SENTIMENT ANALYSIS ON HINDI REVIEWS

UNDER THE GUIDANCE OF

DR. L. PRATAP REDDY

PROFESSOR IN ECE DEPARTMENT

PRESENTED BY:

SAI PRANATHI D 19015A0403

RAGHU VAMSI S 18011A0444

BHARATH K 18011A0422

ABHISHEK J 19015A0405

CONTENTS

INTRODUCTION

- Sentiments are hidden behind online comments on social media of all kinds.
- Sentiment analysis is the natural language processing task.
- It helps to identify and categorize opinions expressed in a piece of text, determine the reviewer’s point of view on a particular topic.

NEED FOR SENTIMENT ANALYSIS ON HINDI REVIEWS

- Little work has been done in Sentiment Analysis for Indian Languages.
- Web content in Hindi is booming .
- Sentiment Analysis of movie reviews could help in better rating of movies.

CHALLENGES & AIM

- Hindi is morphologically rich and a free order language compared to English language.
- Hindi is a resource scarce language which causes problems in collection and generation of datasets.
- Limited resources are available for this language like Hindi SentiWordNet (H-SWN).

➢ Aim is to implement the methods that can accurately predict the sentiment(polarity) of a given Hindi movie review.

METHODS IMPLEMENTED

1. Resource based classification

- ✓ using Hindi SentiWordNet (H-SWN).

2. In-language classification through various classifiers.

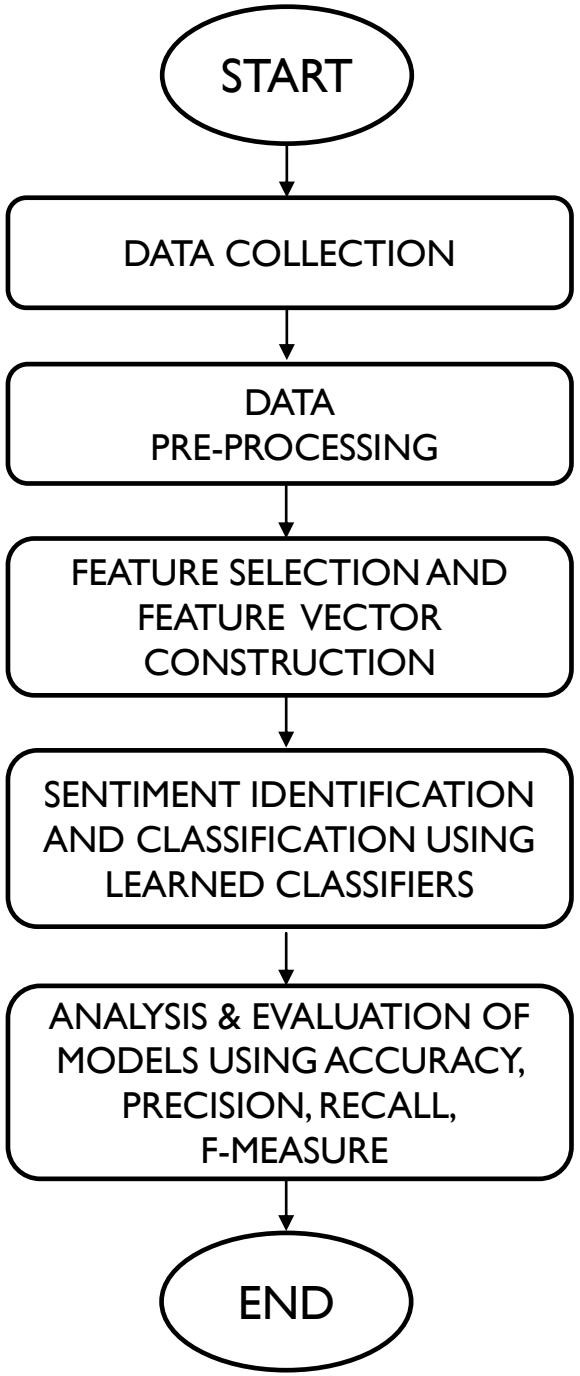
Feature extraction using 2 methods:

- ✓ Unigram Model
- ✓ TF-IDF Model

RESOURCE BASED CLASSIFICATION

- It uses Hindi SentiWordNet (H-SWN).
- Every word available in H-SWN has a corresponding positive and a negative sentiment score.
- A vote is taken for each word in a review.
- The polarity with majority is predicted as the sentiment of a review

FLOW CHART


START
DATA COLLECTION
DATA PRE-PROCESSING
FEATURE SELECTION AND FEATURE VECTOR CONSTRUCTION
SENTIMENT IDENTIFICATION AND CLASSIFICATION USING LEARNED CLASSIFIERS
ANALYSIS & EVALUATION OF MODELS USING ACCURACY, PRECISION, RECALL, F-MEASURE
END

IN-LANGUAGE CLASSIFICATION USING UNIGRAM MODEL

- A lexicon is created that contains all words in dataset except stop words and some highly frequent words that do not affect review sentiment.
- Feature matrix of size m*n

where,

m= number of reviews in our dataset

n= number of words in the lexicon

- For each element of the matrix, if that lexicon word occurs in the review, we add 1 to index of that word in lexicon set.

CLASSIFIERS USED:

We have used various classifiers for our in-language classification model:

- Logistic Regression
- Stochastic Gradient Descent
- Naive Bayes
- Support Vector Machine
- Decision Tree
- Neural Network
- Voting Classifier

IN-LANGUAGE CLASSIFICATION USING TF-IDF MODEL

- TfidfVectorizer() function is available in scikit-learn library.
- It is used to convert a collection of raw documents to a matrix of TF-IDF features.
- TF-IDF score of a word in a review= term frequency* inverse document frequency (idf)

where,

idf= 1/(1+log(no. of reviews in which the word occurs))

EXPERIMENTAL SETUP

Actual Value

Predicted Value	positives	negatives
positives	True Positive(t_p)	False Positive(f_p)
negatives	False Negative(f_n)	True Negative(t_n)

Table 1: Confusion Matrix

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

$$Recall = \frac{t_p}{t_p + f_n}$$

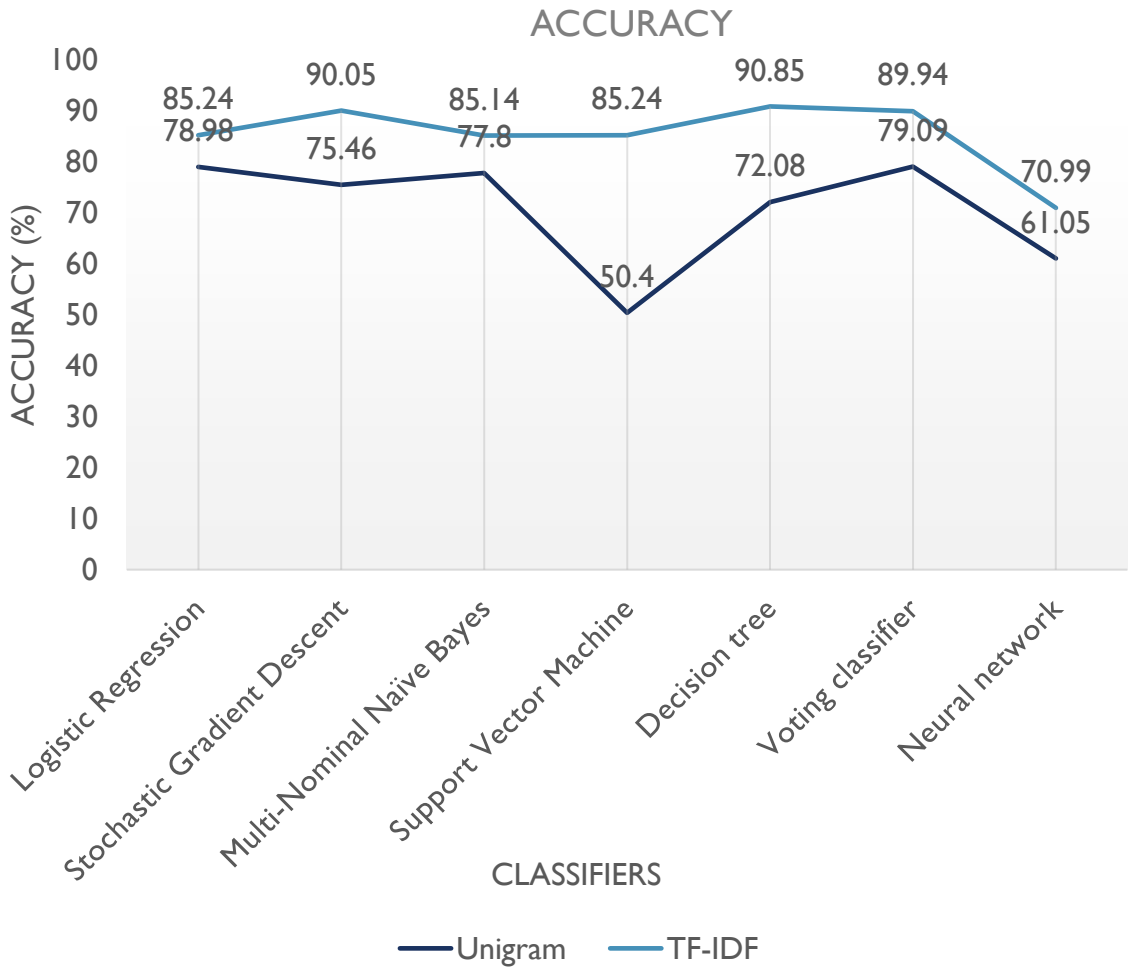
$$F - measure = \frac{2 * Precision * Recall}{Recall + Precision}$$

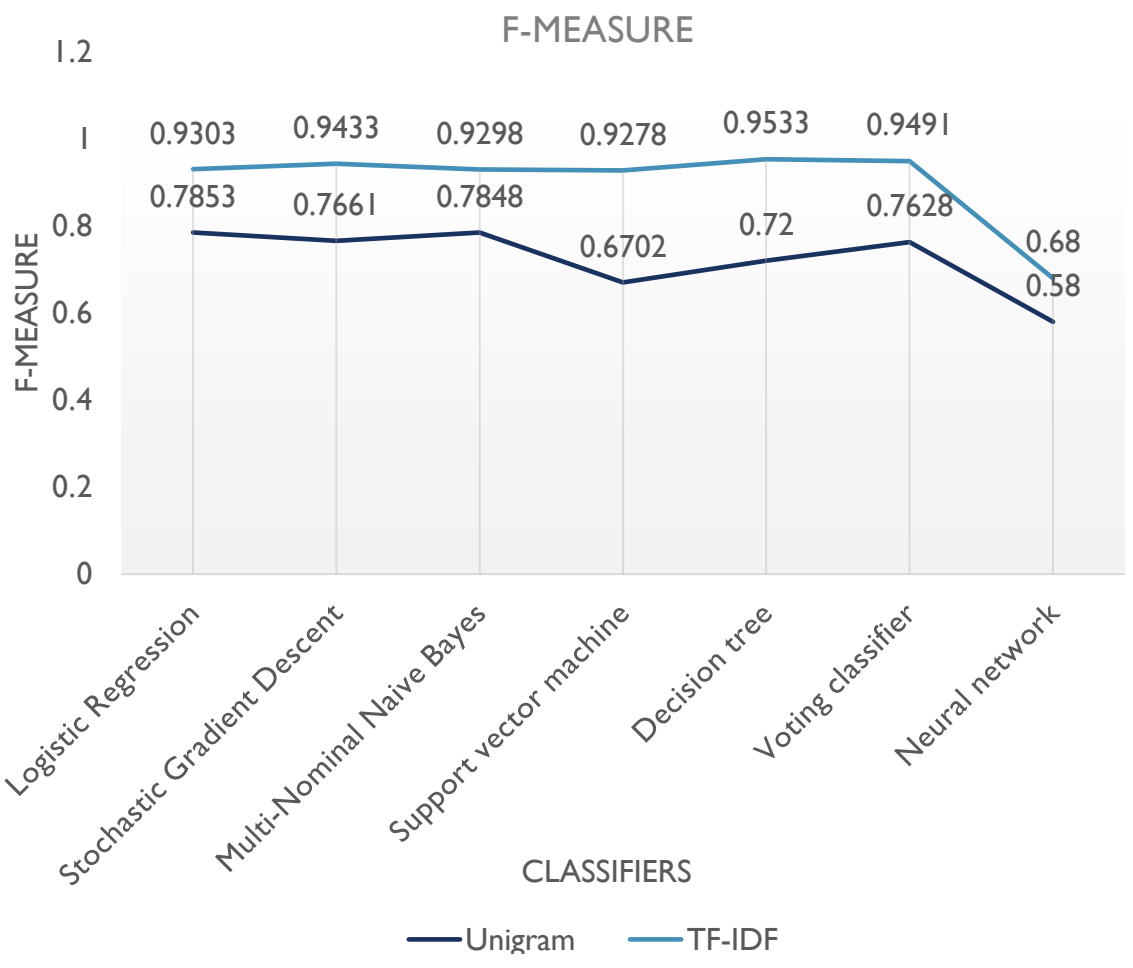
$$Precision = \frac{t_p}{t_p + f_p}$$

RESULTS

S.No	Classifiers Used	Accuracy (%)		F-measure	
		Unigram	TF-IDF	Unigram	TF-IDF
1.	Logistic Regression	78.98	85.24	**0.7853**	0.9303
2.	Stochastic Gradient Descent	75.46	90.05	0.7661	0.9433
3.	Multi-Nominal Naive Bayes	77.8	85.14	0.7848	0.9298
4.	Support Vector Machine	50.4	85.24	0.6702	0.9278
5.	Decision Tree	72.08	**90.85**	0.72	**0.9533**
6.	Voting Classifier	**79.08**	89.94	0.7628	0.9491
7.	Neural Network	61.05	70.99	0.58	0.68

GRAPHICAL ANALYSIS





CONCLUSION & FUTURE SCOPE

- ✓ In-language classification approach performs best than Resource based classification.
- ✓ The TF-IDF method of Feature Matrix Generation gives better results than Unigram Model.
- Resource-based Sentiment Analysis Approach could be extended to include Word Sense Disambiguation and lexical chaining approach to get better results.
- Hindi SentiWordNet (H-SWN) currently contains limited words. Improving dictionary can give much better predictions.
- Present models implemented do not support negation rules.

REFERENCES

1. MEDHAT, WALAA. AHMED HASSAN, and HODA KORASHY. “Sentiment analysis algorithms and applications: A survey.” *Ain Shams engineering journal 5.4* (2014), pp. 1093-1113.
2. ARORA P. “Sentiment analysis for Hindi language.” *MS by Research in Computer Science* (2013).
3. BAKLIWAL A, ARORA P, and VARMA V. “Hindi subjective lexicon: A lexical resource for Hindi polarity classification.” In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)* (2012), pp.1189–1196.
4. BANSAL N, AHMED U.Z, and MUKHERJEE A. “Sentiment analysis in Hindi.” *Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India* (2013), pp. 1-10.
5. MITTAL N, AGARWAL B, VHOUHAN G, PAREEK P and BANIA N. “Discourse based sentiment analysis for Hindi reviews.” In *International Conference on Pattern Recognition and Machine Intelligence* (2013), Springer, pp.720-725
6. PEDREGOSA F, VAROQUAUX G, GRAMFORT A, MICHEL V, THIRION B, GRISEL O, BLONDEL M, PRETTENNHOFER P, WEISS R, DUBOURG V, VANDERPLAS J, PASSOS A, COURNAPEAU D, BRUCHER M, PERROT M and DUCHESNAY E. “Scikit-learn: Machine learning in Python.” *Journal of Machine Learning Research 12* (2011), pp. 2825-2830.

THANK YOU