iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** X **Month of publication:** October 2024

DOI: https://doi.org/10.22214/ijraset.2024.64747

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

Improving Word Embedding on Malayalam Corpus

Fathima Murshida K[1], Ruby Fathima A C[2]
Assistant Professor, Farook College, BVOC Software Development, Kerala, India

***Abstract:** NLP is natural language processing or neuro linguis-tic programming.Natural languages like malayalam are highly inflectional and agglutinative in nature.This is problematic whendealing with nlp based malayalam applications.So that inorder toimprove performance of malayalam nlp based applications, wordembedding improvement on malayalam corpus is needed.The improvement is based on converting the words contained inthe malayalam corpus into a standardised means removingall inflectional parts in the words in the existing malayalamcorpus ie taking root words only.All that needed is a stemmer.Inthis project i have used a malayalam morphological analyser for taking root words of all words in the existing malayalam corpus.The advanatge of removing inflectional parts from allwords is that we can reduce the sparsity in the existing malayalam corpus.Also there will be a high hike in frequency of wordsin the resulting corpus,then the space and time complexity of wordembedding representation of the existing corpus willdecreases.According to zipfs law by increasing frequency ofwords performance of neural word embedding will increases. Zipfs Law is a discrete probability distribution that tells you the probability of encountering a word in a given corpus.By applying zipfs law am proposing there will be improvement on malyalam wordembedding.Here using fasttext, word embeddingsare performed and capture dense word vector representation ofthe malayalam corpus with dimensionality reduction from thesparse word co-occurence matrix.The improvement is mainly used for wordnet,analogy,ontology based malayalam applications.Index Terms—Morphological Analyzer,Zipfs law, Preprocess-ing, Testing, Training*

I. INTRODUCTION

NLP stands for Neuro-Linguistic Programming. Neuro refers to your neurology, Linguistic refers to language, pro- gramming refers to how that neural language functions. In other words, learning NLP is like learning the language of your own mind.It is the sub-field of AI that is focused on enabling computers to understand and process human languages.Computers can't yet truly understand Natural lan- guages in the way that humans do — but they can already doa lot.We might be able to save a lot of time by applying NLP techniques to projects.

In this project am using Malayalam language.Malayalam is an Dravidian Indian language spoken in the Indian state of Kerala. It is one of the 22 scheduled languages of India and was designated a Classical Language in India in 2013. The earliest script used to write Malayalam was the Vatteluttu script, and later the Kolezhuttu, which derived from it. The oldest literary works in Malayalam, distinct from the Tamil tradition, are the Paattus, folk songs, dated from between the 9th and 11th centuries. Grantha script letters were adopted to write Sanskrit loanwords, which resulted in the modern Malay-alam script. Malayalam is a highly inflectional and agglu- tinative language.Algorithmic interpretation of Malayalam's words and their formation rules continues to be an untackled problem. The word order is generally subject–object–verb,although other orders are often employed for reasons suchas emphasis. Nouns are inflected for case and number, whilst verbs are conjugated for tense, mood and causativity (and also in archaic language for person, gender, number and polarity). Being the linguistic successor of the macaronic Manipravalam, Malayalam grammar is based on Sanskrit too.Because of its high complexity nature it is challenging to work on malayalamlanguage.

Morphological analyzer and morphological generator are two essential and basic tools for building any language pro- cessing application. Morphological Analysis is the process of providing grammatical information of a word given its suffix. Morphological analyzer is a computer program which takesa word as input and produces its grammatical structure as output. A morphological analyzer will return its root/stemword along with its grammatical information depending upon its word category. For nouns it will provide gender, number, and case information and for verbs, it will be tense, aspects,andmodularity. In my project need to remove inflections from each words in the corpus so that the frequency of words inthe resulting corpus will increase. So the resulting corpus will contain only the root words. By increasing the frequency of words,complexity of word embedding will decrease.

Because sparsity is more in inflected words.Also with limitted re- souces will get better output provided that not consideringor taking any morphosyntactic information.In my project am using only the conceptual similarity,that is conceptually similar words co occurene so that malayalam NLP based wordnet,ontology,analogy applications can improve their per- formance.

In Natural Language Processing we want to make computerprograms that understand, generate and, more generally speak-ing, work with human languages. But there's a challenge that jumps out: we, humans, communicate with words and sen- tences; meanwhile, computers only understand numbers.For this reason, we have to map those words (sometimes eventhe sentences) to vectors: just a bunch of numbers ,That's called text vectorization.It is also termed as feature extrac- tion.Different ways to convert text into numbers are Sparse Vector Representations and Dense Vector Representations.

II. LITERATURE SURVEY

A. *State of the art*

1) *Word2Vec :* Word2Vec is a statistical method for effi- ciently learning a standalone word embedding from a textcorpus.It was developed by Tomas Mikolov, et al. at Google in 2013 as a response to make the neural-network-based training of the embedding more efficient and since then has becomethe de facto standard for developing pretrained word embed- ding.The model is, in contrast to other deep learningmodels,a shallow model of only one layer without non-linearities.The paper by Mikolov et al. introduced two architectures for unsupervisedly learning word embeddings from a large corpus of text. The first architecture is called CBOW, it triesto predict the center word from the summation of the context vectors within a specific window. The second, and more successful architecture, is called skip-gram . This architecture does the exact opposite, it tries to each of the context words directly from the center word. The used pretrained word2vec embeddings are trained using the Skip-gram algorithm. This algorithm is also the inspiration for the algorithms behind the GloVe and fastText embeddings.
2) *GloVe :* Global Vector for Word Representation by Pennington, Socher, and Manning (GloVe) was inspired bythe skip-gram algorithm and tries to approach the problem from a different direction. Pennington, Socher, and Manning show that the ratio of co-occurrence probabilities of two specific words contains semantic information. The idea is similar to TF-IDF but for weighing the importance of a context word during the training of word embeddings.Classicalvector space model representations of words were developed using matrix factorization techniques such as Latent Semantic Analysis (LSA) that do a good job of using global text statistics but are not as good as the learned methods likeword2vec at capturing meaning and demonstrating it on tasks like calculating analogies.Their algorithm works by gathering all co-occurrence statistics in a large sparse matrix X, whereineach element represents the times word i co-occurs with j within a window similar to skip-gram. After which the word embeddings are defined in terms of this co-occurrence matrix.
3) *fastText :* fastText as a library for efficient learningof word representations and sentence classification. It is writ- ten in C++ and supports multiprocessing during training. FastText allows you to train supervised and unsupervised representations of words and sentences. These representations (embeddings) can be used for numerous applications from data compression, as features into additional models, for candidate selection, or as initializers for transfer learning. Bojanowski et al. introduced the fastText embeddings by extending the skip- gram algorithm to not consider words as atomic but as bags ofcharacter n-grams. Their idea was inspired by a the work fromSch¨utze in 1993, who learned representa- tions of character four-grams through singular value decomposition (SVD). Oneof the main advantages of this approach is that word meaning can now be transferred between words, and thus embeddings of new words can be extrapolated from embeddings of the n- grams already learned. The length of n-grams you use can be controlled by the -minn and -maxn flags for minimum and maximum number of characters to use respectively. Thesecontrol the range of values to get n-grams for. The modelis considered to be a bag of words model because aside ofthe sliding window of n-gram selection, there is no internal structure of a word that is taken into account for featurization,i.e as long as the characters fall under the window, the order of the character n-grams does not matter.
4) *Paragram:* Wieting et al.introduced a method to tune existing word embeddings using paraphrasing data. The focus of their paper is not on creating entirely new word embeddingsfrom a large corpus. Instead, the authors are taking existing pretrained GloVe embeddings and tune them so words in sim- ilar sentences are able to compose in the same manner. Their training data consists of a set of P phrase pairs (p 1 , p 2) , where p 1 and p 2 are assumed to be the paraphrases. The ob- jective function they use focuses to increase cosine similarity,i.e. the similarity of the angles between the composed semanticrepresentations of two paraphrases.Important to mention isthat Wieting et al. expresses similarity in terms of angle and not in terms of actual distance. Additionally, Wieting et al. only explored one algebraic composition function, namely: averaging of the word vectors.

The data for tuning the embeddings used by the authorsis the PPDB 5 or the Paraphrase Database by Ganitkevitch, Van Durme, and Callison-Burch . Specifically,they used ver- sion XXL which contains 86 million paraphrase pairs.

An example of a short paraphrase is: "thrown into jail" which is semantically similar to "taken into custody". Wieting et al. published their pretrained embeddings called Paragram-Phrase XXL 6 , which are in fact tuned GloVe embeddings, alongside their paper. These em- beddings also have a dimensionality of 300 and have a limited vocabulary of 50,000.In order to apply the embeddings, according to Wieting et al., they shouldbe combined with Paragram-SL999 which are tuned on the SimLex dataset.

III. PROPOSED METHOD

The increasing accuracy of word embedding representation of malayalam languae has a great impact on ontology,analogy representation based NLP based applications. Improved Word Vectors on Malayalam language, which increases the accuracyof pre-trained malayalam wordembeddings in NLP Applica- tions . The main aim is to improve word embeddings thatthey do not need any labeled data.The improvement is by converting malayalam language into a standardised format by removing inflections from each word in malayalam corpus.

Any large volume of text can be used to get the word embeddings by feeding it to the model, without any kindof labeling.In this project am using fastText wordembedding. Word embeddings are derived by training a model on large text corpus.

A. System Architecture

The system architecture mainly consist of five modules.They are Corpus Creation ,Data Preprocessing, Training UsingNeural Networks, Model Evaluation,Dataset Visualisation.

1) *Corpus Creation :* Datas are collected from Malayalam news articles and created a malayalam Corpus containing thousands of sentences.
2) *Data Preprocessing:* Data Preprocessing is the impor- tant part of this project,Because datas in the real world is incomplete,noisy,inconsistent. Also malayalam language is highly inflectional and agglutinative. In this , a malayalam morphological analyser is used to generate root words or removing inflectional parts from the existing malayalam cor- pus.By using sandhi rules malayalam morphological analyser will generate root words of each word in the existing corpus.So in the resulting corpus frequency of malayalam words will increases.According to zipfs law in a large corpus of natural language like malayalam, the frequency of any word is in- versely proportional to its rank in frequency table.Frequency isnumber of times a word appears in a given corpus.By remov- ing inflections frequency of words will increases, according to zipfs law by increasing frequency of words performance of neural word embedding will increases.
3) *Training Using Neural Networks :* In this project am using fastText wordembedding.fastText is another word embedding method that is an extension of the word2vec model. Instead of learning vectors for words directly, fastText represents each word as an n-gram of charac- ters. This helps capture the meaning of shorter words and allows the embeddings to understand suffixes and prefixes. Once theword has been represented using character n-grams, a skip- gram model is trained to learn the embeddings.Fasttext can be used both for classification and word-embedding creation.Hereit will generate the improved word embedding of the new malayalam corpus which is containing root words only.
4) *Model Evaluation:* Word embeddings should capture the relationship between words in natural language.In the Word Similarity and Relatedness Task, word embeddings are evaluated by comparing word similarity scores computed froma pair of words with human labels for the similarity or relat- edness of the pair.Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.It is often used to measure document similarity in text analysis.In this project cosine similarity measure is used to calculatethe similar words.
5) *Model Visualisation:* PCA and T-sne are two techniques for visualising dataset in 2D or 3D space. PCA performs a linear mapping of the data to a lower-dimensional space. T- Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction. Well suitedfor the visualization of high-dimensional datasets.So that tsne visualisation technique is used for this project.By using tsne visualisation it can be seen that similar words tend to be close to each othe and dissimilar words tend to be far from each other.

IV. RESULTS

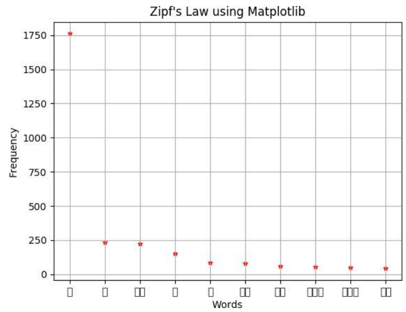


Fig. 1. Graphical result of existing dataset after applying zipfs law

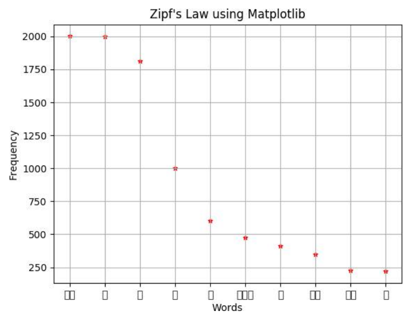


Fig. 2. Graphical result of new malayalam dataset after appying zipfs law

V. CONCLUSION AND FUTURE WORK

It can be visualise that similar malayalam words or syn- onyms occupy close to each other by using tensorflow em- bedding projector. The Eucledian distance and cosine simi- larity of similar words will be lesser than the fastText pre trained word vectors of malayalam language. It can be predict that by using zipfs law malayalam word embedding can be improved. Also sparsity problem in the existing malayalam corpus is decreased by using less resources that is conceptual similarity of words.Space and time complexity is reduced in the new malayalam corpus. The improved word vectors canbe used to all NLP malayalam applications to improve their efficiency,accuracy,speed. In future for machine translationimprovement on malayalam this model can be used as a base model and also learn morphosyntatic information on that model.

REFERENCES

[1] Abdulaziz M. Alayba, Vasile Palade, Matthew Englandand Rahat Iqbal ” Improving Sentiment Analysis inArabic Using Word Representation”, IEEE 2nd International Workshop on Arabic and Derived Script Analysisand Recognition (ASAR),2018 .

[2] Shaosheng Cao Wei Lu, ”Improving Word Embeddings with Convolu- tional Feature Learning and Subword Information”, Proceedings of the2017 Conference on Empirical Methods in Natural Language Processing

[3] YuvalPinter,RobertGuthrie,Jacob Eisenstein, ” Mimicking Word Embed- dings using Subword RNNs”, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. .

[4] Vishwani Gupta,Sven Giesselbach,Stefan R¨uping,Christian Bauckhage,”Improving Word Embeddings Using Kernel PCA”, Proceedings of the 4th Workshop on Representation Learning for NLP.

[5] Procheta Sen,Debasis Ganguly , ”Word-Node2Vec: Improving Word Em-bedding with Document-Level Non-LocalWord Co-occurrences”, Pro- ceedings of the 2019 Conference of the North American Chapterof the Association for Computational Linguistics: Human LanguageTechnologies..

[6] Miguel Ballesteros, Chris Dyer, and Noah A. Smith, ”Improved transition-based parsing by modeling characters instead of words with LSTMs”, In Proc. EMNLP.2015.

[7] Marco Baroni and Alessandro Lenci, ”Distributional memory: A generalframework for corpus-based semantics”, Computational Linguistics, 36(4):673–721, 2010.

[8] Jan A. Botha and Phil Blunsom, ” Compositional morphology for word representations and language modelling”, In Proc. ICML, 2014.

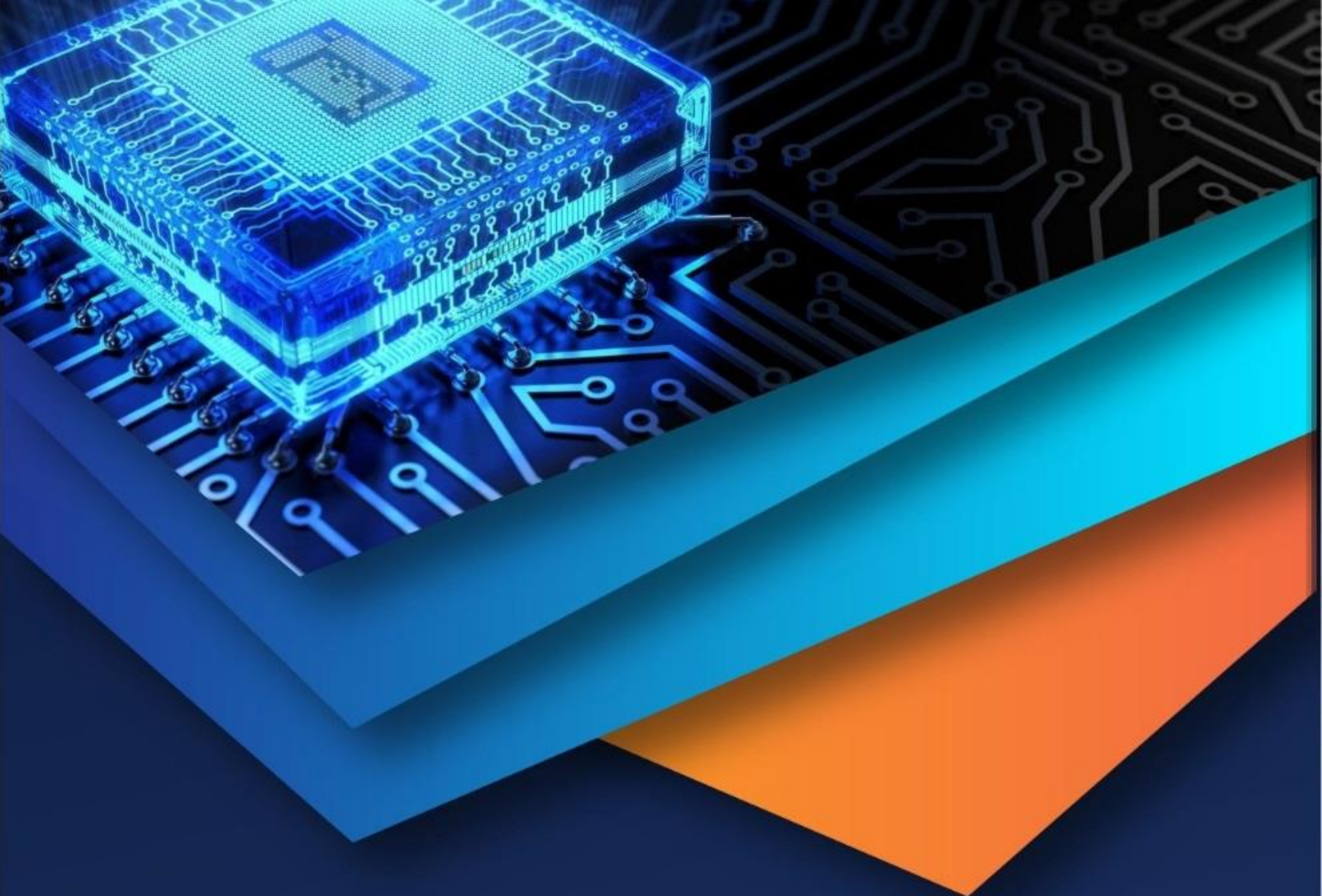
[9] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan, ”Joint learning of character and word embeddings”, In Proc. IJCAI,2015.

[10] Cicero Nogueira dos Santos and Bianca Zadrozny, ”Learning character- level representations for part-ofspeech tagging”, In Proc. ICML,2014.

[11] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, ”Detecting spammer on twitter”, in Proc. 7th Annu. Collaboration, Electron. Mes- saging, Anti-Abuse Spam Conf., Jul. 2012, p. 12

[12] J. Song, S. Lee, and J. Kim, ”Spam filtering in Twitter using sender receiver relationship”, in Proc. 14th Int. Conf. Recent Adv. Intrusion Detection, 2011, pp. 301317.

[13] Chao Chen, Yu Wang, Jun Zhang, Yang Xiang, WanleiZhou, ”StatisticalFeatures-Based Real-Time Detection of Drifted TwitterSpam, Twit- terSpam,”IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 12, NO. 4, APRIL 2017.
[14] Sayali Kamble,S.M.Sangve, ””Real Time Detection of Drifted Twit- terSpam Based on Statistical Features”, Features”2018 International Conference on Information, Communication, Engineering and Technol- ogy (ICICET) .
[15] Surendra Sedhai , Aixin Sun, ”Semi Supervised Spam Detection TwitterStream”, ”IEEE Transactions On Computational Social Systems 2017.
[16] Hajime Watabe, Mondher Boduazizi,And Tomoaki Ohtsuki, ””Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection”, ”IEEETransactions On Computational Social Systems 2018.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY