**

| Training Set | Testing Set | Accuracy |
|---|---|---|
| $P_2,\ldots..P_K$ | $P_1$ | $A_1$ |
| $P_1,P_3,\ldots.P_K$ | $P_2$ | $A_2$ |
| …….. | … | …… |
| $P_1,P_2,\ldots.P_{K-1}$ | $P_K$ | $A_K$ |
| **Average** | | A |

This evaluation procedure ensures that each data points is trained and tested exactly once. Every data point is used to understand how well our model performs the tasks of learning from data and predicting new data. This method gives us an idea how well the classifier will do when asked to make new predictions of unseen data. The k-fold Crossvalidation evaluation was performed on 10 ambiguous words with their varioust sense occurrences in different context.

# 6. METRICS USED

Crossvalidation evaluation is performed and following measures or metrics are used to determine the performance in our disambiguation task. For binary classification problem, each testing instance may have four possible situations as shown in Table2:

**Table 2. Situations of a testing instance**

| | | | |
|---|---|---|---|
| Predicted | YES | Actual | YES |
| Predicted | YES | Actual | NO |
| Predicted | NO | Actual | YES |
| Predicted | NO | Actual | NO |

**Table 3. Contingency table**

| | | Predicted Class | |
|---|---|---|---|
| | | YES | NO |
| Actual Class | YES | TP | FN |
| | NO | FP | TN |

Table 3 above shows a contingency table that records the total number of testing instances for each situation

TP: true positive: class members classified as class members only

FN: false negative: class members classified as class non-class members only

FP: false positive: non class members classified as class members only

TN: true negative: non-class members classified as non-class members

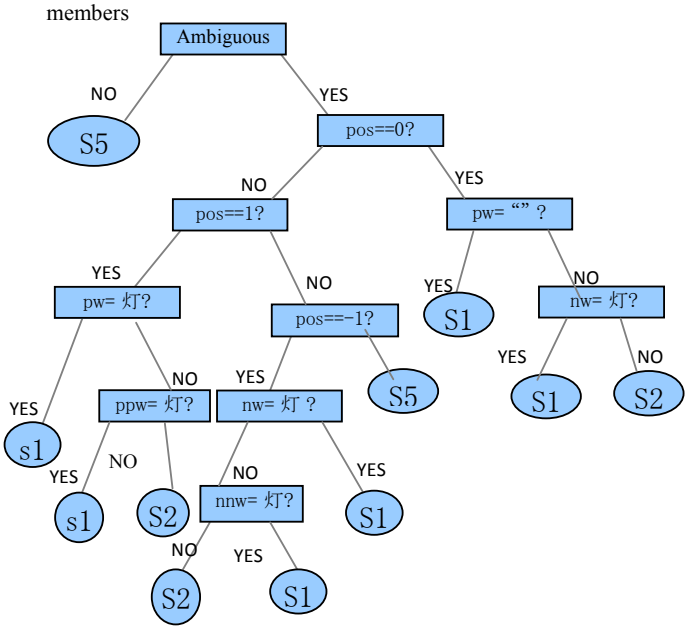


**Fig 7: Decision Tree based Assamese WSD**

Precision: Number of class members correctly classified over the total number of instances classified as class members.

$$\Pr ecision = \frac{|TP|}{|TP + FP|}$$

Recall: Number of class members correctly classified over total number of class members

$$\text{Re}call = \frac{|TP|}{|TP + FN|}$$

F-measure: Precision and Recall can be combined in the F-measure

$$Fmeasure = \frac{2 \times recall \times precision}{recall + precision}$$

Precision and Recall are equally important for evaluation task as they measure how precise and complete the classification is on positive class.

## 7. RESULT ANALYSIS

Testing was done on a few ambiguous words with their various occurrences in a text. Results of disambiguation of 20 ambiguous word are shown below in Table4. The column header indicates the ambiguous word, NOS-number of senses of each ambiguous word, POS(parts-of-speech) where N indicates Noun, V indicates Verb, Pre indicates preposition and Adj as Adjective. Precision, Recall, F-measure respectively.

**Table 4. Result Analysis**

| Sl NO | Ambiguous word(below-Transliterated form) | NOS | POS | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|
| 1 | কল (kol) | 2 | N | 0.67 | 0.64 | 0.65 |
| 2 | উত্তৰ (uttar) | 2 | N | 0.27 | 0.29 | 0.27 |
| 3 | আদি (aadi) | 2 | N, Pre | 0.29 | 0.29 | 0.29 |
| 4 | অর্থ (artha) | 2 | N | 0.82 | 0.64 | 0.63 |
| 5 | কবি (kobi) | 2 | N | 0.73 | 0.71 | 0.7 |
| 6 | কলা (kola) | 3 | Adj, N | 0.94 | 0.93 | 0.93 |
| 7 | মূৰ (mur) | 2 | N | 0.61 | 0.53 | 0.57 |
| 8 | ফল (fol) | 2 | N | 0.4 | 0.4 | 0.4 |
| 9 | কালি (kali) | 4 | N | 0.91 | 0.83 | 0.87 |
| 10 | আই (aai) | 5 | N | 0.84 | 0.76 | 0.80 |

From the above table it was found that average F-measure of 10 ambiguous words is .611. Most of the ambiguous words have their Parts Of Speech as Noun. Those words which have more than two senses like “কলা (kola)”, “কালি (kali)” Decision Tree have a high precision recall score.

## 8. CONCLUSION

In this paper we proposed a supervised approach- Decision Tree model for Assamese lexical semantic disambiguation task. A classifier algorithm learns from a training sample made up of database associated with their sense labels. Say, a tuple A1 in a database D is denoted by attribute vector A1= {a1,a2,.....,an} and is assumed to belong to a class label. Various challenges like Sense-inventory (consisting of ambiguous words only) along with their senses were discovered, sense-annotated data as a training sample was manually prepared. Crossvalidation evaluation was performed and Precision, Recall and F-measure was used as a metric for the proposed WSD task. Assamese is a less computational aware language and WSD task using a supervised approach- Decision Tree with Cross validation evaluation was the first initiative towards Assamese Language. This will provide a remarkable contribution to Natural Language Processing field.

## 9. REFERENCES

[1] Ide, N. and Véronis, J. 1998. Word sense disambiguation: The state of the art. MIT Press *Computational Linguistics Journal*, 24(1):1-40.

[2] Sarmah, J. and Sarma, S.K., Survey on Word Sense Disambiguation: an initiative towards an Indo-Aryan Language. Accepted in IJEM, March 2016, ISSN: 2305-3631 (Print), ISSN:2306-5982 (Online)

[3] Linden, K., Word Sense Discovery and Disambiguation Thesis, PUBLICATION No. 37, 2005. ISSN 0355-7170.

[4] https://en.wikipedia.org/wiki/C4.5_algorithm.

[5] Sarmah, J. and Sarma, S.K., Word Sense Disambiguation for Assamese, Accepted in 6th IEEE IACC 2016, Feb 27-28, ISBN: 978-1-4673-8285-4

[6] Borah, P.P., Talukdar, G., Baruah, A., In Proceedings of IEEE IC3I, 2014, Nov 27-29.Pg: 946-950

[7] Singh, R.L., Ghosh, K., Nongmeikapam, K. and Bandyopadhyay, S., A decision tree based Word Sense Disambiguation System in Manipuri Language. Advanced Computing: An International Journal (ACIJ), Vol.5, No.4, July 2014

[8] Kumar, A.M., Rajendran, S., Soman, PK., Tamil Word Sense Disambiguation using support vector machines with rich features. International Journal of Applied Engineering Research, Research India Publications, Volume 9, Number 20, p.7609-7620 (2014)

[9] Haroon, R.P., "Malayalam Word Sense Disambiguation" In Proceedings of IEEE International Computational Intelligence and Computing Research (ICCIC), 2010.

[10] Sinha, M., Reddy R.M.K., Bhattacharyya, P., Pandey, P., Kashyap,L.,*www.cfilt.iitb.ac.in/**word**net/webhwn/papers/HindiWSD.pdf*

[11] Parameswarappa, S., Target Word Sense Disambiguation system for Kannada language. In Proceedings of 3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011).

[12] Roy, A., Sarkar, S., and Purkayastha, B.S., Knowledge Based Approaches to Nepali Word Sense Disambiguation. International Journal on Natural Language Computing(IJNLC) Vol. 3, No.3, June 2014

[13] Kalita, P. and Barman. AK, Word Sense Disambiguation: A Survey. International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 4 Issue 5 May 2015, Page No. 11743-11748V

[14] Zampieri, M., A supervised Machine Learning Method for Word Sense Disambiguation of Portuguese Nouns, A Project submitted as part of a program of study for the award of MA Natural Language Processing & Human Language Technology, UNIVERSITY OF WOLVERHAMPTON .

[15] Al_Bayaty, B.F.Z., Joshi, S., International Conference on Emerging Trends in Science and Cutting Edge Technology (ICETSCET-2014) EMPIRICAL IMPLEMENTATION DECISION TREE CLASSIFIER TO WSD PROBLEM.

[16] Dai, W., and Ji, W., A MapReduce Implementation of C4.5 Decision Tree Algorithm, International Journal of Database Theory and Application, Vol 7, No 1(2014), pp 49-60

[17] Han, J., Kamber., M., Pei, J., Third Edition Data Mining Concepts and Techniques– Book Published by Morgan Kaufmann Publishers, ISBN: 978-93--80931-91-3

[18] [18Barman. A.K., A Structured Approach for Building Assamese Corpus: Insights, Applications and Challenges. In Proceedings of the 10th Workshop on Asian Language Resources, pages 21–28, COLING 2012, Mumbai, December 2012.

[19] Sarma, S.K., Gogoi, M., Saikia, U., Medhi, R., Foundation and structure of Developing Assamese WordNet. In Proceedings of 5th International Conference of the Global WordNetAssociation(GWC-2010).