Group - 13

Hindi Poem: Poet and Era Classification

Abhishek Jaiswal (170101002)
Hardik Katyal (170101026)
Manan Gupta (170101035)
Mayank Wadhwani (170101038)

Overview

01 Problem Formulation

02 Proposed Goal

03 Achieved Goal

04 Contribution of each member

05 Details of implementation

06 Key papers

07 Lessons Learnt

01

PROBLEM FORMULATION

Problem Formulation

- Much work done in English poems.

- **Our project** → An attempt to classify a given poem on the basis of Era and Poet for Hindi poems.

02

PROPOSED GOALS

Proposed Goals

- To create a database of Hindi poems using web crawling.
- Employ different models and methods to improve the accuracy of classification.

03

ACHIEVED GOALS

Achieved Goals

- Created a dataset of **50,000+ poems** from websites like **kavitakosh** and **hindwi**.
- Tested various models to get a best accuracy of **89.718%** for **era** and **32.40%** for **poets**.

04

CONTRIBUTION OF ALL TEAM MEMBERS

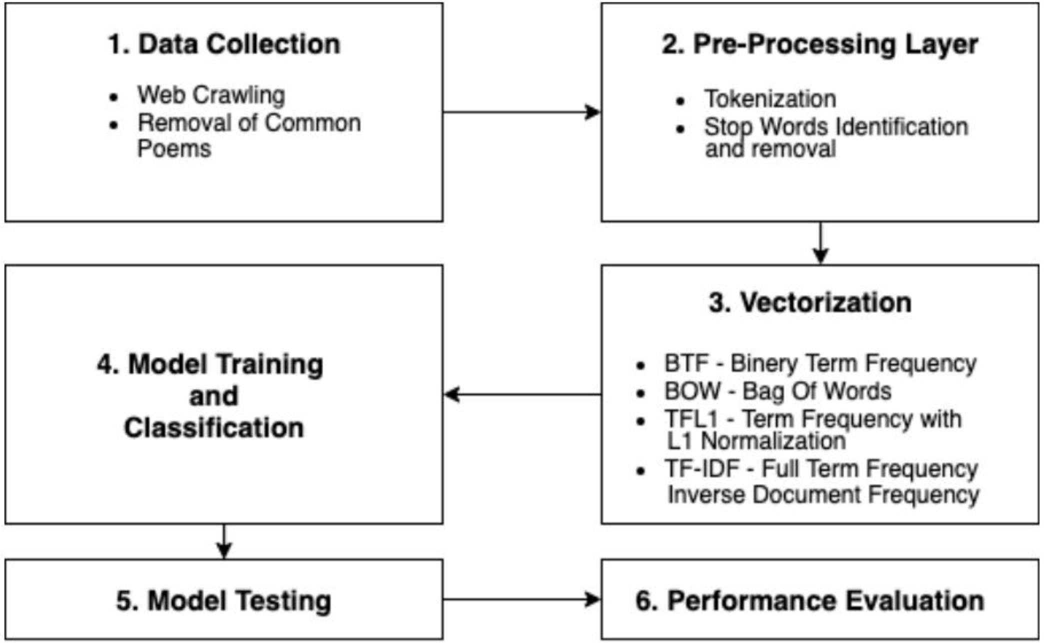
Contribution of Team Members

- Dataset Creation → Mayank and Manan
- Merging and Pre-processing → Abhishek and Hardik
- Vectorization → Manan and Mayank
- Model Training →
  - Cosine Similarity → Mayank
  - Logistic Regression → Abhishek
  - CNN → Hardik
  - LSTM → Manan

05

IMPLEMENTATION / ANALYSIS

Workflow of the project



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

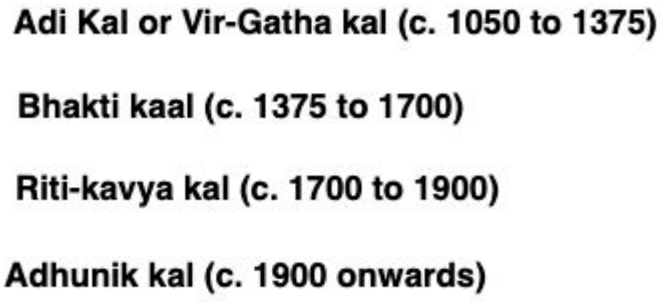
6. Performance Evaluation

Dataset Creation and Merging

- Used **Scrapy** and **BeautifulSoup** to scrap poems from **kavitakosh** and **hindwi** respectively to create a dataset of 50,000 poems.
- **Efforts** in understanding and using the libraries to build the web crawler.

Dataset Creation and Merging (Cont.)

- Dataset may contain many duplicate poems → **Merging**
- Used **year** in which poem was written for classification to create the final corpus.
  Split the poems in **9:1 ratio** to create the training set and test set respectively.



Adi Kal or Vir-Gatha kal (c. 1050 to 1375)

Bhakti kaal (c. 1375 to 1700)

Riti-kavya kal (c. 1700 to 1900)

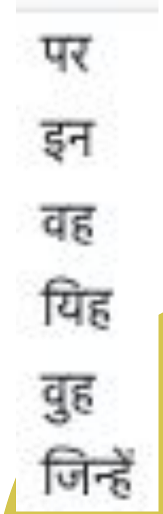
Adhunik kal (c. 1900 onwards)

Various classes of poems

Pre-processing

- **Pre-processed** the data to remove reduce noise in data:
  - Numbers.
  - Punctuations.
  - White spaces.
- Used **tokenization** to split the poems into words. (Used **iNLTK**)
- Removed stop words using 3 different sets of words for optimization.

Stop Words



पर

इन

वह

यिह

वुह

जिन्हें

Vectorization

- **Bag Of Words.**
- **Binary Term Frequency**
- **L1 normalized Term Frequency**
- **L2 normalized tf-idf**

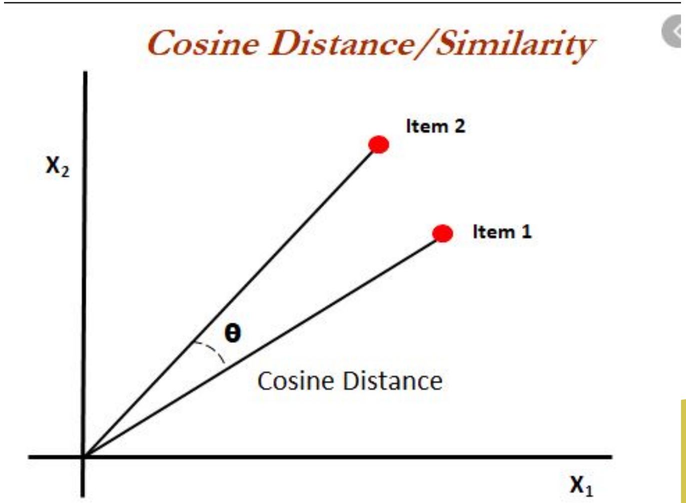
- **These techniques provide long and sparse vectors**

GloVe embeddings

- Used implementation details from https://nlp.stanford.edu/pubs/glove.pdf to create word embeddings for all the words in the vocabulary.
- Used **gradient descent** to find the minimum cost.
- Efforts in implementation and choosing different learning rates to get minimized cost.

Cosine Similarity

- Used **sklearn** to perform cosine similarity.
- For all testing data poems, found the poem in training data with smallest angle and returned its era and author name.
- **Accuracy** → **87.59%** and **16.43%** for era and poet respectively.
- Poet prediction → Why bad?
  - Maybe multiple authors have similar writing styles.
  - Same author has multiple writing styles.



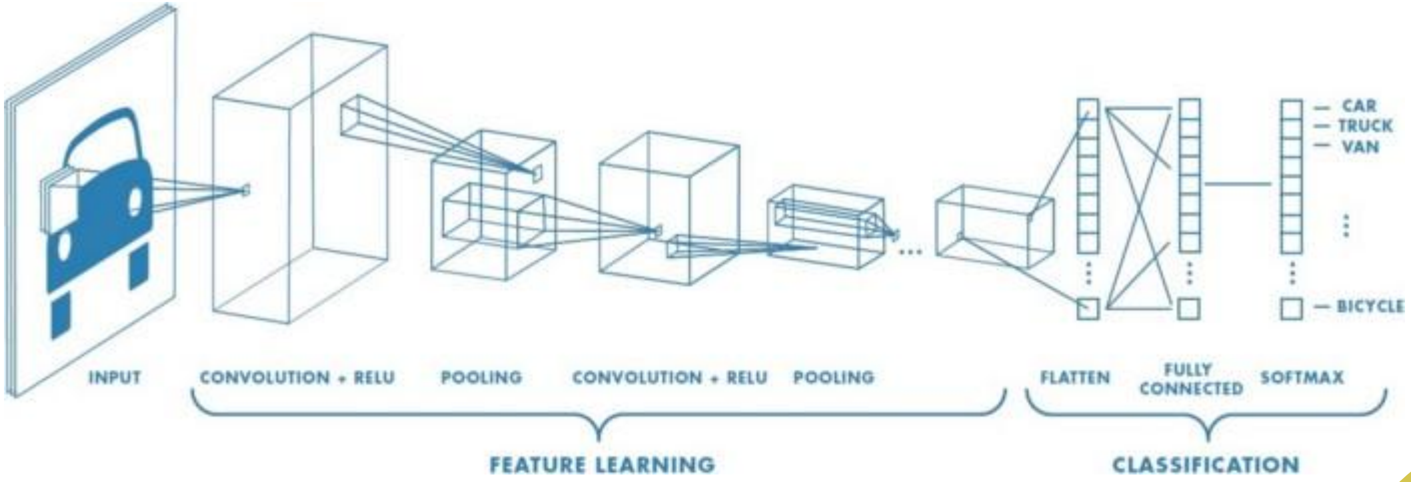
Logistic Regression

- Used **scikit-learn** library, to implement logistic regression with L2 regularization penalty.
- Tried two different implementations of it :-
  - GridSearchCV
  - RandomizedSearchCV
- **Accuracy** -> **89.71%** and **32.11%** for era and poet prediction.

CNN

- Used Keras open-source library.
- Implemented a sequential model.
- Contains Conv1D, MaxPooling1D & 2 Dense Layers.
- Parameters were tuned accordingly.
- Used softmax activation function.

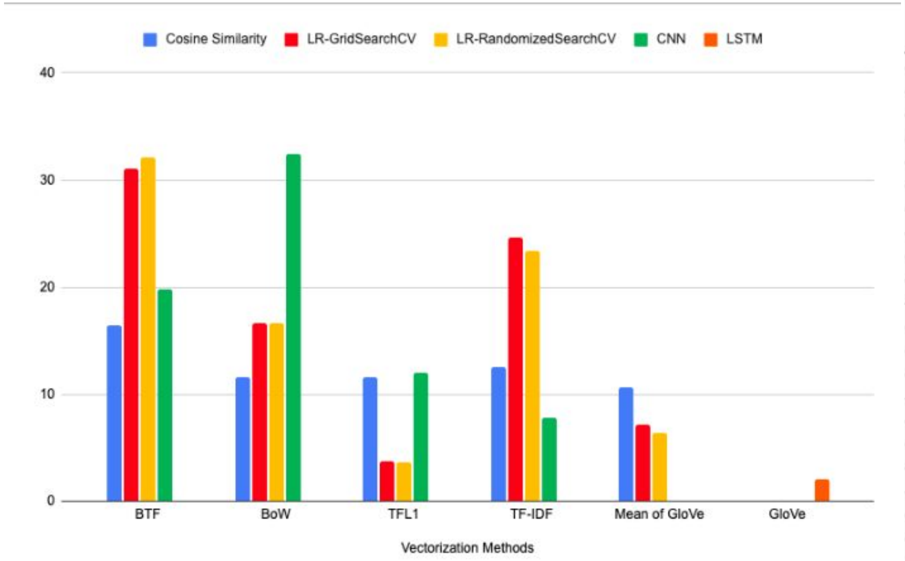
CNN - Implementation Details

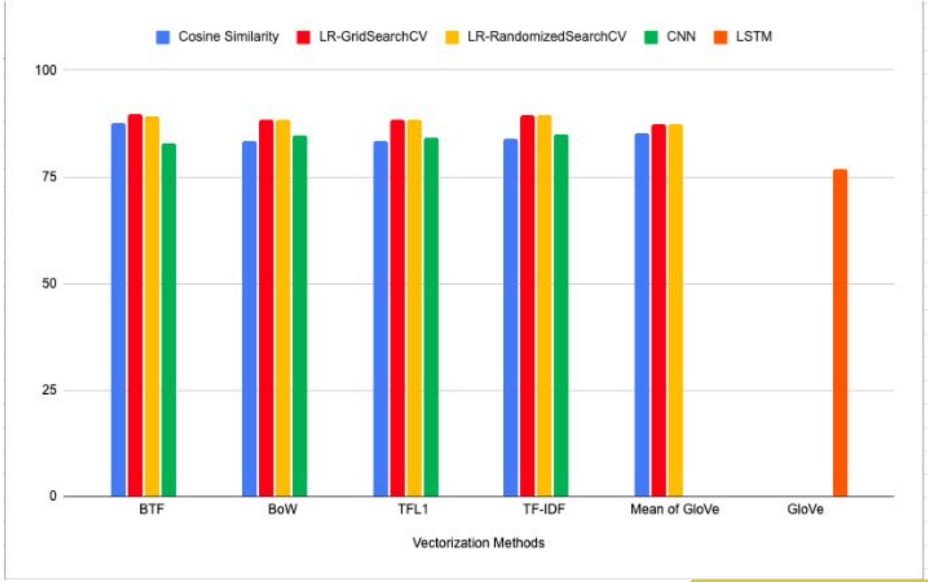


LSTM

- Used Keras open-source library.
- Implemented a sequential model.
- Contains Embedding, LSTM & 2 Dense Layers.
- Parameters were tuned accordingly.
- Used softmax activation function for final layer.
- GloVe embeddings used

Final Results





06

REFERENCES / KEY PAPERS

- **Title :** Automatic poetry classification using natural language processing.
  - Year - 2019
  - Journal - University of Ottawa
  - Author(s) - Vaibhav Kesarwani
- **Title :** Automated Analysis of Bangla Poetry for Classification and Poet Identification.
  - Year - 2015
  - Journal - IITB-Monash Research Academy
  - Author(s) - Geetanjali Rakshit, Anupam Ghosh, Pushpak Bhattacharyya, Gholamreza Haffari
- **Title :** Neural Machine Translation of Rare Words with Subword Units.
  - Year - 2016
  - Journal - School of Informatics, University of Edinburgh
  - Author(s) - Rico Sennrich, Barry Haddow, Alexandra Birch

07

LESSONS LEARNT

LESSONS LEARNT

01
Web Crawling
Implemented web crawlers from scratch

02
NLP
Models & making them work together

03
Expectations
don’t materialize the way we want

04
Complexity
Is not always better.

THANK YOU