Feature based Entailment Recognition for Malayalam Language Texts

Sara Renjit
Department of Computer Science
Cochin University of Science and Technology
Kerala, India

Sumam Mary Idicula
Department of Computer Science
Muthoot Institute of Technology and Science
Kerala, India

***Abstract*—Textual entailment is a relationship between two text fragments, namely, text/premise and hypothesis. It has applications in question answering systems, multi-document summarization, information retrieval systems, and social network analysis. In the era of the digital world, recognizing semantic variability is important in understanding inferences in texts. The texts are either in the form of sentences, posts, tweets, or user experiences. Hence understanding inferences from customer experiences helps companies in customer segmentation. The availability of digital information is ever-growing with textual data in almost all languages, including low resource languages. This work deals with various machine learning approaches applied to textual entailment recognition or natural language inference for Malayalam, a South Indian low resource language. A performance-based analysis using machine learning classification techniques such as Logistic Regression, Decision Tree, Support Vector Machine, Random Forest, AdaBoost, and Naive Bayes is done for the MaNLI (Malayalam Natural Language Inference) dataset. Different lexical and surface-level features are used for this binary and multiclass classification. With the increasing size of the dataset, there is a drop in the performance of feature-based classification. A comparison of feature-based models with deep learning approaches highlights this inference. The main focus here is the feature-based analysis with 14 different features and its comparison, essential to any NLP classification problem.**

Keywords*—Textual entailment; natural language inference; Malayalam language; machine learning; deep learning

I. Introduction

Textual entailment (TE), also called natural language inference (NLI) is a relationship between a pair of sentences. It identifies the similarity between the sentences based on their inferential semantic content. A text is said to entail another sentence, called a hypothesis, if the hypothesis has its semantic content derived from the text. In the same way, the text contradicts the hypothesis if the semantic content of the hypothetical sentence is just opposite to the text. Both sentences remain neutral to each other if the hypothesis derives zero information from the text.

A classical definition for entailment is that a text t entails hypothesis h if h is true in every circumstance of a possible world in which t is true. This definition is too strict in applying to real-world applications. An applied definition says that text t entails hypothesis h if human reading t infers that h is most likely true. Mathematically computable definition for text entailment is provided as hypothesis h is entailed by text t if P(h is true $|$t) $>$ P(h is true), where P(h is true $|$t) is the Entailment Confidence [1].

Semantic variability in expressions is an essential factor in any natural language processing application. NLI is also a necessary sub-task for almost all NLP applications such as multi-document summarization, question answering systems, information extraction, information retrieval. In multi-document summarization, the redundant sentences are identified using entailments, and those sentences can be removed. The answer to a question can be evaluated based on its entailment to the reference answer in the question answering system. In information extraction and retrieval systems, the text should entail the extracted information.

Natural language inference also finds application in analysis of user tweets, posts and experiences in social networks, where people share their thoughts, experiences in the form of texts in various languages. These texts are useful to relate between users by analysing inferences (entailment, contradiction and neutral) between the texts. This helps in customer segmentation, product analysis from the customer viewpoint as well as in recommender systems.

As information is available in digital text form in almost all languages, recognizing entailment is important for almost all languages. Text entailment is recognized in various languages, namely, English, French, Spanish, Italian, Japanese, Hindi, Swahili, Urdu. Very few works are reported for the Malayalam language.

In this work, we classify entailments for Malayalam, a South-Asian language from the Dravidian family. Malayalam is the language officially used and spoken in the state of Kerala. This language has its origin from the Dravidian scripts of Tamil. The language has various dialects, agglutinations, and inflectional word forms used in different parts of the state. This language also has very few resources in terms of datasets and other language processing applications and falls in the class of low resource languages.

The main contributions in this work includes:

1) The application of machine learning methods for Malayalam language textual entailment recognition, which is not attempted so far and also required for current literature and future research in this area.

2) A comparison between machine learning and deep learning approaches for Malayalam language entailment recognition.

3) The limitations of feature based methods with increasing dataset size.

4) An inference that deep learning without explicit feature-based engineering helped in more accurate classification for datasets of larger size.

The rest of the article is organized as follows: Section II describes the related literature in English and other languages. Challenges and contributions in Malayalam language for entailment is provided in Section III. Textual entailment for the Malayalam language using feature set is detailed in Section IV. The experimental evaluations are in Section V. Section VI discuss the results and Section VII concludes the work.

II. Related Work

Textual entailment has its inception in 2005 as PASCAL (Pattern Analysis, Statistical Modelling, and Computational Learning) challenge programme 'Recognizing Textual Entailment (RTE)' to develop systems that can recognize inferences from text fragments across various applications like multi-document summarization, information retrieval, information extraction and question answering systems.

In 2008, PASCAL RTE became a track at the Text Analysis Conference organized by NIST (National Institute of Standards and Technology), which brought different NLP communities to work on the textual entailment application scenarios. The earliest approaches for determining textual entailment include bag of words, logic-based reasoning, lexical entailment, machine learning methods, and graph matching [2].

The English language: The challenge started for the English language, and all major works are implemented in English language using RTE(Recognizing Textual Entailment), SNLI (Stanford Natural Language Inference) [3], MNLI (Multi-genre Natural Language Inference) [4] and XNLI (Cross-lingual Natural Language Inference) datasets. Lexical and syntactic similarity based entailment classification is done using rule-based similarity features such as unigram, skip-gram, longest common subsequence, stemming, subject-subject comparison, subject-verb, object-verb comparison [5].

RTE datasets were used to train and test these systems. Entailment recognition is also attempted by resolving anaphoras in sentence pairs [6]. [7] does similarity metrics-based recognition of entailments in the text, where features like cosine similarity, unigram match, Jaccard similarity, dice similarity, overlap, harmonic mean, and machine translation evaluation metrics, namely BLEU and METEOR, are used for machine learning. Following are the other approaches:

Bag of Words: In this approach, both text and hypothesis are represented as a collection of words. Every word from the hypothesis collection is compared with every word from the text collection. If the match between T and H is more than a preset threshold, then the sentence pair is classified as entailment, else, not entailment. It ignores the word order, syntax, and semantics of the sentences.

Lexical Entailment: Entailment is determined based on lexical concepts. A hypothesis is valid if its lexical components are true [1]. It is based on a probabilistic model and does not consider syntax and semantics.

Machine Learning approaches: Linear classifiers, logistic regression, support vector machines are classifiers used to train and learn from a dataset of text hypothesis pairs. It is a feature-based approach using similarity measures on words, stems, POS tags, chunk tags, negation, length ratio, of best partial match [8].

Graph based approaches: Text and hypothesis can be represented as directed graphs (dependency graphs), nodes representing words or phrases, and edges representing the relation between nodes [9]. Entailment is determined in these graphs using a matching cost based on vertex substitution and path substitution.

Deep learning approaches: The entailment recognition attempts in English from 2005 to 2015 are either rule-based or feature-based machine learning approaches. With the introduction of the SNLI dataset in 2015, a large dataset has enabled deep techniques for sentence representation using LSTM (Long Short Term Memory), CNN (Convolutional Neural Network) [10], BERT [11], and other transformer models and classification through deep neural networks [12]. Textual entailment is also used for fake news detection [13].

a) Datasets for Textual Entailment in English: The current works are mainly carried out in datasets, namely, RTE, SNLI, MNLI, and XNLI and in legal texts [14]. The collection of RTE datasets with their specifications are mentioned in the Table I.

TABLE I. RTE Datasets [15]

Dataset	size	Specification
RTE1	1367	manually collected pairs
RTE2	800	more realistic examples
RTE3	800	more longer texts
RTE4	1000	3 way classification (Entailment, Contradiction and Unknown)
RTE5	600	unedited texts
RTE6	15955	221 hypothesis
RTE7	21420	longer texts.

Other NLI datasets are SNLI (Stanford Natural Language Inference) dataset which is a collection of 570k English sentence pairs collected using Amazon mechanical trunk [3], and MNLI, Multi-Genre Natural Language Inference dataset is a collection of 433k sentence pairs from multiple genres [4].

b) Other languages: Entailment recognition in Japanese, Simplified Chinese, and Traditional Chinese language is attempted with RITE (Recognizing Inference in Texts) dataset [16], which has forward entailment, reverse entailment, bidirectional entailment, contradiction, independence as different classes for the Chinese sentence pairs. Surface textual features, lexical-semantic feature, syntactic feature, linguistic feature are used for classification using an SVM model [17].

Italian dataset is used in EVALITA campaign 2009 to recognize entailments in Italian text pairs [18]. Arabic dataset for textual entailment is detailed in [19]. Traditional features and distributed representations are used for recognizing textual entailment in Arabic [20]. Cross-lingual natural language inference dataset (XNLI) derives its collection from MNLI dataset and contains translated pairs in 15 languages, namely French, Spanish, German, Greek, Bulgarian, Russian, Turkish, Arabic,

Vietnamese, Thai, Chinese, Hindi, Swahili, and Urdu, out of which Hindi, Swahili, and Urdu are Indian languages [21]. Textual entailment for Indo-Aryan languages like Hindi is important to the language community of Northern parts of India. In this attempt we focus on Malayalam language from the Dravidian family. The Dravidian languages are mostly spoken in southern parts of India and has very minimal contributions when considering inferences. Attempts to different families of languages helps to gather significant contributions which are specific to those languages or language family and generic to all languages.

III. Challenges and Contributions

The Malayalam language is a South Indian Dravidian language, which has minimal works for textual entailment. The automatic and manual translation of SNLI pairs with linguistic corrections by experts forms the basis for the MaNLI (Malayalam Natural Language Inference) dataset. Prior work in Malayalam textual entailment reports the use of different embedding techniques, namely, Doc2Vec (paragraph vector), fastText, BERT(Bidirectional Encoder Representations from Transformers) and LASER(Language Agnostic SEntence Representations) for embedding sentence pairs for classification through Densenet [22]. Another attempt use siamese networks for binary classification of inference in texts [23]. The accuracy measure in Table II shows that LASER embedding based classification achieves the best results.

TABLE II. Performance of Different Embedding Methods for NLI in Malayalam

Embedding method	Binary	Multiclass
Doc2Vec	0.58	0.49
fastText	0.68	0.52
BERT	0.66	0.5
LASER	0.77	0.64

A. MaNLI Dataset

The development of language resources for Malayalam is in a progressing stage by different organizations and individual contributors. The Malayalam Natural Language Inference dataset is a dataset developed for natural language inference in the Malayalam language. It is created by manual and machine translation of text hypothesis pairs from the SNLI (Standford Natural Language Inference) dataset. Certain incorrect translations were corrected through manual efforts. Olam dictionary [24] is also used to get common substitutes for the English words. The details of the dataset are in Table III.

TABLE III. Dataset Statistics

MaNLI dataset	
Total sentence pairs	12000
Entailment pairs	4026
Contradiction pairs	3963
Neutral pairs	4011
Unique words	16194
Avg. premise sentence length	9.17
Avg.hypothesis sentence length	5.04

The dataset is created because an adequately annotated and linguistically correct entailment dataset is unavailable in this language. Hence, the translation method with linguistic corrections from language experts is adopted as one method to produce this dataset. This method involves less time and cost than creating an entirely new dataset that requires more time and human involvement to create sentence pairs and annotations.

The MaNLI dataset [22] [25] is a collection of 12K text-hypothesis pairs classified into entailment, contradiction, and neutral. Translations are done in such a way that the semantic content is maintained the same. Hence the annotated class labels are reused. It has been manually verified by linguists from the Thunchath Ezhuthachan Malayalam University, Kerala. The sentence length distribution for text and hypothesis sentences from the corpus is shown in Fig. 1. The premise sentences have word length between 5 and 15 whereas the hypothesis word length varies between 5 and 10 for most cases.

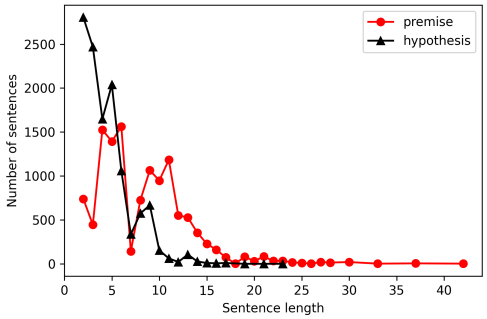

Fig. 1. Sentence Length Distribution

IV. Proposed Method

Textual entailment or natural language inference in English is attempted using machine learning and also deep learning approaches. But feature based machine learning approaches are not reported for the Malayalam language. In this work, we aim to develop systems for the Malayalam language using feature-based machine learning methods, which is essential to understand any classification problem. Also, comparison of feature-based models with deep learning methods became more feasible and realistic.

The design of the proposed work is shown in Fig. 2. Input pairs of text and hypothesis are preprocessed, and various lexical, semantic, and set-based features are extracted. The machine learning module classifies the text hypothesis pairs based on the extracted features using ML algorithms, such as Logistic regression, Support Vector Machine, Decision tree, Random Forest, Multinomial Naive Bayes and Adaboost.

A. Preprocessing

The sentence pairs are split into tokens, and prefixes and suffixes are removed in the preprocessing stage through tokenization and stemming. Tokenization is the process by which the words in the sentences are split into individual units called tokens for processing. The splitting is done using space as a separator. Stemming removes affixes from words.

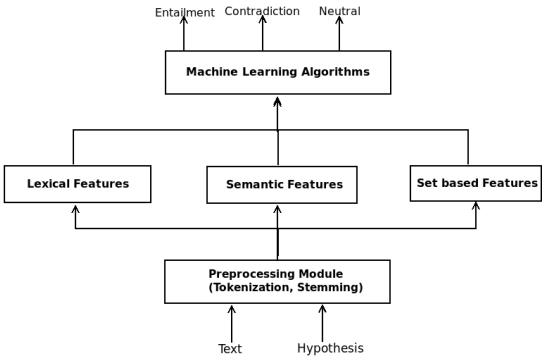

Fig. 2. Design of Textual Entailment Recognition System using Feature based Machine Learning.

For example, the word 'flowers' can have its stem word as 'flower', removing the suffix 's'. For the Malayalam language, libindic stemmer [26] available online is used. It is a rule-based stemmer using iterative suffix stripping to handle inflectional words.

B. Feature based Classification

This section details the different features used for the entailment classification. The features fall into different categories, namely lexical features, semantic features, and set-based features.

Lexical features: Lexical features are word or surface-level features that deal with the overlap of words. The different lexical features used are:

1) Unigram match: Overlap score of unigrams in text and hypothesis is computed. The unigram match score for text is defined as the number of unigram overlaps by the total number of unigrams in text. The Unigram match score for the hypothesis is defined as the number of unigram overlaps by the total number of unigrams in the hypothesis.

2) Bigram match: It is the number of bigram overlap divided by the number of bigrams in hypothesis.

3) Longest Common Subsequence: The LCS match is calculated as the length of the longest common subsequence/length of hypothesis.

4) Skip gram match: Skip-gram is a combination of n words in the sentence with few gaps. Skip grams with degree 2 and skip distance 1 are found for text and hypothesis. These skip grams matched count is divided by the number of skip grams in the hypothesis.

5) Length features: This consists of different length measures, such as $|B-A|$, $|A\&B|$, $(|B|-|A|)/|A|$, $(|A|-|B|)/|B|$, $|A\&B|/|B|$ where A is the text and B is the hypothesis.

Semantic features: Semantic features deal with the semantics of the sentences. For this, we have used word vector representation and term frequency-inverse document frequency (TF-IDF) of sentences.

1) Word embedding similarity: Word vectors from Word2Vec [27] are used to represent the words. Summation of word vectors of a sentence (text/hypothesis) is computed, and cosine between the two gives a similarity feature value.
2) TF-IDF similarity: Term Frequency -Inverse Document Frequency (TF-IDF) is a numerical statistic that evaluates the importance of a word in a collection. It is the product of term frequency and inverse document frequency. Text and hypothesis are represented using TF-IDF representation and cosine similarity is computed between the two.

Set/Distance based measures: Set/Distance based measures are the different types of similarities using counts for set-based unions and intersections. The various set-based similarities are:

1) Dice similarity: It measures the spatial overlap between two sentence pairs.

$$Dice(X,Y) = \frac{2\,\|X \cap Y\|}{\|X\| + \|Y\|} \tag{1}$$

If X and Y are similar, Dice coefficient will be 1 and otherwise 0.

2) Cosine similarity: It measures the cosine of the angle between the two sentences.

$$Cosine(X,Y) = \frac{\|X \cap Y\|}{\sqrt{\|X\|.\|Y\|}} \tag{2}$$

3) Levenstein similarity: It measures the minimum number of insertions, deletions and substitutions required to transform one word to another.

4) NeedleWunsch similarity: It is a sequence alignment based similarity measure. It measure global alignment score by finding the no of edits required which is calculated from the alignment matrix.

5) Smith Watermann similarity: It is a dynamic programming method that uses local alignment as a metric to measure similarity. The alignment matrix is created with no negatives and the scores are calculated.

6) Jaro Winkler similarity: It is also a string metric that measures the edit distance between two sequences from beginning to a set of prefix length.

$$sim = sim_{\mathrm{j}} + lp(1 - sim_{\mathrm{j}}) \tag{3}$$

where sim_{j} is the Jaro similarity between strings s_1 and s_2, l is the prefix length, p =0.1 (constant scaling factor).

$$sim_{\mathrm{j}} = \begin{cases} 0 \; if \; m = 0 \\ \frac{1}{3}(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}) \end{cases} \tag{4}$$

where $|s|$ is the string length, m = no of matching characters, t = no of transpositions.

7) Jaccard similarity: This metric has the ratio of similarity and dissimilarity of sample sequences.

$$Jaccard(X,Y) = \frac{\|X \cap Y\|}{\|X \cup Y\|} \quad (5)$$

C. Machine Learning Approaches

Inference in the Malayalam language is considered as binary and multiclass classification. Binary classes are entailment and contradiction. Multiclass includes entailment, contradiction, and neutral. The following machine learning algorithms are used to evaluate the performance.

1) Logistic Regression: Logistic regression has dependent variable in two classes. With two classes x1 and x2 and the binary response variable Y (p= P(Y=1)),

$$logistic\ regression,\ l = log_b \frac{p}{1-p} \quad (6)$$

Binary classification is done with liblinear solver and class weight is balanced. Multinomial logistic regression is used to predict the different possible outcomes of a categorically distributed dependent variable. The classifier with multinomial class weights and lbfgs solver is used for multiclass classification.

2) Support Vector Machine: SVM maps the training examples to points in n-dimensional space. For binary classification, it maps into a 2-D plane separated by a line, and samples are mapped into either of the side of the plane. For multiclass classification, the samples are separated into different categories by a hyperplane.

3) Random Forest: It is an ensemble learning method which constructs many decision trees at training. For classification task, output class is the class selected by majority of the trees.

4) Decision Tree: It has a predictive modeling approach, start of the tree has different observations, that it traverse through the branches and ends in leaf nodes belonging to the target category for the sentence pair.

5) MultinomialNB: It is a Naive Bayes classifier for multi class classification. The feature vector consists of frequencies or integer counts. It is based on the Bayes' theorem stated below: P(c |x) = P(x |c) * P(c) / P(x) where c is a class and x is the sample instance that is to be classified.

6) AdaBoostClassifier: Also called adaptive boosting, it consists of weak classifiers in which one of the classifier is used to train on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

V. Experimental Settings and Evaluation

Implementations are done in Spyder integrated environment. The libraries used are Libindic stemmer for stemming, NLTK toolkit for extracting bigrams, text distance library for evaluating the distance between two or more sequences, and Scikit Learn for machine learning algorithms and classification reports. Grid searchCV is used for SVM classification. Table IV shows the specific settings applied in Scikit Learn based classifiers.

TABLE IV. Experimental Settings for LR, SVM and Random Forest

Model	Settings
Logistic Regression	solver= liblinear, class weight=balanced (Binary)
Logistic Regression	solver= lbfgs, multiclass=multinomial (Multiclass)
Support Vector Machine	kernel: ovr, rbf, C:(1, 10), gamma: (1, 0.1, 0.01, 0.001, 0.0001)
Random Forest	no of estimators=100, max depth=5

We have used different combinations of the feature set to arrive at the results. The different feature set configurations are in Table V.

TABLE V. Different Feature Set Configurations based on Combination of Features.

Feature set	Features
F1	Lexical (L)
F2	Semantic (S)
F3	Distance (D)
F4	Lexical, Semantic (L,S)
F5	Lexical, Distance (L,D)
F6	Semantic, Distance (S,D)
F7	Lexical, Semantic, Distance (L,S,D)

Evaluation Metrics The classification performance is evaluated using the Scikit-Learn classification metrics namely accuracy, precision, recall and F1-score.

- Accuracy: Accuracy is defined as the ratio of number of correct predictions to total predictions. Accuracy $=(tp+tn)/(tp+fp+fn+tn)$
- Precision: Precision is defined as the ability of the classifier not to misclassify samples (label negative sample as positive). Precision $= tp/(tp+fp)$
- Recall: Recall is defined as the ability of the classifier to find all positive samples. Recall $= tp/(tp+fn)$
- F1-score: F1-score is the harmonic mean of precision and recall. F1-score $= 2.precision.recall/(precision+recall)$

where tp is true positive, fp is false positive, tn is true negative and fn is false negative.

TABLE VI. BINARY CLASSIFICATION RESULTS IN TERMS OF WEIGHTED AVERAGE ACCURACY, PRECISION, RECALL, F1-SCORE AND SUPPORT

Model	Accuracy	Precision	Recall	F1-score	Support
LR	0.66	0.66	0.66	0.66	1598
SVM	0.67	0.67	0.67	0.67	1598
RF	0.67	0.67	0.67	0.67	1598
DT	0.66	0.66	0.66	0.65	1598
MNB	0.62	0.62	0.62	0.61	1598
AdaBoost(AB)	0.66	0.66	0.66	0.66	1598

TABLE VII. MULTICLASS CLASSIFICATION RESULTS IN TERMS OF WEIGHTED AVERAGE ACCURACY, PRECISION, RECALL, F1-SCORE AND SUPPORT

Model	Accuracy	Precision	Recall	F1-score	Support
LR	0.48	0.48	0.48	0.48	2400
SVM	0.50	0.50	0.50	0.50	2400
RF	0.49	0.49	0.49	0.48	2400
DT	0.46	0.46	0.46	0.46	2400
MNB	0.42	0.42	0.42	0.42	2400
AdaBoost(AB)	0.49	0.49	0.49	0.49	2400

The results of the classification evaluated in terms of accuracy, precision, recall, and F1-score is shown in Table VI with the whole 7989 pairs for binary classification and Table VII with 12k pairs for multiclass classification with the feature set configuration F7 having all the features. The train test split is 80:20. The performance of the rest of the feature sets (F1 to F6) is low compared to F7, hence we selected the feature set F7 for our study and comparisons. The performance of other feature sets is detailed in Section VI-B. From Tables VI and VII, it can be inferred that SVM, random forest and AdaBoost better classifies the Malayalam texts into entailment, contradiction and neutral classes.

We have evaluated our system with an increasing size of the data ranging from 2000 to 12000. The variation in the performance is shown in Fig. 3 for binary classification. The plot for mutliclass classification is shown in Fig. 4.

VI. RESULTS AND DISCUSSION

A. Effect of Increasing Size of Dataset

This section discusses the difference in the performance of deep learning and feature-based machine learning classification for binary and 3-way classification. As the size of the dataset increases from 2000 to 12k, there is a reduction in performance of feature-based classification. The features selected may be suitable for a few samples, but they can be misleading for other samples. Hence the model is not able to generalize with the samples.

LASER-based approach [22]: In the case of deep learning approaches with embedding that captures the context and places the sentences in semantic space, the model can generalize in a much more efficient manner. Prior work on entailment classification using LASER based sentence embedding has a BiLSTM encoder trained for 93 languages and includes Malayalam also. With character and word level representations, it produces sentence embeddings which are mapped in a semantic space. A feed forward neural network having sigmoid/softmax activations classifies the dataset into binary/ 3-class. It is more generic approach and the size of the dataset is immaterial when using a pretrained model.

In Fig. 3, and Fig. 4, the notation 'LS' denotes the LASER-based approach using deep learning approach, and the rest are the machine learning feature-based methods. From the figure, it can be inferred that when the dataset size is around 2000, both machine learning and deep learning approaches perform similar classifications. As and when the data is increased, deep learning based methods become more suitable, and it is observed through the comparison with this feature-based machine learning implementation. It also supports the fact that earlier works in English with RTE datasets used feature based approaches.

With 2000 samples of data, we have obtained good results with feature based classification. As the sample size increases, deep learning methods became more efficient in classification supporting the related works with SNLI dataset. This work adds to the literature for Malayalam entailment or inference tasks as a baseline for machine learning based on the feature set approach, which is novel with respect to this language. As the dataset is generic in nature, the distinguishing characteristic of features becomes low, and this can lead to poor classification on large datasets. Thus the performance of feature-based classification is limited in terms of features that generalize well with datasets of high semantic variability. Hence, the rise in performance of deep learning approaches hints that these are methods that can be adopted from small to large datasets.

B. Ablation Study

The ablation study for this work includes the performance of different features contributing to the classification of inferences in text in the Malayalam language. With the set of features, namely, lexical, semantic, and distance measures, we have studied the performance of different feature set combinations, and the results are discussed here.

TABLE VIII. F1-SCORE FOR DIFFERENT FEATURE SET COMBINATIONS WITH DIFFERENT CLASSIFIERS.

	F1	F2	F3	F4	F5	F6	F7
LR	0.47	0.38	0.43	**0.48**	**0.48**	0.43	**0.48**
SVM	0.48	0.37	0.43	0.48	0.49	0.44	**0.50**
RF	**0.49**	0.39	0.43	**0.49**	0.48	0.43	0.48
NN	0.37	0.15	0.15	0.43	**0.49**	0.15	0.47
MNB	0.40	0.15	0.28	0.41	0.41	0.31	**0.42**
DT	**0.47**	0.36	0.41	**0.47**	0.46	0.38	0.46
AdB	0.48	0.38	0.42	0.48	0.48	0.43	**0.49**

TABLE IX. MODEL SELECTION

Feature set	Performance (#classifiers with max F1-score/#classifiers)
F1	0.29
F2	0
F3	0
F4	0.43
F5	0.29
F6	0
F7	**0.57**

The chosen setting for the experimental results combines lexical, semantic, and distance measures (F7). Also, we have studied the model performance with only lexical (F1), semantic (F2), distance-based (F3), lexical and semantic (F4), lexical and distance (F5), and semantic and distance-based (F6). Based on Table VIII, the feature set that performs good on a

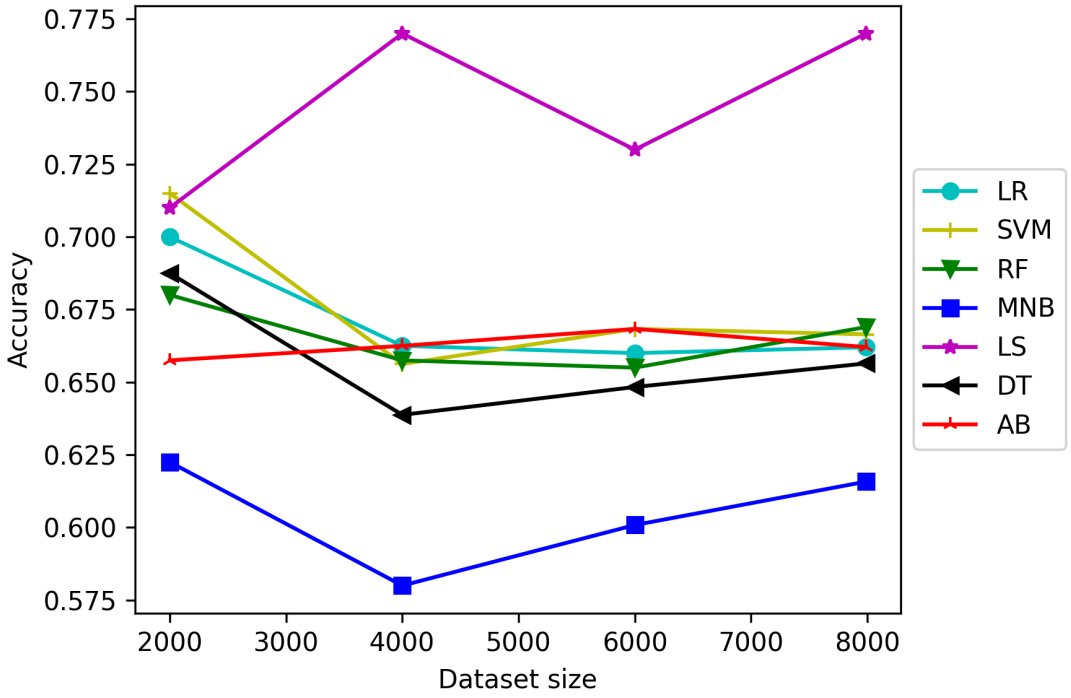

Fig. 3. Accuracy based Comparison (ML vs DL) Plot for Binary Classification, ML Methods are LR: Logistic Regression, SVM: Support Vector Machine, RF: Random Forest, MNB: Multinomial Naive Bayes, DT: Decision Tree, AB: Adaptive Boosting, DL method is LS: LASER based classifier.

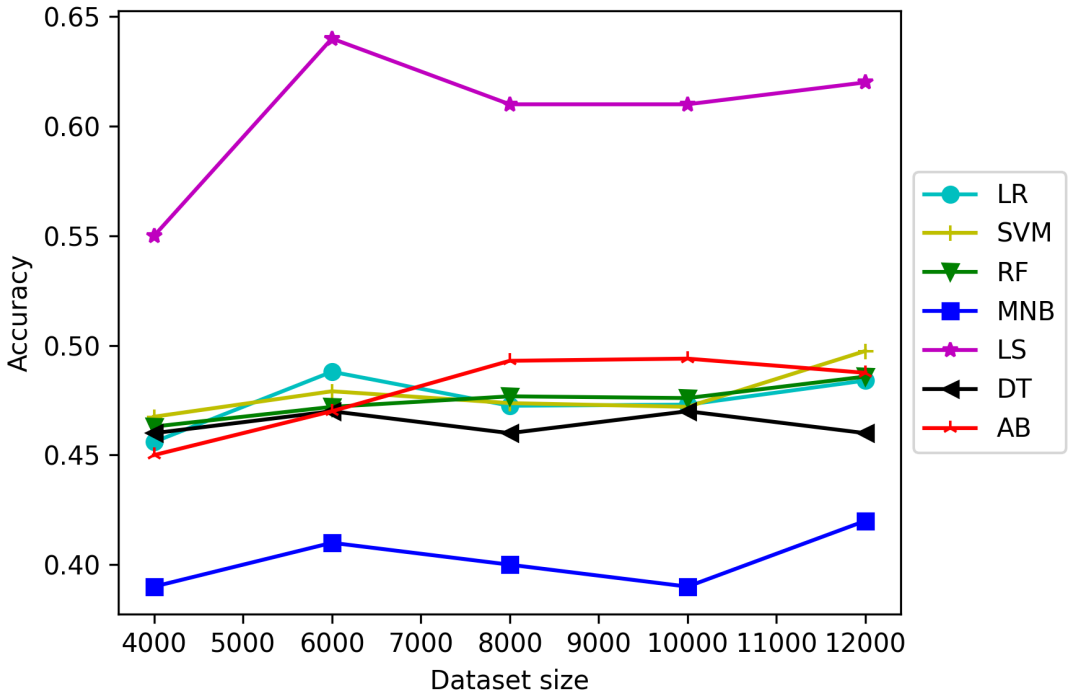

Fig. 4. Accuracy based Comparison (ML vs DL) Plot for Multiclass Classification, ML Methods are LR: Logistic Regression, SVM: Support Vector Machine, RF: Random Forest, MNB: Multinomial Naive Bayes, DT: Decision Tree, ADB: Adaptive Boosting, DL method is LS: LASER based Classifier.

majority of classifiers is chosen for analysis and comparison. The feature set performance is evaluated as in Table IX. The feature set performance is evaluated as the ratio of the number of classifiers with maximum F1-score to the total number of

classifiers. This justifies the selection of feature set F7, having maximum performance for experimental evaluations.

VII. Conclusion And Future Work

In this work, textual entailment is recognized for the Malayalam language with a feature-based approach. A set of classifiers are used to evaluate the performance accuracy. The best feature set model is chosen based on the F1-score measures. It is the first feature based attempt in this language for textual entailment recognition. This method also helped us understand the significant performance of deep learning methods, which is evident in the comparison. Thus this work on feature-based textual entailment recognition for the Malayalam language is substantial to the language resources community. The work is also essential and useful in identifying inferences in Malayalam texts for various language processing and social networking applications. Future work can include deep learning models to recognize entailment and these systems can be used in language processing applications.

Acknowledgment

The authors would like to thank the Department of Computer Science, CUSAT for the support extended in carrying out this research work.

References

[1] O. Glickman and I. Dagan, "A probabilistic setting and lexical coocurrence model for textual entailment," in *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005, pp. 43–48.

[2] S. Ghuge and A. Bhattacharya, "Survey in textual entailment," *Center for Indian Language Technology, retrieved on April*, 2014.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

[4] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1112–1122.

[5] P. Pakray, S. Bandyopadhyay, and A. Gelbukh, "Textual entailment using lexical and syntactic similarity," *International Journal of Artificial Intelligence and Applications*, vol. 2, no. 1, pp. 43–58, 2011.

[6] P. Pakray, S. Neogi, P. Bhaskar, S. Poria, S. Bandyopadhyay, and A. F. Gelbukh, "A textual entailment system using anaphora resolution." in *TAC*, 2011.

[7] T. Saikh, S. K. Naskar, C. Giri, and S. Bandyopadhyay, "Textual entailment using different similarity metrics," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2015, pp. 491–501.

[8] P. Malakasiotis and I. Androutsopoulos, "Learning textual entailment using svms and string similarity measures," in *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, 2007, pp. 42–47.

[9] R. Basak, S. K. Naskar, P. Pakray, and A. Gelbukh, "Recognizing textual entailment by soft dependency tree matching," *Computación y Sistemas*, vol. 19, no. 4, pp. 685–700, 2015.

[10] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

[11] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[12] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *arXiv preprint arXiv:2010.11934*, 2020.

[13] P. K. Rath and R. Basak, "Automatic detection of fake news using textual entailment recognition," in *2020 IEEE 17th India Council International Conference (INDICON)*, 2020, pp. 1–6.

[14] J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, and K. Satoh, "Coliee 2020: methods for legal document retrieval and entailment," in *JSAI International Symposium on Artificial Intelligence*. Springer, 2020, pp. 196–210.

[15] I. Dagan, B. Dolan, B. Magnini, and D. Roth, "Recognizing textual entailment: Rational, evaluation and approaches–erratum," *Natural Language Engineering*, vol. 16, no. 1, pp. 105–105, 2010.

[16] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda, "Overview of ntcir-9 rite: Recognizing inference in text." in *Ntcir*. Citeseer, 2011.

[17] M. Liu, Y. Guo, and L. Nie, "Recognizing entailment in chinese texts with feature combination," in *2015 International Conference on Asian Language Processing (IALP)*. IEEE, 2015, pp. 82–85.

[18] J. Bos, F. M. Zanzotto, and M. Pennacchiotti, "Textual entailment at evalita 2009," *Proceedings of EVALITA*, vol. 2009, no. 6.4, p. 2, 2009.

[19] M. Alabbas, "A dataset for arabic textual entailment," in *Proceedings of the Student Research Workshop associated with RANLP 2013*, 2013, pp. 7–13.

[20] N. Almarwani and M. Diab, "Arabic textual entailment with word embeddings," in *Proceedings of the third arabic natural language processing workshop*, 2017, pp. 185–190.

[21] A. Conneau, R. Rinott, G. Lample, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov, "Xnli: Evaluating cross-lingual sentence representations," in *EMNLP*, 2018.

[22] S. Renjit and S. Idicula, "Natural language inference for malayalam language using language agnostic sentence representation," *PeerJ Computer Science*, vol. 7, p. e508, 2021.

[23] S. Renjit and S. M. Idicula, "Siamese networks for inference in Malayalam language texts," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. Held Online: INCOMA Ltd., Sep. 2021, pp. 1167–1173. [Online]. Available: https://aclanthology.org/2021.ranlp-main.131

[24] K. Nadh, "Olam dictionary," https://olam.in/, 2010, accessed: 2021-11-16.

[25] S. Renjit, S. & Idicula, "Manli dataset," https://github.com/SaraRenG/MaNLI-data, 2020, accessed: 2021-11-16.

[26] S. Thottingal, "Libindic stemmer," https://github.com/libindic/indicstemmer, 2018, accessed: 2021-11-16.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.