International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 03 | Mar 2022 www.irjet.net p-ISSN: 2395-0072

# Malayalam Word Sense Disambiguation using Machine Learning Approach

**Sruthi S[1], Vysakhi K.[2], Siji P.P.[3], Vismaya[4]**

[1]*Department of Computer Applications, Cochin University of Science and Technology, erala, India*

---

**Abstract -** *This paper proposes a Word Sense Disambiguation(WSD) system in Malayalam, using a Support Vector Machine approach. Word sense disambiguation is a major part in the area of Natural language processing. In different contexts, a word may give an absolutely different meaning. The detection of the correct sense of these ambiguous words from their context is known as WSD. The major parts of this work include creation of a WSD dataset in Malayalam and training of an ML model with both stemmed and unstemmed data.*

***Key Words***: Artificial Intelligence; Natural Language Processing; Support Vector Machine; Word Sense Disambiguation; Trigrams'nTags(TnT) tagging; Malayalam Processing

## 1.INTRODUCTION

"Word Sense Disambiguation" is one of the most influential parts in language processing technology. WSD means selecting the sense of a specific word from a set of predefined possibilities based on its context [5]. Hence WSD can be called a classification problem in natural language processing. Due to the increasing growth of technologies and applications, automatic WSD systems are now available for many languages[8,9,10]. But when it comes to Malayalam, the richness in agglutination and morphological operations makes our language processing tedious . It is a challenge to have so many colloquial words in Malayalam and to have the networks of the same word that give many different meanings. Here is an example that shows an example of semantic disambiguation in Malayalam.

1.അവൻ ആ ചോദ്യത്തിന് ഉത്തരം/answer നൽകിയില്ല. (He did not answer that question). In this context the word "ഉത്തരം" means answer.

2.ഉരുക്ക് ഉത്തരങ്ങൾ/beams ബലം ഉള്ളവയാണ്.

(Steel beams are strong. In this context "ഉത്തരം" means a beam for support or attic.

Here the word 'ഉത്തരം' has different meanings in two sentences. The "Malayalam Word Sense Disambiguation" task is important in determining what sense of such words are used in those sentences and how accurate or apt they are in those sentences. Many standard machine learning techniques can be used for resolving this issue. The issues faced in designing an automatic WSD in Malayalam was the lack of a standard corpora. Here three polysemic Malayalam words (which is shown in Table 1) were chosen and a corpora has been constructed containing 300 contexts, specifically for this purpose. Even though some works in Malayalam automatic disambiguation have been conducted, they did not result in the creation of corpora. Naive Bayes, Support Vector Machine, Maximum Entropy, Decision Tree etc. are some among them (Gopal, S ; Haroon, R. P. 2016). In this work, a multi-class linear based support vector machine approach is used. There are many WSD works done in other languages, especially in English,Chinese etc., but very few in Malayalam.

Mainly this paper includes 4 sections:

First section includes related works in this area. Second section presents a brief description about the proposed systems. Third section includes the results and discussion. Finally the fourth section ends the paper with a conclusion.

**Table-1**: Accuracy obtained based on stemming

| **Ambiguous Words** | **Senses** |
|---|---|
| അടി ("Adi") | Slap, Downwards |
| താമസം ("Thamasam") | Delay, Stay |
| ഉത്തരം ("Utharam") | Beam, Answer |

### 1.1 Related Works

Some of the work done related to WSD with different methods in different languages are specified below.

English word sense disambiguation (Pedersen, T., 2000) : This work shows that WSD can be performed by a number of machine learning methods. It presents a corpus based approach that builds a whole naive Bayes classifier, each of which is based on lexical features that represents co-

occurrence of words in different sized windows of context. Tamil word sense disambiguation (Anand Kumar M; Rajendran, S; Soman, K. P, 2014) : In this paper, an SVM based approach is used for word sense disambiguation. The SVM algorithm can classify the context according to different senses of ambiguous words with great accuracy . SVM classifiers predict the correct sense of target words using a set of feature values.

Malayalam word sense disambiguation using maximum entropy model (Jayan, J. P.; Junaida, M. K.;Sherly, E. 2015) : In this work, semi supervised machine learning techniques mainly maximum entropy is used for Malayalam, which shows result, for a set of trained corpus of Malayalam words. Accuracy of WSD depends on the size of the corpus.

Word sense disambiguation using deep neural networks (Calvo, H.; Rocha-Ramirez, A. P.; Moreno-Armendáriz, M. A.;Duchanoy, C. A. ,2019) : This method can be considered as a hybrid between knowledge based and supervised approach. It proposes a general disambiguation method based on English word embedding representation of words and context, along with a diverse comparison method between them, to select a specific meaning.

Unsupervised approach to word sense disambiguation in Malayalam (KP, S. S.; Raj, P. R.;Jayan, V., 2016) : This thesis proposes and implements the WSD based on context similarities, which is an unsupervised method. Based on similarity between the given input text and sense clusters most similar senses are selected as the sense of ambiguous words. Unsupervised algorithms work directly from annotated raw corpora.

## 2. Proposed system

Here, the major steps involved in the Malayalam word sense disambiguation, using SVM classifier, is described. Fig. 1 shows the block diagram of the model.

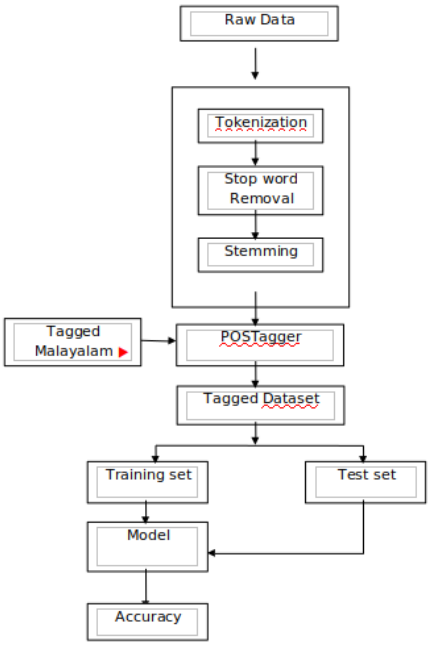


**Fig-1**: Proposed System

Dataset :

Since our language does not have a standard dataset for WSD, a task-specific dataset for this purpose was created. The dataset for this experiment consists of 300 sentences which contains 100 instances of each of the ambiguous word താമസം, അടി, ഉത്തരം. The dataset is balanced with 50 instances for each sense. It has been preprocessed and manually tagged with its meaning and correct form of the ambiguous word.

A. PREPROCESSING

Initially some preprocessing steps are performed to clean the collected data. There are three major steps involved in the data preprocessing.

Tokenization

The collected data is converted into sentences, and then to word tokens

E.g.: "അടി കൊണ്ട ഉടനെ പശു അവിടെ വീണു ചത്തു." is converted into "അടി", "കൊണ്ട", "ഉടനെ", "പശു", "അവിടെ", "വീണു", "ചത്തു".

Stop word removal

Stop words are the words that are frequently occurred among the data, which do not really contribute to the meaning of the discourse

E.g.: "ഇവ", "അത്", "ഇത്", "അവിടെ", "അവ", "എന്ന്", "അന്ന്", etc.

In order to remove stop words from the word tokens, a list of such words was developed.

Stemming

After removing the stop words, the resultant words are stemmed into their root words.

E.g. : "ഓടുക", "ഓടി", "ഓടുന്നു", "ഓടിയില്ല", "ഓടിക്കൊണ്ടിരിക്കുന്നു".

B. POS TAGGING

To convert the cleaned data into useful dataset, a corresponding POS tag is attached to it. The POS tagging is important because, as the meaning of a word changes with the context, the POS tag of that particular word may also change[4].

E.g.: അടിക്കുക: /V_VM_VINF

TnT tagger is used for this purpose, which can be used for training corpus from a number of languages. A sample of tagged Malayalam corpus is used to train the TnT tagger. Then the preprocessed data is tagged using the trained TnT tagger and the result is stored in a csv file along with the ambiguous word, label, and sense of that ambiguous word. Labeling is performed by numbering according to the sense, and hence the dataset. Dataset required for the supervised Malayalam word sense disambiguation using SVM classifier, is stored in the form of a csv file as tokens , sentence by sentence. Dataset of each ambiguous word is stored in separate csv files. Figure 2 shows the tagged dataset obtained after POS tagging with TnT tagger.

```
sentence,ambigous_word,label,sense
"['സ്ഥിരമായ/JJ',
'വീടുകളില്ലാത്ത/JJ',
'മലക്കാരന്മാരുടെ/N_NN',
'താമസം/N_NN',
'പാറപ്പുറത്തുകളിലും/N_NN',
'മരപ്പൊത്തുകളിലുമൊക്കെയാണ്/N_NN']",താമസം,1,
താമസിക്കുക
```

**Fig-2**. POS tagged dataset

C. VECTORIZATION AND IMPLEMENTATION OF SVM CLASSIFIER

The dataset containing text along with POS tags is converted into corresponding feature vectors in order to make the machine understand. A sample converted vector format is given below in Figure3. Count vectorizer from the scikit-learn library was used for vectorization purposes. To implement SVM, the dataset is splitted into a training set and test set; then the model is trained with the training set and predicts the labels of the test set. After comparing the predicted result to the actual result, the accuracy is determined. During training we have used both stemmed data and unstemmed data. The unstemmed data results in better accuracy.

(0, 122) 0.17213244188373364

(0, 228) 0.34090901536599605

(0, 136) 0.2911564266639601

Figure : 3. List of vectors generated after vectorization

## 3. Results and discussion

We had collected more than 100 instances of each of the ambiguous word താമസം, അടി, ഉത്തരം. In total the dataset contains 350 contexts from each of these words. It has been preprocessed and manually tagged with its meaning and correct form of the ambiguous word.

The accuracy of our model for each word under varying conditions, is given in two tables. Table1. gives the accuracy for both stemmed and unstemmed data separately. It is observed that unstemmed data gives more accurate results than stemmed data. We found out that this was because of the inaccuracies in stemming. The root-pack package was used for stemming in Malayalam. Datasets with varying training sizes were also used to study the performance of the system.The accuracy of the model with varying data size is shown in Table 2, with contexts in the training dataset as 50,75 and 100 and test dataset as 10,20 and 25. The graphical representation of each of these words with varying training data size is shown in figure 4, figure 5 and figure 6. It is very clear that accuracy of model increases with increase in data size except for the word അടി.

**Table-2**: Accuracy obtained based on stemming

| Words | With-out stemming | | | With stemming | | |
|---|---|---|---|---|---|---|
| Amount of data | 50 | 75 | 100 | 50 | 75 | 100 |
| "Thaamasam" | 78.2 | 85.7 | 82.6 | 69.5 | 77.1 | 65.2 |
| "Adi" | 60.89 | 65.7 | 67.39 | 65.21 | 68.57 | 69.56 |
| "Utharam" | 91.30 | 100 | 100 | 73.9 | 71.42 | 82.60 |

**Table-3:** Accuracy obtained based on the size of data

| Ambiguous Words | Unstemmed data | Stemmed data |
|---|---|---|
| "Thamasam" | 82.60 | 65.21 |
| "Adi" | 67.39 | 69.56 |
| "Utharam" | 100 | 82.60 |

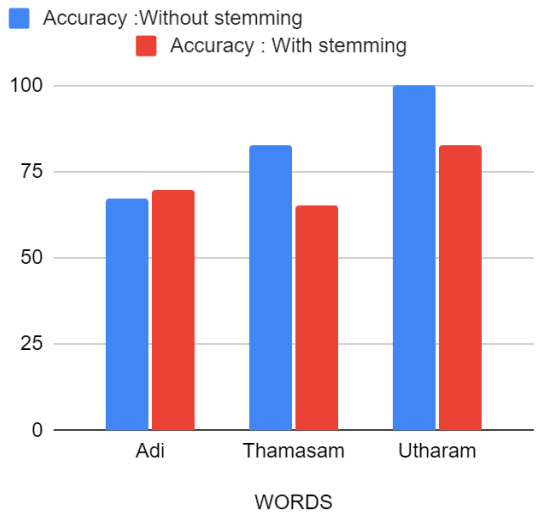


**Fig-4**. WSD Comparison between ambiguous words when training data size=50

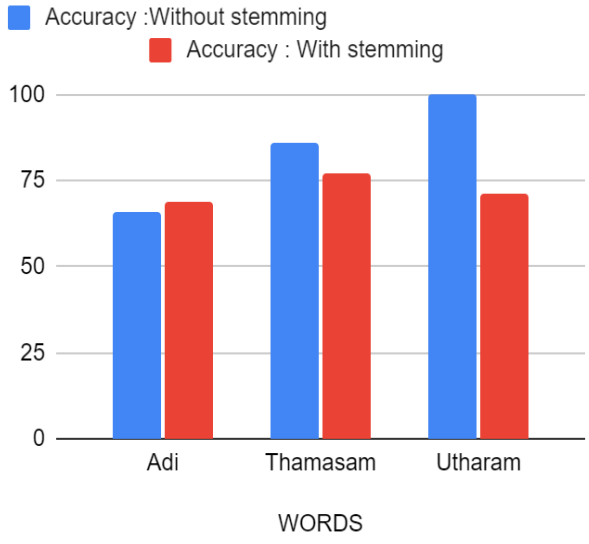


**Fig-5** . WSD Comparison between ambiguous words when training data size=75

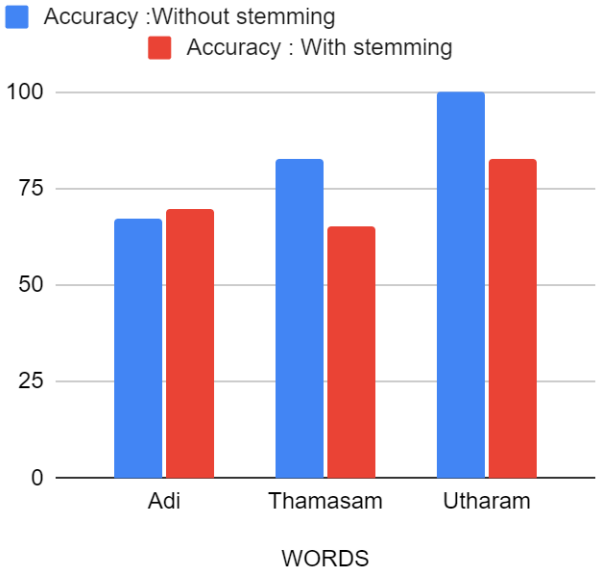


**Fig-6** . WSD Comparison between ambiguous words when training data size=100

## 4. CONCLUSIONS

The automatic WSD techniques are now abundantly used in English and many other languages. But, WSD techniques for Malayalam are very rare because of its complexity. So our paper is relevant in this scenario. The accuracy of this model depends upon some factors like the size of corpus and preprocessing techniques. From our observations it is clear that when more data is provided, a more efficient model could be created. It is also observed with better accuracy when an unstemmed dataset is used. We are in the process of creating a standard dataset for WSD in Malayalam, so that upcoming researchers in this field could also be benefitted. With a larger dataset in hand, we can use deep transformer based pre-trained models like BERT, GPT-3, fastext etc. for generating contextualized embeddings, which could result in better performance of the system.

## REFERENCES

[1] Anand Kumar, M., Rajendran, S., & Soman, K. P. (2014). Tamil word sense disambiguation using support vector machines with rich features. International Journal of Applied Engineering Research, 9(20), 7609-20.

[2] Pedersen, T. (2000). A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. arXiv preprint cs/0005006.

[3] Jayan, J. P., Junaida, M. K., & Sherly, E. (2015) Malayalam Word Sense Disambiguation using Maximum

Entropy Model.

[4] Junaida, M. K., Jayan, J. P., & Elizabeth, S. (2015, December). Malayalam Word Sense Disambiguation using Yamcha. In 2015 International Conference on Computing and Network Communications (CoCoNet) (pp. 720-724). IEEE.

[5] Gopal, S., & Haroon, R. P. (2016, March). Malayalam word sense disambiguation using Naïve Bayes classifier. In 2016 International Conference on Advances in Human Machine Interaction (HMI) (pp. 1-4). IEEE.

[6] KP, S. S., Raj, P. R., & Jayan, V. (2016). Unsupervised approach to word sense disambiguation in malayalam. Procedia Technology, 24, 1507-1513.

[7] Calvo, H., Rocha-Ramirez, A. P., Moreno-Armendáriz, M. A., & Duchanoy, C. A. (2019). Toward Universal Word Sense Disambiguation Using Deep Neural Networks. IEEE Access, 7, 60264-602

[8] Navigli, R. (2009). Word sense disambiguation: A survey. ACM computing surveys (CSUR), 41(2), 1-69.

[9] Zhou, X., & Han, H. (2005, May). Survey of Word Sense Disambiguation Approaches. In FLAIRS conference (pp. 307-313).

[10] Pal, A. R., & Saha, D. (2015). Word sense disambiguation: A survey. arXiv preprint arXiv:1508.01346.