Indian Institute of Information Technology Vadodara

Design Project 2021

on

Verselet - The Poem Classifier

Submitted by

Nikhil Rana **201951103**

Vishal Singh Rajput **201951171**

Shashank Jaiswal **201951140**

Subhanjali Sharma **201952236**

Under The Supervision of

Dr.Jignesh Bhatt

Signature Link https://bit.ly/30p3AN2

***Abstract*—Poetic eloquence is a leaf of the linguistic arts tree which just like its neighbours,visual and performing arts is profoundly influenced by the perceptions and ideologies of its creators. Take the work of any great author one can always find the tinge and involvement of the era thus materializing itself in the form of expressions and vocabulary and making that era feel eternal. The works of the great authors like Kabir , Mirabai can be seen to be influenced by the Bhakti movement. They wrote most of their linguistic work in their vernacular Hindi of the devanagari script while also borrowing words from their neighbouring dialects like Braj, Mewari , Marwari enclosing within them the topics of dedication, discipline and mysticism. All such famous and ever living works include the use of various literary devices, rhyming schemes and the inborn preferences of the creator. But such vast differences of the work of the authors with each author's style of expressing akin to a fingerprint can be used to predict the name of the poet and the era it was created for any unseen poem. This task has been effectively carried out for the English language but Indic language mainly the largest one that is Hindi has been completely untouched. We therefore proposed to implement a model that handles this task for Hindi poems.**

I. Introduction

POetry has been one of the oldest form of art known to mankind. It has also been regarded as one of the noblest, with poets being bestowed with prestigious prizes by the most thriving kings throughout Indian history. One of the leading jewels of all time is Tan Sen who was one of the nine jewels of Akbar's court. Where all other genres are restricted with plot, narration and grammatical consistency, Poetry is free from all such restrictions. The words used in poems may not be used in their literal meaning but a deeper contextual connotation.

A. Motivation

Poetry analysis for Indic languages has received less attention compared to English language in recent years so this project is motivated to change that perception. Adding on different challenges that this study can offer were also the major reasons to make this project like to get a complete set of vocabulary and word embedding because of the extensive use of vernacular dialects prevalent across different eras and because the language Hindi in general has evolved so much across the scores of centuries. Creating a corpus was another challenge as there was no available corpus that we can use at the time.

B. Objective

We started our project by creating a database of the poems with the poet names and the eras in which those poems were written and used this database to predict the poets and eras for test samples. Our project classify the computer generated poems towards the era and poet they closely resembles. This project may help in more deep exploration of poetry in Indic Languages in future.

II. Literature Survey

According to best of our knowledge much work has not been done in Hindi language so we present you the work that has been done in different languages as well.
In the paper,[4] submitted in 2019 has made an attempt to perform the automatic poetry classification for English poems using NLP. It has been highlighted in this paper that automatic poetry diction is possible by using word embedding as features. He has also used CNN based model. There are some models which classify poem based on sonic elements

but we restrict ourselves to text only. This will make it easy for us to train NLP and ML models. There are some tools like SPARSER[2] which aims to study poetry by the use of NLP tools like tokenisation, sentence splitters and taggers. Since corpus of SPARSER is about 500 poems so its efficiency is good.

Text-mining methods have been developed which when applied to the poems helps in recog- nizing the author of the text done by ([6]). In pre-processing step, all the words are converted from uppercase into lowercase, punctu- ation marks and digits are deleted, tokenization is done according to the white space character, stemming type of all the words is identified and stop words are deleted. tf-idf method have been used to apply term weighting.

III. The Present Investigation

The entire process of our classifier is divided in four major parts **Dataset Creation, Data cleaning and prepossessing, Vectorisation and Models for prediction**.

A. *Dataset Creation*

The initial step involved in poem classification is dataset creation. Hindi being a second world language there was no precreated corpus available publicly. So our first task was to create a corpus for our classifier. We used the web crawling technique to create our dataset. A web crawler, sometimes called a spider is an internet bot that systematically browses the world wide web, typically operated by search engines for the purpose of web indexing. We used the web crawlers provided by two libraries BeautifulSoup and Scrappy in python to crawl through and scrap various sites, most prominently- http://kavitakosh.org/ and http://www.hindwi.org/poets. Not all the sources provided the year in which the poem was written which made it hard to assign the era to the poem. The poems for which the year was not available we used the date of birth of the poet as an estimate for the time of the creation. But in many cases this was also missing, to outwit this obstacle we manually elucidated the poets with the years they were active in.The final step is to assign eras to the poems on the basis of the year of their creation. We use the classification into eras as listed on https://en.wikipedia.org/wiki/Hindiliterature .Therefore after all this work of collecting poems from various sources, we eliminate the repeated and duplicates from them using hashing. Thus we created our final dataset that we split into train and test set randomly(90%-10% split). We have 47088 poems in our train set and 5232 poems in our test set.

B. *Data Cleaning and Preprocessing*

The next step in the poem classifier is data cleaning and preprocessing. We passed the poems through 2 sub phases of text classification which are tokenization and stop word removal.Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. For tokenization we used iNLTK library which has pre-trained model for tokenization.A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.Various punctiation marks like comma(,), poorna viram(—) and exclamation mark(!) among other are commonly occurring symbols in the poems, therefore we further go on to remove the various stop words from the tokenized output. The stop words have been collected from various sources - https://data.mendeley.com/datasets/bsr3frvvjc/1, https://sites.google.com/site/kevinbouge/stopwords-lists, https://github.com/taranjeet/hindi-tokenizer/blob/master/stopwords.txt.

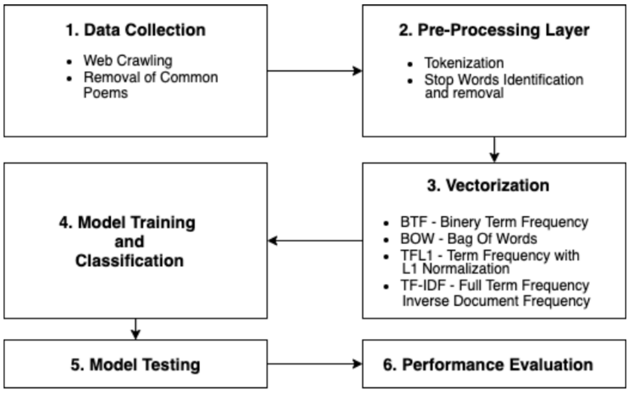
1. Data Collection
Web Crawling
Removal of Common Poems
2. Pre-Processing Layer
Tokenization
Stop Words Identification and removal
3. Vectorization
BTF - Binery Term Frequency
BOW - Bag Of Words
TFL1 - Term Frequency with L1 Normalization
TF-IDF - Full Term Frequency Inverse Document Frequency
4. Model Training and Classification
5. Model Testing
6. Performance Evaluation

Fig. 1. Project Architecture

C. *Vectorisation*

Next step in our process is vectorisation where we will convert the preprocessed data obtained from step two into the format which later can be used to train different models. We have used different vectorisation techniques in order to make it more accurate for the models we were gonna train further. Basically we convert the text into numerical representation by using several different techniques.These different vectorization techniques were ap- plied by passing appropriate parameters to the TfidfVectorizer included in the sklearn (sci-kit learn) package in python.

1) Binary Term Frequency(BTF): In this techniques we uses presence as 1 and absence as 0 of a term in a document. Means if the term is there in the document its matrix value will be 1 and if it is not present then it will be 0.

2) Bag of Words(BOW): Here the term frequency is used to write the exact frequency of the term present in the document rather then just telling the presence "1" or absence "0" of the term as done in BTF. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things : A vocabulary of known words and A measure of the presence of known words. It is called a "bag" of words, because any information about the order or structure of words in the document is discarded. The technique is only concerned with whether known words occur in the document, not where in the document.

3) L1 Normalised Term Frequency(tfl1): It is a technique which applies L1 nor- malization on the BoW term frequency in the document.In L1 normalization each element in a vector is divided by sum of absolute values all elements. There is an option to L1 normalize the values in sklearn.

4) Term Frequency - Inverse Document Frequency(TF-IDF): TF-IDF is stands for Term Frequency-Inverse Document Frequency. The process of this method is used in information retrieval and text mining. It works on dataset or word of document-collection. The process of TF-IDF is used to define The time taken by a specific word in a document, number of documents with that specific word and ratio between the documents with that specific term by all documents. Stop words, high frequency and low frequency word are removed by TF-IDF. Tf-IDF methods are applicable for text summarization and classification.So distictness of term is taken into account means if a term is present in few documents, then the corresponding TF- IDF value is expected to be higher than other- wise.

5) GloVe: It is yet another method to convert words into their corresponding vectors. The main steps involves building a global co-occurence matrix for the given corpus and applying gra- dient descent to find the embeddings. We use these glove embeddings in the LSTM model described in 5 bullet. Moreover, we also rep- resent the documents (the poems in our case) by taking the mean of the vectors of words present in the poem.We have implemented our own GloVe model taking help from dirretent sources and mainly [5].

D. Model Training and Testing

The fourth step in poem classification is model training and testing. At the end of the third stage we have the representation of the poems data as vectors. We now test different NLP and ML based models for the task of classification with all the different vectorization methods explained previously. We have used appropriate functions from the sklearn package in python to implement the various models. We now briefly explain all the different models that we have explored in the course of this project.

*1) **Cosine Similarity**:* Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.Mathematically, it finds the dot product of the documents represented as vectors in a multi-dimensional space. Given a poem, we try to find the most similar poem from the training set (which has the highest value of cosine similarity) and then return the corresponding era and the poet name.

*2) **Logistic Regression**:* Here we are writing the code to predict the author name and the ero of the poem on the test data based on our training data for which the model we use here is logistic regression. Logistic Regression is a famous ML algorithm which comes under supervised learning technique. It is a predictive analysis algorithm that works on the concept of probability. We first load the data in our code and create separate lists of author and era of poem and poem from our data. We then convert these lists into arrays so that we can use it for prediction through this model. Than we pass the value of C as a parameter which is used to determine the strength of the regularization.

GridSearchCV is an effective method for adjusting the parameters in supervised learning and improving the generalization performance of the model. We try all the combinations of parameters and find the best one and find the accuracy of our model and the best value of C.

RandomisedSearchCV is very useful when we have many parameters to try and the training time is very long. Here also we try to find the accuracy and value of C which best fits our model.

*3) **Convolutional Neural Network**:* CNN is a kind of neural network, where for text classification, we work with a 1-Dimensional convolutional layer. In our model, we require a word embedding layer and a one-dimensional convolutional network. We consider text data as sequential data like data in time series, a one-dimensional matrix. Our CNN model is made up of neurons that have learnable weights and biases. Each neuron receives some weights and then performs a dot product and occasionally follows with a non-linearity depending on the need. The whole network expresses a single differentiable score function: from the raw vectorized input on one end to class scores at the other. There is a loss function on the last (fully - connected) layer. In our model of CNN, we have, firstly, a one dimensional convolutional layer, which is then followed by a max-pooling and flatten layer and then finally by an output layer. We have used the Keras open source library and implemented a sequential model, our model contains Conv1D, MaxPooling 1D and 2 Dense Layers and tuned the parameters accordingly and also used softmax activation

*4) **Long Short Term Memory**:* It is an artificial recurrent neural networks (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data.An LSTM unit is composed of a cell, input gate, output gate and a forget gate. This feature of Lstm is thoroughly useful for our modelling because it can remember long term relations and is less prone to vanishing gradient problems. To use the LSTM model we first found the GloVe word embeddings as described above. Then we passed these embeddings for the words of the poems into a LSTM layer. the output of this layer is then proceeded to 2 fully connected layers and then the final output layer which prints the softmax distribution of probabilities. We used the keras library to implement this model.

Models	Vectorisation Technique	Poet Accuracy	Era Accuracy
Cosine Similarity	BTF	15.815%	88.767%
	BOW	10.779%	82.725%
	TFL1	10.778%	82.721%
	TF-IDF	12.289%	84.671%
Logistic Regression (Random Search)	BTF	30.541%	89.506%
	BOW	14.784%	88.512%
	TFL1	3.210%	88.832%
	TFIDF	23.229%	89.506%
Logistic Regression (Grid Search)	BTF	29.034%	89.812%
	BOW	14.732%	88.780%
	TFL1	3.56%	88.527%
	TF-IDF	23.562%	89.559%
CNN	BTF	19.842%	82.93%
	BOW	30.40%	84.76%
	TFL1	11.98%	84.94%
	TF-IDF	7.84%	84.82%
LSTM	GloVe	3.732%	76.282%

TABLE I
ACCURACY OF DIFFERENT MODELS AND VECTORIZATION METHODS ON THE TEST SET

IV. RESULTS

As shown in table 1 the accuracy of different models with different vectoristation techniques for both poet prediction and era prediction. We have also presented the result in graphical formatin in figure 2 and figure 3. The best result obtained from era prediction has the value of **89.812%** and the poet prediction is **30.514%**. As we can see in the result that the efficiency of prediction in case of poet is much lower then that of era this may be result of several reason, out of which most prominent being that the number of poets to be predicted is much larger than the number of eras. Also because we have used a random split for test set creation, there are cases were the poet in the test set is not part of train set altogether.This makes it impossible for any model to give a correct result on these examples. This however shows the power of deduction of the era as even in the cases where we do not have any poem from a poet in the test set, the era is predicted with high accuracy.

Further in LSTM model we have seen GloVe is a great technique to learn word embed- dings, so we used them to get the document vector by taking the mean of all the words embeddings occurring in a poem. This however did not provide good results. We feel that this was because taking the mean of all the words GloVe embeddings polluted the document vector with embeddings from common words and therefore using these embed- dings for classification was no longer feasible.This model suffered from a problem of low generalizabilty. After training the model, the accuracy on the training set was 66.23% and 96.33% for poet and era respectively. This is much better than the results of any other model but unfortunately, this model did not generalize well to the test set. In order to tackle this problem, we explored various techniques like regularization, recurrent drop-out and early stopping. These helped a little but we were unable to beat the previously seen results.

Also the four vectorization methods used gave long and sparse vectors which made training the models a difficult task and we were forced to run lower number of iterations of the optimization algorithm for CNN model because of time constraints. Hence, the trained model is not the most optimal one. Nevertheless, Logistic Regression(Random search) model gave us the best result for poet classification. With more computation power and time, we would be able to get a better trained model for logistic regression which would further improve our results.

For era prediction, we saw similar results for all the vectorization techniques and the entire deviation was between 15%. Our models do not work well for poet predictions because the few number of poems that we have from each poet are not sufficient for differentiating their writing styles and vocabulary usage quirks. As we saw best result in case of era prediction was provided by Logistic Regression(Grid Search) model.

So in the result we can say that in our analysis Logistic regression model came out to be one of the best model for the prediction and classification type of data. As this model provided best accuracy result both in poet as well as in era prediction.

V. FUTURE WORK

In future we would like to work on GloVe embedding on other models also as in this case we were only able to use it in LSTM so further in future we would try to use it in other models. Also, we would like to work on CNN model as it is known for its better results since this time we were not able to get better results from this model we would like to do that in future and train this model to its optimised version. Also the size of the model had to be restricted. As a future work better models can be used for prediction.

Also, we were unable to generalize the long- short term memory model. This issue can be explored further and other

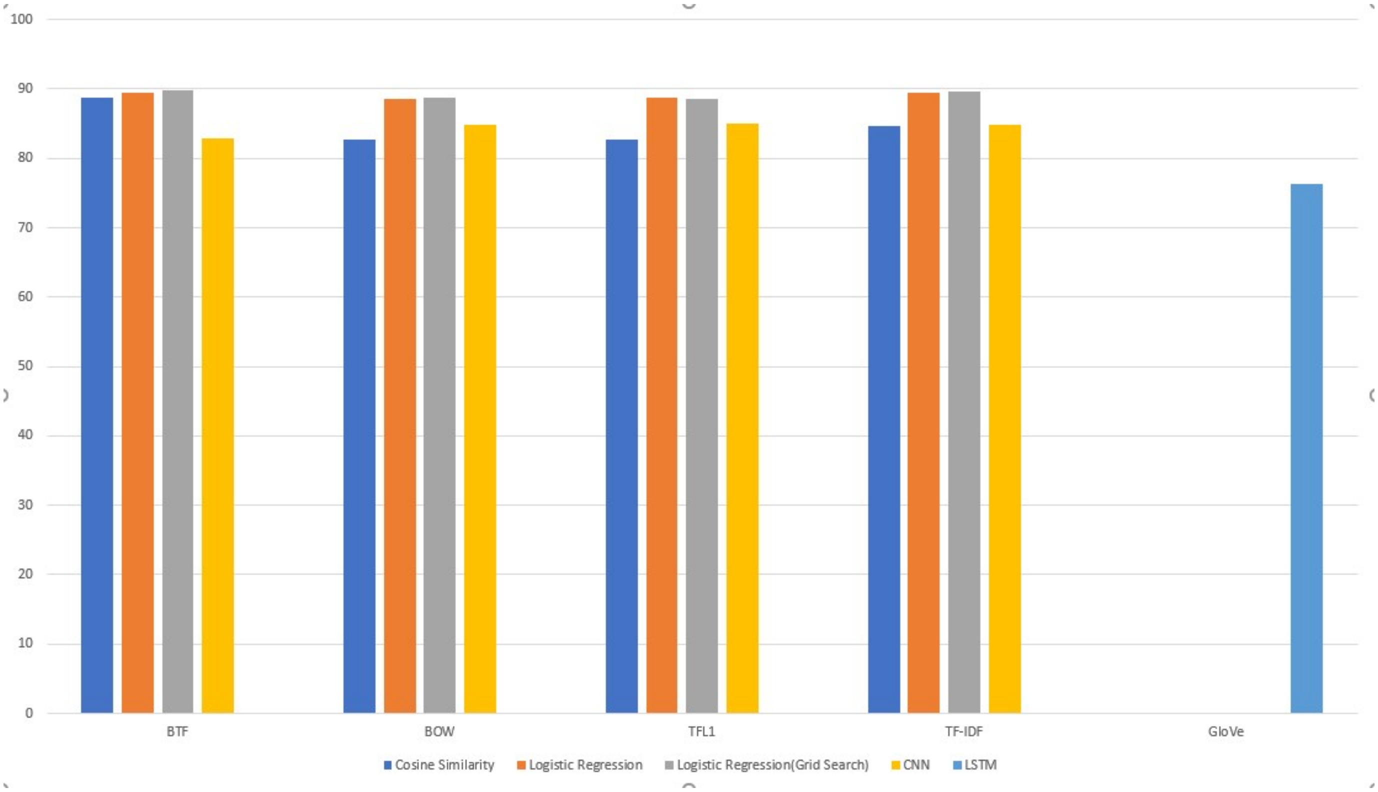

Fig. 2. Accuracy of Different Models and Vectorization Methods for Era Prediction

regularization techniques can be used.
In this project we only used random train-test split which led to some of the authors not being part of the training set entirely. Better splitting techniques can be used so that this problem is eliminated.
Instead of learning GloVe embeddings from the training set, other word embeddings like fast text[1],word2vec[7] can be used which are known to provide better results.

We have seen that bag of words and binary term frequency are providing the best results but give long and sparse vectors. Dimensionality reduction techniques can be used to reduce the vector sizes while preserving the information. This would also allow us to explore better models. The problem of few poems from each poet can be handled if more resources are devoted into dataset creation by augmenting the current dataset to incorporate more poems from all the poets.
And at last we would like to analyse similar models for other languages of world which are not recognised similarly to Indic languages.

VI. Conclusion

During the course of the project we compared the different approaches for determining the era and poets of the Hindi poems. The vectorization was done using 5 methods which are names: Bag of Words, Binary Term Frequency, Normalized Term Frequency(L1), Normalized TF-IDF and GloVe(used as word embeddings in LSTM model and to calculate document vectors for the other working models). The different Machine Learning models which we implemented were Cosine Similarity, Logistic Regression, Convolution Neural Networks and Long Short Term Memory. CNN models gave the better result of all the given working models on the given dataset. Our project explores new dimensions in the field of Poem Classification for Hindi Language and can be considered as the starting step for the creation and work in the Indic languages domain applying principles of Artificial Intelligence and machine learning for more exceptional results.

VII. Acknowledgement

We would like to express our sincere gratitude to Dr. Jignesh Bhatt, for his valuable guidance and support in completing our project. We would also like to thank Dr. Kamal Kishor Jha for conducting the process of design project smoothly.

References

[1] Ben Athiwaratkun, Andrew Gordon Wilson, and Anima Anandkumar. *Probabilistic FastText for Multi-Sense Word Embeddings*. 2018. arXiv: 1806.02901 `[cs.CL]`.

[2] Rodolfo Delmonte and Anton Maria Prati. “SPARSAR: An Expressive Poetry Reader”. In: *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 73–76. DOI: 10.3115/v1/E14-2019. URL: https://aclanthology.org/E14-2019.

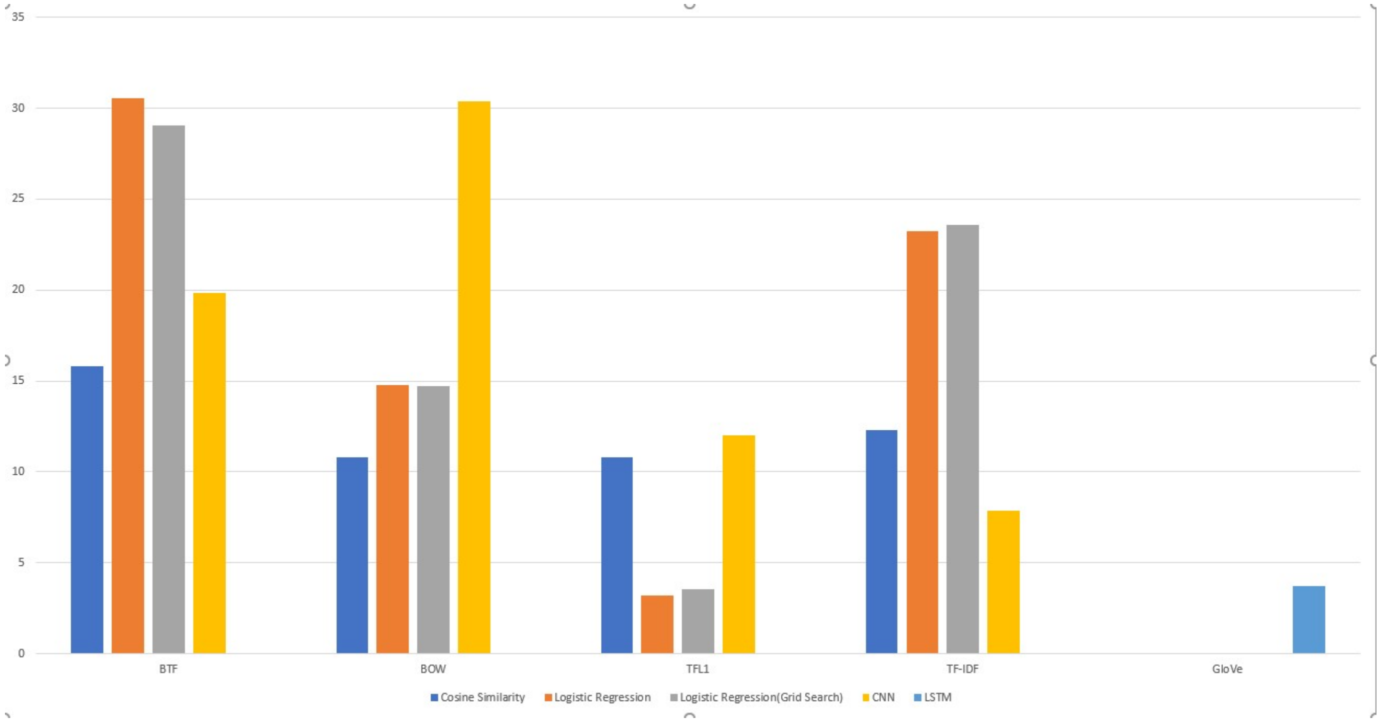

Fig. 3. Accuracy of Different Models and Vectorization Methods for Poet Prediction

[3] Pushpak Bhattacharyya Geetanjali Rakshit Anupam Ghosh and Gholamreza Haffari. *Automated Analysis of Bangla Poetry for Classification and Poet Identification*. 2015. URL: http://cdn.iiit.ac.in/cdn/ltrc.iiit.ac.in/icon2015/icon2015_proceedings/PDF/12_rp.pdf.

[4] Vaibhav Kesarwani. *Automatic poetry classification using natural language processing*. 2019. URL: https://ruor.uottawa.ca/bitstream/10393/37309/1/Kesarwani_Vaibhav_2018_thesis.pdf.

[5] Jeffrey Pennington, Richard Socher, and Christopher Manning. “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162.

[6] Alexandra Birch Rico Sennrich Barry Haddow. *Neural Machine Translation of Rare Words with Subword Units*. 2016. URL: https://aclanthology.org/P16-1162.pdf.

[7] Xin Rong. *word2vec Parameter Learning Explained*. 2016. arXiv: 1411.2738 [cs.CL].