English Translator Classification for Shri Guru Granth Sahib Ji

Japneet S. Kohli
George Mason University
jkohli2@masonlive.gmu.edu

Abstract

Shri Guru Granth Sahib Ji (SGGS), the holy scripture for the Sikh faith, has been translated in various languages thus far, including English. An analysis of the authorship of selected English translations is conducted in this paper. This is performed through machine learning (ML) models created to classify the authors of the English translations. Such an analysis would be useful for the administrative bodies in the Sikh faith to verify the authorship claims pertaining to the Sikh scriptures and their translations. In this paper, three predictive models are proposed. Baseline models are created using the Naïve Bayes and Logistic Regression approach. A deep learning model using bidirectional Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) architecture is also proposed. The overall results for the three models are evaluated on the basis of predictive accuracy as measured through the F1 score, which is consistently above 91% for the three models.

Keywords: English Translations, Translator Classification, Author Classification, Shri Guru Granth Sahib Ji, Naïve Bayes, Logistic Regression, bi-LSTM, RNN

1. Introduction

Shri Guru Granth Sahib Ji (SGGS) is the authoritative body of texts that is considered the "living Guru" or teacher by the followers of the Sikh faith. It is a collection of hymns composed by various Gurus of the Sikh faith. The text also includes verses composed by other devotees that were considered to have been "divinely ordained" by the Sikh Gurus. There's a wide breadth of variety in terms of the languages in which these hymns and verses were originally composed. The authors belonged to linguistically diverse areas of India and were separated by time as well, considering that the SGGS corpora has verses all the way from the 15th to the 18th century AD [1].

Considering the wide linguistic and temporal variety in the original text of the SGGS corpora, it is difficult for the lay person to read and understand these texts. Therefore, several scholars have undertaken the task to translate the SGGS corpora into other languages to make it more accessible for the common man. Considering that the Sikh faith got its primary foothold in the Punjab region of India, where the predominant language is Punjabi, the earliest translations were also made in Punjabi. Eventually, the SGGS corpora was translated in other common Indian languages, such as Hindi. As the efforts to translate the SGGS corpora gained momentum, several successful attempts resulted in English translations, the lingua franca of the West and, increasingly, the entire world.

There are currently five English translations of SGGS that have been authorized for use by the Shiromani Gurdwara Prabandhak Committee (SGPC), which is an organization in India responsible for the management of Sikh gurdwaras, or places of worship [2]. Namely, these are the Dr. Gopal Singh Translation, Bhai Manmohan Singh Translation, Gurbachan Singh Talib Translation, Pritam Singh Chahil Translation, and the Dr. Sant Singh Khalsa Translation [3]. For the purpose of this project, the Bhai Manmohan Singh translation and the Dr. Sant Singh Khalsa translation have been analyzed.

It is interesting to find out how the various translations compare with each other. The underlying text of the SGGS remains the same; however, each translator brings their own perspective on how to convey the meaning best. Some authors adhere to strict word-by-word translation, even at the cost of losing grammatical correctness, while others translate with the vision of ease-of-understanding at mind, and therefore, may not strictly adhere to a word-by-word translation, as long as the intended meaning is conveyed. In this regard, the two chosen translations are different from each other. The Bhai Manmohan Singh translation is a little dated as it is from the 1960s while the Dr. Sant Singh Khalsa translation is more modern as it was started in the 1980s and is revised every year to this day. The former has more outdated expressions and vocabulary, whereas the latter tries to mimic the modern English language as much as possible and, so, keeps the verbiage modern as well. [1]

The purpose of this project is to use machine learning (ML) and natural language processing (NLP) techniques to decipher whether the author of a given translation can be extracted by analyzing the translated text. This is an important analysis for the Sikh faith as it could help administrative bodies such as the SGPC to verify the authorship claims pertaining to the Sikh scriptures.

Similar analysis of scriptural corpora has been performed for texts like the Bible and the Quran [4]. However, there is a dearth of statistical analysis of the SGGS corpora.

In this project, three NLP models based on the Naïve Bayes, Logistic Regression, and Bidirectional Long Short-Term Memory (bi-LSTM) Recurrent Neural Networks (RNN) approaches have been used. Although it is difficult to determine the state-of-the-art and the baseline models for this kind of analysis on the SGGS corpora dataset, as this has not been done before, the results of this project are highly encouraging as the predictive models consistently showed an accuracy greater than 90%.

2. Literature Review

The scriptures of the various faiths followed in the world have been thus-far linguistically analyzed using a plethora of different techniques. Sandborg-Peterson analyzed the Old Testament texts in Hebrew and converted them into English-based conceptual graphs [5].

A quantitative analysis of the texts of the Bible was performed by Hajime Murai, in which an attempt was made to interpret the Bible in a scientific manner. To this end, methods such as citation analysis for interpreter's texts, vocabulary analysis for translations, variant text analysis for canonical texts, and evaluation method for rhetorical structure were employed. The author conceded that these techniques were not sufficient to determine the correct interpretations and that narratology and discourse analysis, including the contextual information of the authors such as historical or cultural background, must also be included for a more complete analysis. [6]

Zhao and Liu implemented a domain specific question answering (QA) system for the texts in the Bible using deep learning techniques such as Long Short-Term Memory (LSTM) RNN, Convolutional Neural Network (CNN), and Bi-Directional Attention Flow (BiDAF). The LSTM RNN model was found to be the most accurate with an F1 score of 0.54 and a Mean Reciprocal Rank (MRR) of 0.61. [7]

F. Ahmad et al. used an Articial Neural Network (ANN) architecture comprising of three different back propagation training algorithms, namely Gradient Descent with momentum (Traingdx), Resilient Backpropagation (Trainrp), and Levenberg Marquart (Trainlm) to analyze the audio files consisting the correct pronounciations of Quranic verses based on the prescribed rule of Tajweed. The ANN model was used to classify the Tajweed based on the sounds of the Idgham, which is the sound made when voweled and non-voweled letters meet. The classification accuracy for the Trainlm model was found to be the highest at over 77.7%. [8]

The Quran was also used as the dataset by N.S. Huda et al. for creating a multi-label classification model using a Back Propagation Neural Network, incorporating the Term Frequency – Inverse Document Frequency (TF-IDF) statistic for feature extraction.Topics of English-translated versions of the Quran were used as the classification labels. Hamming Loss was used to evaluate the model, and the best performance was found with the architecture that produced a Hamming Loss value of 0.129. [9]

Another interesting application of data science concepts for capturing information from a religious text is the iPhone mobile application created by M. Alshayeb et al., which uses audio fingerprinting to identify the reciter of Quranic verses. The application uses a recorded clipping of the audio and detects its signal along with the peak intensity of frequency at certain time intervals. These frequencies are used to create an audio fingerprint, which is compared against a previous store of such fingerprints to identify the reciter. The performance of this system is evaluated in terms of the time taken by the application to return the results. This was measured at a maximum of 8 seconds. [10]

As is clear from the above literature review, there is an interest in the academic world to analyze the scriptural texts followed by major faiths of the world. The focus, thus far, has been on the texts of the Western world. With this project, an attempt is made to foray into the scriptural texts of the Eastern world, with a specific focus on the SGGS English translations. Two baseline model using the Naïve Bayes and Logistic Regression approach are complemented with a deep learning model using bi-LSTM. Each model classifies the author of the translation, and the accuracies of the models are compared to find the best fit.

3. Data Description

To acquire the data for this project, the ShabadOS API was used. ShabadOS is an open source, collaborative Gurbani database [12] that is maintained on GitHub [13]. The original text of the SGGS corpora comprises of 1,430 pages containing 5,894 verses. ShabadOS stores these documents in over 1,391 JSON files.

The data consists of attributes such as the writer of the original text, section, subsection, line type, line translation, translation language, translation author, and others. The data described included each verse of the SGGS corpora line-by-line, with the original lines in the Gurmukhi script in Punjabi. Table 1 below shows the various translations available in the dataset.

Table 1: SGGS Corpora Translation Details

Translation Language	Author of the Translation/Translator
Punjabi	Bhai Manmohan Singh, Fareedkot Teeka, Professor Sahib Singh
English	Dr. Sant Singh Khalsa, Bhai Manmohan Singh
Spanish	SikhNet

The data comprises of over 121,020 lines, or rows, of data. Half of these lines, 60,510, belong to each English translator, namely Dr. Sant Singh Khalsa and Bhai Manmohan Singh. The original writers of the SGGS corpora were also represented in varying proportions in the dataset, as shown in Table 2.

Table 2: Distribution of Lines by Original Author

Original Author	Number of Rows/Lines
Guru Arjan Dev Ji	49,746
Guru Nanak Dev Ji	24,078
Guru Amardas Ji	20,230
Guru Ramdas Ji	13,086
Bhagat Kabir Ji	6,770
Bhagat Namdev Ji	1,502
Guru Tegh Bahadur Ji	1,182
Bhagat Ravidas Ji	924
Sheikh Farid Ji	636
Guru Angad Dev Ji	614
Bhatt Kalh Sahar	566
Sattaand Balwand	180
Bhagat Beni Ji	178
Bhatt Nalh	168
Bhatt Gayand	160
Bhatt Mathura	128
Poet Alam	124
Bhagat Trilochan Ji	114
Bhatt Keerat	80
Baba Sundar	76
Bhatt Jaalap	62
Bhagat Dhanna Ji	62
Bhatt Balh	52
Bhagat Jaidev Ji	46
Bhatt Bhikha	38
Bhagat Ramanand Ji	32
Bhagat Bheekhan Ji	32
Bhatt Harbans	24
Bhatt Salh	24
Bhagat Sadhna Ji	24
Bhagat Sain Ji	22
Bhagat Parmanand Ji	20
Bhagat Surdas Ji	18
Bhagat Pipa Ji	14
Bhatt Bhalh	8

It is apparent from Table 2 that the highest number of lines were written by Guru Arjan Dev Ji. This is consistent with the SGGS corpora, as Guru Arjan Dev Ji was, indeed, the most prolific among the SGGS writers and contributed the highest number of verses to the SGGS corpora.

4. Methodology

The 1,391 JSON files that were acquired from the ShabadOS API were converted into 1,391 CSV files, for each of which the JSON tags were converted into CSV columns. The Pandas library in Python was found to be insufficient in unnesting the JSON tags successfully and completely. Therefore, alternative solutions for this purpose were tried, with the JSON to CSV Converter API [14] showing the best results. This API was used to convert the JSON files to CSV successfully. The API implements daily limits of 1MB for usage. Therefore, the conversion was completed in batches over several days.

The English translations shown in Table 1 were of direct interest for this project. Using the Pandas library in Python, the attributes representing only the English translation and author information, along with the line type and author of the original text, were collected for all 1,391 CSV files, and stored in a single file so that it could be used as the final, singular dataset for the project.

The distribution of lines by the original author, as shown in Table 2, helped confirm that the several data conversion steps that had been followed thus far in acquiring the data were successful in capturing the required data correctly.

Data was then preprocessed using regular expressions to remove punctuations, any non-ascii characters, double spaces, and HTML tags. Capitalized words were lower cased, unless they were in all caps. This preprocessing was applied to the line translation attribute. The original author and line type attributes were dropped from the dataset as they were not required for training the ML models.

After this preprocessing step, only two attributes remained: the translator name, which would become the classification label for the ML models, and the line translations, which would become the data used for classification in the ML models.

Table 3: Size of Train, Test, and Validation Datasets

Dataset	Size in Rows
Train	77,452
Validation	19,364
Test	24,204

The dataset was then divided into training, validation, and testing sets. 20% of the entire dataset was assigned for testing, while 20% of the remaining data was used for validation. The remaining data was used for training. The row distribution for these datasets is provided in Table 3.

The final dataset, with train-test-validation splits in place, was ready to be fit into ML models. For this project, three different models were created, namely the Naïve Bayes, the Logistic Regression, and the bi-LSTM RNN models.

For the Naïve Bayes and Logistic Regression models, word features were extracted using TF-IDF. To classify the translators, the Multinomial Naïve Bayes classifier was used in the Naïve Bayes model. The Logistic Regression model used the eponymous classifier instead.

10-fold cross validation was performed on both these models using the GridSearchCV method. The results of this cross validation helped determine the hyperparameters to be used for the two models. These are outlined in Tables 4 and 5 below.

Table 4: Hyperparameters for Naive Bayes Classification Model

Method	Hyperparameter Value
N-gram Range for Count Vectorizer for Tokens	Unigrams and Bigrams
Use IDF Decision in TF-IDF Transformer	False
Norm Value to Normalize for TF-IDF Transformer	L2
Alpha Value for Multinomial Naïve Bayes	0.1

Table 5: Hyperparameters for Logistic Regression Classification Model

Method	Hyperparameter Value
N-gram Range for Count Vectorizer for Tokens	Unigrams and Bigrams
Use IDF Decision in TF-IDF Transformer	True
Choice of Solver for Logistic Regression	Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) Algorithm
Maximum Iterations for Solvers to Converge	220
Multi Class Scheme Choice for Logistic Regression	Multinomial

The third model that was created was a deep-learning model using bi-LSTM RNN. For this model, it was required that the text of the line translation attribute be arranged in matrix representation so that this data could be fit in a bi-LSTM RNN model.

Towards this end, each word in the text of the line translation attribute was converted into a token so as to keep track of the total vocabulary of the text. The word values for each row of the line translation attribute were replaced by their token index and this sequence of numbers was then padded to ensure that the sequences were all the same length. This was done to ensure that the matrix created to store the line translation attribute would be able to accommodate the longest lines.

Next, the word embeddings were used to represent each token in a sentence in the form of a vector representation. For this purpose, pretrained embeddings were used to create the embedding matrix for this model. The Global Vectors for Word Representation (GloVe) algorithm was leveraged for this purpose as it curates and stores word-embeddings as determined by an unsupervised learning algorithm that obtains vector representations for words [15]. The pretrained embeddings had embeddings for over 6 billion words in 100 dimensions.

The resulting embedding matrix that was created for the tokens had 100 columns, representing the 100 dimensions of the GloVe pretrained embeddings, and 14,295 rows, representing the number of tokens or, in other words, the size of the vocabulary in the text of the line translation attribute.

The network architecture of the bi-LSTM RNN model was instantiated using a tensor with as many dimensions as found in the padded tokenized sentences of the line translation attribute. This was the maximum length of the sentences observed in the text, and its value was 266.

The embedding layer was then added to the architecture to convert the tokenized sentences into dense vectors of fixed size using the weights resulting from the embedding matrix created previously. The next steps incorporated a hidden bi-LSTM layer with 100 nodes and an activation layer using the Rectified Linear Unit (RELU) activation function.

At this point of the architecture, a dropout layer was specified to dropout random units at a rate of 0.8, and then a dense layer was specified to reduce the dimensionality of the output to 1. Another activation layer was then added, using the Sigmoid activation function for the binomial classification.

The model thus created was compiled with the Adam optimizer set at a learning rate of 0.01. The loss for this model was set to be calculated using binary cross-entropy and the metric for evaluation was set to accuracy.

The model was trained with a batch size of 128. Although 20 epochs were specified, early stopping was also put in place, conditioned on the increasing value of loss.

For the bi-LSTM RNN model, a different training and testing dataset split was used vis. a vis. the Naïve Bayes and Logistic Regression models. Here, the train and test datasets were splits at 80% and 20% respectively.

The model was trained on half the training data; the remainder of the training data was used for validation.

5. Results and Discussion

After the training of the three models developed to classify the translators of the English translations of the SGGS was complete, the models were run upon the test dataset to make predictions. The predictions were compared with the actual value of the test data labels. Based on the predicted and actual values, it was possible to calculate the classification accuracy of the three models. The confusion matrices of these results for the Naïve Bayes and the Logistic Regression model are shown in Tables 6 and 7. Both the models show a very similar result in terms of the predictions.

Table 6: Confusion Matrix for the Naive Bayes Model

Actual / Prediction	Dr. Sant Singh Khalsa	Bhai Manmohan Singh
Dr. Sant Singh Khalsa	11,061	945
Bhai Manmohan Singh	997	11,201

Table 7: Confusion Matrix for the Logistic Regression Model

Actual / Prediction	Dr. Sant Singh Khalsa	Bhai Manmohan Singh
Dr. Sant Singh Khalsa	11,058	948
Bhai Manmohan Singh	997	11,201

To evaluate the performance of these models, an evaluation of their classification accuracy was performed. The F1 score was used for this purpose. The F1 score takes into account the precision and recall statistics to present an overall score for the accuracy of predictions. These scores are tabulated in Table 8 below.

Table 8: Comparison of F1 Score for ML Models

ML Model	F1 Score (Accuracy)
Naïve Bayes	0.9198
Logistic Regression	0.9195
bi-LSTM RNN	0.9175

As can be observed from Table 8, the F1 score for the three models is extremely close, separated by only the third and fourth decimal points between models. Also, the number for this accuracy is between 91.75% and 91.98%, which is an indicator of a very high prediction accuracy.

The best model based on the F1 score is the Naïve Bayes model as it slightly outperformed the other two models in terms of accuracy. However, having said that, it may be possible to increase the accuracy of the bi-LSTM RNN model by changing the architecture and incorporating other methods in the layers.

For future analysis, it would be interesting to see other implementations for predicting the translator from translated Sikh literature. It would also be interesting to find out if the models trained on the SGGS corpus to predict the translator are successful at predicting translators for other texts outside the SGGS corpus that are still considered part of the Sikh pantheon of literature that has been translated by the same translators. Such a study would provide a window on whether the translations were influenced by any features of writing that are carried across translated writings. While such features, including the writing style, sentence construction, word choice, etc. may be expected to remain constant for a writer, there may be a possibility to inspect whether these underlying core instruments of putting thought in writing form are unduly influenced when the writing being done is a translation and not original work.

Other avenues for future research in this area include the determination of authority of texts that are in dispute in the Sikh pantheon. The authorship of several of the texts purportedly composed by the tenth Guru, Guru Gobind Singh Ji are debated. An exercise in analyzing the composition of these texts based on the principles of data science may be additive to the historic evidence that is frequently debated.

6. References

[1] Sikhs.org, "History of Sri Guru Granth Sahib," 2011. [Online]. Available: https://www.sikhs.org/granth1.htm. [Accessed 15 11 2020].

[2] SGPC, "About SGPC," 2020. [Online]. Available: http://sgpc.net/about-sgpc/. [Accessed 15 11 2020].

[3] S. S. Khalsa, "Siri Guru Granth Sahib English Translation Comparison," 2020. [Online]. Available: https://www.sikhnet.com/pages/siri-guru-granth-sahib-english-translation-comparison. [Accessed 15 11 2020].

[4] R. C. Popa, N. Goga and M. Goga, "Extracting Knowledge from the Bible: A Comparison between the Old and the New Testament," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, London, 2019.

[5] U. Sandborg-Peterson, "Genesis 1:1-3 in Graphs: Extracting Conceptual Structures from Biblical Hebrew," 01 2007. [Online]. Available: https://www.researchgate.net/publication/228734851_Genesis_1_1-3_in_Graphs_Extracting_Conceptual_Structures_from_Biblical_Hebrew. [Accessed 15 11 2020].

[6] H. Murai, "Introducing Scientific Methods for the Interpretation of the Bible: Quantitative Analysis of Christian Documents," Kyoto, 2012.

[7] H. J. Zhao and J. Liu, "Finding Answers from the Word of God:

Domain Adaptation for Neural Networks in Biblical Question Answering," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, 2018.

[8] F. Ahmad, S. Z. Yahya, Z. Saad and A. R. Ahmad, "Tajweed Classification Using Artificial Neural Network," in *2018 International Conference on Smart Communications and Networking (SmartNets)*, Yasmine Hammamet, Tunisia, 2018.

[9] N. S. Huda, M. S. Mubarok and Adiwijaya, "A Multi-label Classification on Topics of Quranic Verses (English Translation) Using Backpropagation Neural Network with Stochastic Gradient Descent and Adam Optimizer," in *2019 7th International Conference on Information and Communication Technology (ICoICT)*, Kuala Lumpur, Malaysia, 2019.

[10] M. Alshayeb, A. Hakami, A. Altokhais, A. Almousa, M. Albarrak and O. Alessa, "Towards the Identification of Quran Reciters," in *2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, Madinah, 2013.

[12] ShabadOS, "@shabados/database," [Online]. Available: https://docs.shabados.com/database/#/. [Accessed 15 11 2020].

[13] ShabadOS, "Shabad OS," 02 10 2017. [Online]. Available: https://github.com/shabados/database. [Accessed 15 11 2020].

[14] JSON-CSV.com, "JSON to CSV API," JSON-CSV.com, 2012. [Online]. Available: https://json-csv.com/api. [Accessed 15 11 2020].

[15] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," 08 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed 15 11 2020].