JU_NLP at HinglishEval: Quality Evaluation of the Low-Resource Code-Mixed Hinglish Text

Prantik Guha[1] and **Rudra Dhar**[2] and **Dipankar Das**[3]
Department of Computer Science & Engineering, Jadavpur University, Kolkata, India
{[1]prantikguha706, [2]rudradharrd, [3]dipankar.dipnil2005}@gmail.com

Abstract

In this paper we describe a system submitted to the INLG 2022 Generation Challenge (GenChal) on Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text. We implement a Bi-LSTM-based neural network model to predict the Average rating score and Disagreement score of the synthetic Hinglish dataset. In our models, we used word embeddings for English and Hindi data, and one hot encodings for Hinglish data. We achieved a F1 score of 0.11, and mean squared error of 6.0 in the average rating score prediction task. In the task of Disagreement score prediction, we achieve a F1 score of 0.18, and mean squared error of 5.0.

1 Introduction

In India, social media's enduring popularity has resulted in massive amounts of user-generated textual content. During a conversation, multilingual speakers frequently flip between languages. Speakers frequently talk in multiple languages, and often transliterate. Listeners may not always be able to keep up with the multilingual speakers. That's why we need automated systems for transliterated translations.

But we don't have a significant amount of transliterated translation data to train our models. So we might use synthetic data for this purpose. Synthetic data has become a common resource for a variety of applications. It may be required because of data unavailability, cost savings, security, or privacy concerns. Because synthetic data matches the statistical properties of production data, it can be used to train models, validate models, and evaluate performance. Machine learning models have now made it possible to create incredibly fast natural language generating systems by building and training a model.

Now the next challenge is to evaluate the data which is synthetically generated. In this paper we have introduced an algorithm to check the quality of the generated data. We have proposed a supervised learning model using multiple Bi-LSTM and dense layers to predict two types of scores (Average Rating score and Disagreement score). In this paper we are using the data from Srivastava and Singh (2021a).

This is a transliterated translation verification problem which essentially boils down to a task of document similarity evaluation. Document similarity evaluation is a well researched task in NLP. As Merlo et al. (2003) suggests, various Machine learning techniques, and Natural Language Processing tools can be used for this purpose. Linhares Pontes et al. (2018) shows us how hybrid models of LSTM's can be used for document similarity prediction. Some work has also been done in the multilingual senario, as in Wang et al. (2018). However not much work has been done in transliterated translation verification, and certainly none has been done in the Indian domain. Srivastava and Singh (2020) explains the challenges in both generating transliterated translations and evaluating it.

2 Dataset

The phenomena of code-mixing are the mingling of words and phrases from various languages in a single text or spoken utterance. Examples of code-mixed Hinglish sentences created from parallel Hindi and English utterances are shown in Fig-1.

In this shared task, there are two subtasks for evaluating the quality of the code-mixed Hinglish text in this common task (Srivastava and Singh, 2021b). In the first sub-task, they proposed using a scale of 110 to determine the quality of Hinglish content. They want to figure out what elements influence text quality, so high-quality code-mixed text generating systems can be created. The second sub-task is to predict how much the two annotators

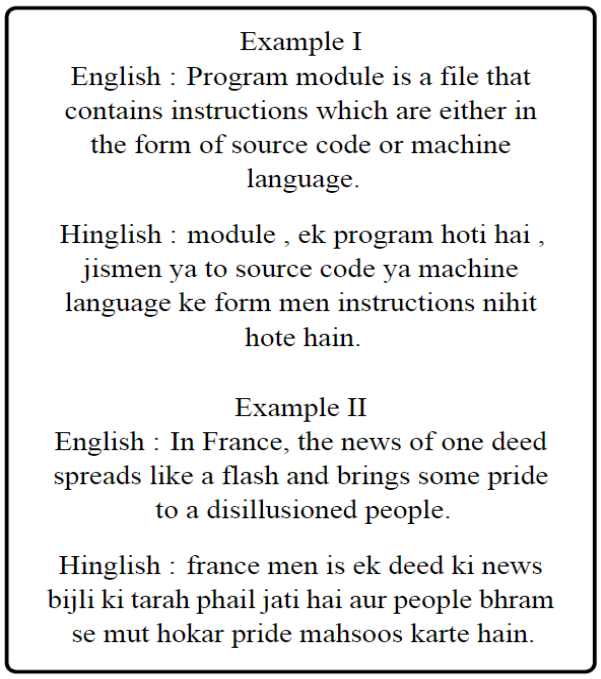

Figure 1: Example from (Srivastava and Singh, 2021a) data

who annotated the synthetically generated Hinglish sentences differ on a scale of 09. Various factors influence human disagreement.

The dataset consists of five columns (English, Hindi, Hinglish, Average Rating, Disagreement). Highlish sentences are generated using two rule-based algorithms (i.e., WAC and PAC). For the two rating columns (Average Rating & Disagreement) each sentence is rated on a scale of 1(low-quality) to 10 (high-quality) by two annotators. The quality of the synthetically generated sentences is calculated by rounding off the average of the two human ratings and using this score (in the range of 1-10) in the Average rating column. And the Disagreement score is calculated by the absolute difference of the two human ratings as the disagreement score (in the range of 0-9).

3 System Description

We used a sequence of Glove embeddings as input for English and Hindi sentences. However, for Hinglish sentences we used one hot vector as inputs. We fed the English and Hindi embeddings to separate Bi-lstm's[l-e, l-h], and retrieved sequence output from them. To capture the word sequences of different Hindi and English sentences we have used two different LSTMs. Then we concatenated these 2 outputs and passed it through another Lstm layer to get a fixed (not sequence) vector output [l-h-e].

We fed the one hot vector from the Hinglish data to a dense layer and received a vector output [d-he]. Since one hot vector does not capture the sequential information, we have used a dense layer. We then concatenated these two [l-h-e and d-he] vectors, and passed it through a dense layer to get a final class (score between 1 to 10). We used the same model for both the tasks. Please refer to Fig-2 for complete system architecture.

No. of data	F1-Score	Cohen's Kappa	Mean Squared Error
395	0.09899	-0.01521	6.00

Table 1: This result is obtained from 395 validation data for Sub-task 1(Average rating score)

No. of data	F1-Score	Mean Squared Error
395	0.21622	5.00

Table 2: This result is obtained from 395 validation data for Sub-task 2(Disagreement score)

4 Training

On a total of 2766 training data points, we train the LSTM model using the Adam optimizer with a batch size of 32. Started with loss of 0.1810 & accuracy of 0.9658. In the final epoch loss was 0.0300 & accuracy was 0.9864.

In this phase, we validated the input using our developed model. For this phase the total available data was 395. We have validated our model for both Average Rating as well as Disagreement.On 395 data we validated our system to predict Average rating for corresponding inputs. Please refer to Table: 1 for detailed results related to this validation. On 395 data we validated our system to predict Disagreement score for corresponding inputs. Please refer to Table: 2 for detailed results related to this validation.

5 Test

In this phase, our developed model gets tested on test data. For this phase the total available data was 791. Model was tested for both Average Rating as well as Disagreement.

On 791 data, our system is able to predict Average rating for corresponding inputs. Please refer to

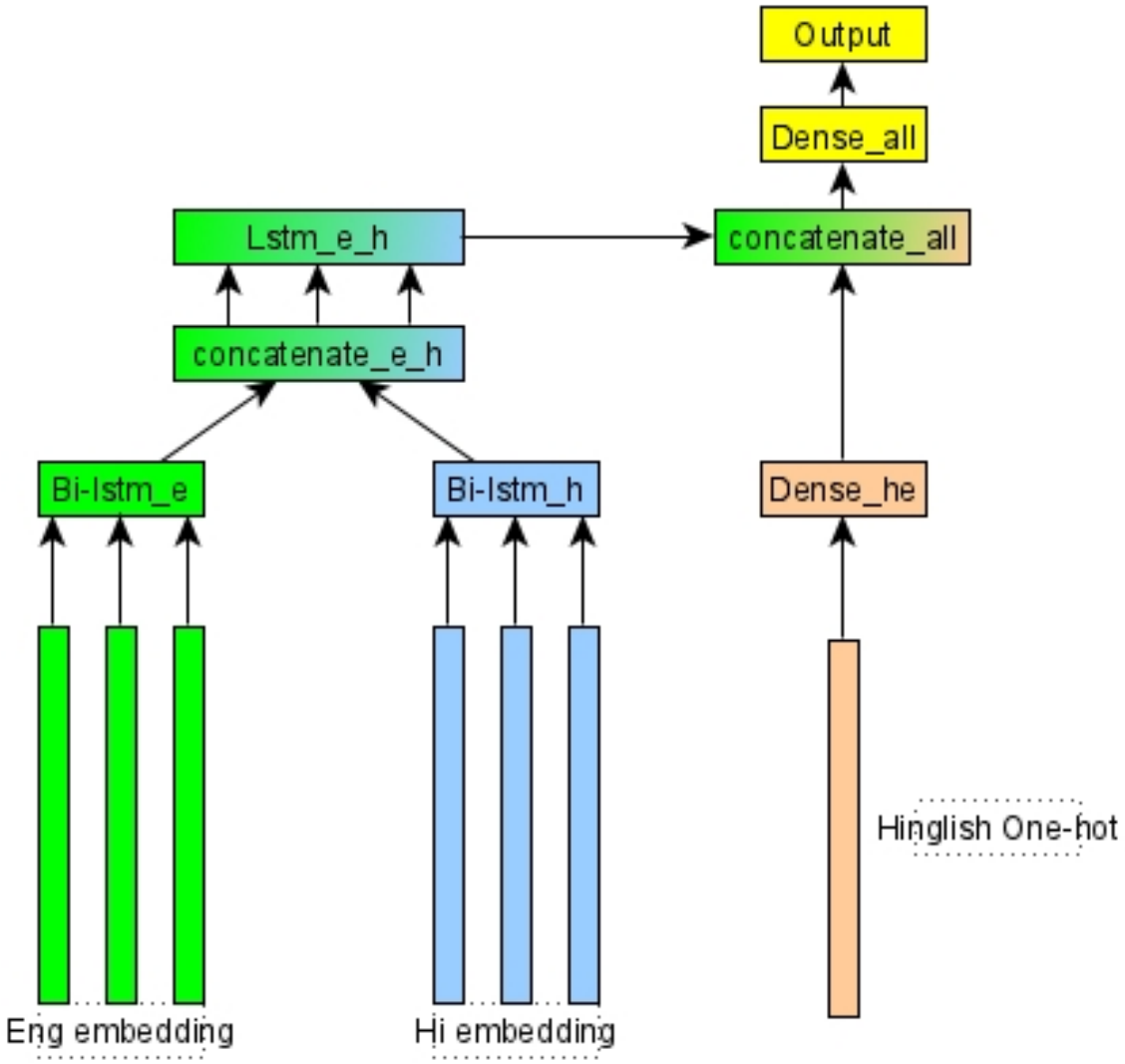

Figure 2: system architecture

No. of data	F1-Score	Cohen's Kappa	Mean Squared Error
791	0.11582	0.00337	6.00

Table 3: This result is obtained from 791 test data for Sub-task 1(Average rating score)

No. of data	F1-Score	Mean Squared Error
791	0.18331	5.00

Table 4: This result is obtained from 791 test data for Sub-task 2(Disagreement score)

Table: 3 for detailed results related to this validation. On 791 data, our system is able to predict Disagreement score for corresponding inputs. Please refer to Table: 4 for detailed results related to this validation.

6 Conclusion

For INLG 2022, we created a system to predict the Average Rating of synthetically generated Hinglish sentences (Sub-Task 1) & Disagreement score for the same (Sub-Task 2). We didn't use any outside information. We have used GLOVE embedding for English and Hindi sentences. And for Hinglish sentences we have used multi label vectors.

References

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian

social media text.

Jinyu Li, Sibel Yaman, Chin-hui Lee, Bin Ma, Rong Tong, Donglai Zhu, and Haizhou Li. 2006. Language recognition based on score distribution feature vectors and discriminative classifier fusion. In *2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*, pages 1–5.

Elvys Linhares Pontes, Stéphane Huet, Andréa Linhares, and Juan-Manuel Torres-Moreno. 2018. Predicting the semantic textual similarity with siamese cnn and lstm.

Ruibo Liu, Jason Wei, and Soroush Vosoughi. 2021. Language model augmented relevance score. *arXiv preprint arXiv:2108.08485*.

Paola Merlo, James Henderson, Gerold Schneider, and Eric Wehrli. 2003. Learning document similarity using natural language processing. *Linguistik online*, 17.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017. *arXiv preprint arXiv:1803.06745*.

Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018. Word embeddings for code-mixed language processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3067–3072.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021a. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed Hinglish text. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 314–319, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2022. Code-mixed nlg: Resources, metrics, and challenges. In *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 328–332.

Zhouhao Wang, Enda Liu, Hiroki Sakaji, Tomoki Ito, Kiyoshi Izumi, Kota Tsubouchi, and Tatsuo Yamashita. 2018. Estimation of cross-lingual news similarities using text-mining methods. *Journal of Risk and Financial Management*, 11(1).

Siddharth Yadav and Tanmoy Chakraborty. 2020. Unsupervised sentiment analysis for code-mixed data. *arXiv preprint arXiv:2001.11384*.