Neural-based Tamil Grammar Error Detection

Murugesapillai Dineskumar
University of Moratuwa, Sri Lanka
170141X@uom.lk

Ravinthirarasa Anankan
University of Moratuwa, Sri Lanka
170032N@uom.lk

Kenatharaiyer Sarveswaran
University of Moratuwa, Sri Lanka
sarves@cse.mrt.ac.lk

Gihan Dias
University of Moratuwa, Sri Lanka
gihan@uom.lk

Abstract

This paper describes an ongoing development of a grammar error detector for the Tamil language using the state-of-the-art deep neural-based approach. This proposed checker captures a vital grammar error called subject-predicate agreement errors. In this case, we specifically target the agreement error between nominal subjects and verbal predicates. We also created the first-ever grammar error annotated corpus for Tamil. In addition, we experimented with different multi-lingual pre-trained language models to capture syntactic information and found that IndicBERT gives better performance for our tasks. We implemented this grammar checker as a multi-class classification on top of the IndicBERT pre-trained model, which we fine-tuned using our grammar-error annotated data. This baseline model gives an F1 Score of 84.0. We are now in the process of improving this proposed system with the use of a dependency parser.

1 Introduction

Grammar error detection is the task of identifying grammatical errors in the text. This feature is available as a part of stand-alone applications, such as Microsoft Word, Libre Office, and online applications, such as Grammarly and Google Docs. However, none of these applications supports the grammar error detection of Tamil and most other Indian languages.

In recent times, neural-based approaches are also being employed for grammar error detection tasks. However, unlike other well-resourced languages such as English and German, applying neural-based approaches to Tamil is difficult due to the lack of quality annotated data.

This paper outlines how we implemented an application to detect grammar errors related to the subject-predicate agreement in Tamil. We have created a grammar error annotated corpus to train the application. We have employed a neural-based approach and a transfer learning technique to implement the proposed application.

2 Motivation

Tamil is a morphosyntactically rich and free-order language. It is spoken by more than 78 million people around the world,[1] and is the official language of Sri Lanka, Singapore, and Tamil Nadu, India. Tamil is a diglossic language with a spoken and written form. The spoken form varies from region to region; however, the written form is almost the same among regions. Tamil documents are being prepared electronically nowadays, including official documents. However, most of the time, these documents are typed in by people who are not well versed with Tamil grammar.

On the other hand, official government documents are not supposed to have any grammar mistakes. It is even more critical in a multi-lingual country such as Sri Lanka, where sometimes even a non-Tamil person may type in official letters. Therefore, all documents have to be checked for all types of errors and corrected. Further, nowadays, many efforts are being made to develop machine translation systems. We need to ensure that translations are grammatically correct before using them to train systems.

3 Literature review

Several studies have been carried out to develop spell checkers for Tamil, including (Sakuntharaj and Mahesan, 2016, 2018, 2019; Segar and Sarveswaran, 2015; Uthayamoorthy et al., 2019; Rajendran, 2012). However, no grammar error checkers are found online or integrated into other

[1] http://www.languagesgulper.com/eng/Tamil.html

Proceedings of the ICON 2021 Workshop on Parsing and its Applications for Indian Languages (PAIL) , pages 27–32
December 16, 2021. Silchar, India.

applications. On the other hand, grammar error checkers for well-resourced languages are readily available online as cloud-based tools such as Google Docs, Grammarly, and stand-alone office suites.

There are 28 different types of errors that have been reported in literature (Ng et al., 2014). In addition to the listed errors, Tamil also has a special type of error called *Sandhi* error. Although *Sandhi* is considered as the result of a phonological operation between two words or two morphs, *Sandhi* also shows syntactic clues as discussed shown by Sarveswaran and Butt (2019). In this respect, *Vaani*[2], which is developed using a rule-based approach, can be considered as a partial grammar checker as it handles *Sandhi* errors.

Nowadays, several multi-lingual pre-trained models(Conneau and Lample, 2019; Conneau et al., 2019; Xue et al., 2021; Kakwani et al., 2020; Devlin et al., 2019) are available online. These models are trained with millions of sentences and tokens. The pre-trained models capture various linguistic information, including morphological, syntactical and semantic information of sentences. However, these details are not in a specific format; therefore, not very easy to retrieve. XLM-R (Conneau et al., 2019), IndicBERT (Kakwani et al., 2020), and BERT (Devlin et al., 2019) are also trained with Tamil data. Therefore, these models also capture linguistic features of Tamil.

We require a large set of annotated corpus to train a machine learner to carry out the task of our interests. However, the Tamil language does not have an error annotated corpus. This kind of error annotated corpora can be created not only by hand but also with the assistance of tools like Part of Speech taggers, morphological analysers, and syntactic parsers.

4 The proposed grammar error detector

This section outlines the process that has been followed to develop the proposed grammar error detector using a neural-based approach and transfer-learning technique.

4.1 Scope

We handle only the modern written Tamil text. Because Tamil is a diglossia language that evolved over several millennia, even the spoken forms vary significantly among different regions. Therefore, it is complicated to draw grammar rules for them. Further, over time Tamil also underwent several grammatical changes. Therefore, we decided to focus only on the modern text that was written after 2000. We collected text from this period for training, evaluation, and testing of the proposed grammar checker.

Further, instead of considering all the grammar errors, we handle only the type of error called subject-predicate agreement. In Tamil, the subject-predicate agreement is an important condition that needs to be met for any sentence to be grammatical. Tamil can have nominated subjects and non-nominative subjects. However, in our case, we focus only on nominative subjects as there is no agreement between non-nominative subjects and the verbal predicates. Similarly, we do not handle a nominal predicate as there are no agreements between a subject and the nominal predicates. Therefore, our focus is only on the nominative subject-verbal predicate agreement where both of them need to agree on gender, number and person. Even if one of these does not match, it is considered a grammar error. Although this agreement needs to be held on rationality, we do not handle it separately as rationality errors can be tracked using person, number, and gender errors.

4.2 Data

Except for a spelling error annotated word list,[3] which is tiny in size, there was no other error annotated list found online. Therefore, we created a grammar annotated dataset that marks subject-predicate agreement errors, specifically person, number, and gender errors. Table 1 shows details of our corpus. The dataset has 5546 sentences taken from news sources. We decided to use this to develop a baseline system and then get the baseline system to generate more error annotated datasets incrementally.

The task of grammatical error detection is formalized as such, given Tamil sentence **X** as input, the error detector outputs its prediction **Y** where,

$$Y = \begin{cases} 0, & \text{if X is correct.} \\ 1, & \text{if X has gender error.} \\ 2, & \text{if X has person error.} \\ 3 & \text{if X has number error.} \end{cases}$$

The dataset we collected has been divided into training, validation, and testing sets, containing 4645, 460, and 481 sentences. It is non-

[2] http://vaani.neechalkaran.com/

[3] https://www.kaggle.com/neechalkaran/error-annotated-tamil-corpus

Table 1: Size of each class in the dataset

Class	Number of sentences			Total
	Train	Validation	Test	
grammatical	2455	120	121	2696
person	913	120	120	1153
number	772	120	120	1012
gender	505	100	120	725
Total	**4645**	**460**	**481**	**5546**

Table 2: Example data entries from our error annotated corpus

Erroneous sentence	Errorless sentence	Error type
கவிதா வந்தான் . kavitā vantān Kavitha.NOM-**3SgF** come.**3SgM** .	கவிதா வந்தாள் . kavitā vantāḷ Kavitha.NOM-**3SgF** come.**3SgF**.	Gender
நான் நாளை வந்தாள் . nān nāḷai vantāḷ I.NOM-**1Sg** come.**3Sg** .	நான் நாளை வருவேன் . nān nāḷai varuvēn I.NOM-**1Sg** come.**1SgF**.	Person

overlapping and balanced in terms of the type of errors. Table 2 shows two example entries of error annotated corpus, a number error and a gender error.

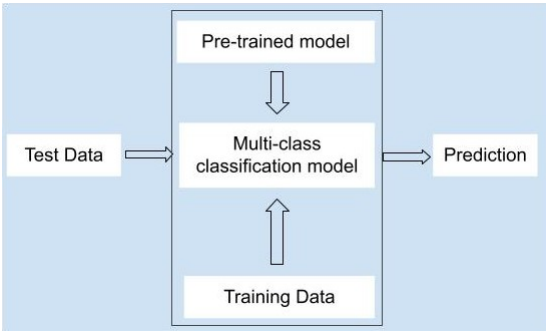

Figure 1: Overview of methodology

5 Approach

As illustrated in Figure 1, we used a supervised approach to develop the proposed grammar error detector. However, instead of training a model from scratch, which requires a significant amount of data and processing power, we used a pre-trained language model to capture the morphosyntax and then modelled the grammar error detection as a multi-class classification problem on top of it. In order to do that, we have created a grammar error annotated corpus to fine-tune the pre-trained model and implement our classification model. We used our training set and validation set for this purpose, and then we evaluated the system using the test set.

5.1 Identifying a pre-trained model

As a first step, we have identified a pre-trained model which works better for our problem. We experimented with XLM-R (Conneau et al., 2019), IndicBERT (Kakwani et al., 2020), and mBERT (Devlin et al., 2019). Table 3 shows the comparison of these models in respective to their token size, parameters, and test results as reported by Kakwani et al.,(2020). We made use of a framework called Simple Transformers[4] to carry out our experiments.

The Simple Transformer framework provides supports for various pre-trained models and tasks such as text classification, token classification, question answering, and language modelling. We can easily set up a classification layer on top of the pre-trained model using this framework. Further, this framework also supports changing various parameters, including learning rate, batch size, and epochs.

We fine-tuned the given three models using our error annotated corpus and by varying different parameters as shown in Table 5. Finally, we also evaluated the model using the test set.

Table 4 shows the results we obtained for all three models, and from which it is clear that the IndicBERT pre-trained model outperforms other models with the F1 score of 73.4%. Therefore, we

[4] https://simpletransformers.ai/

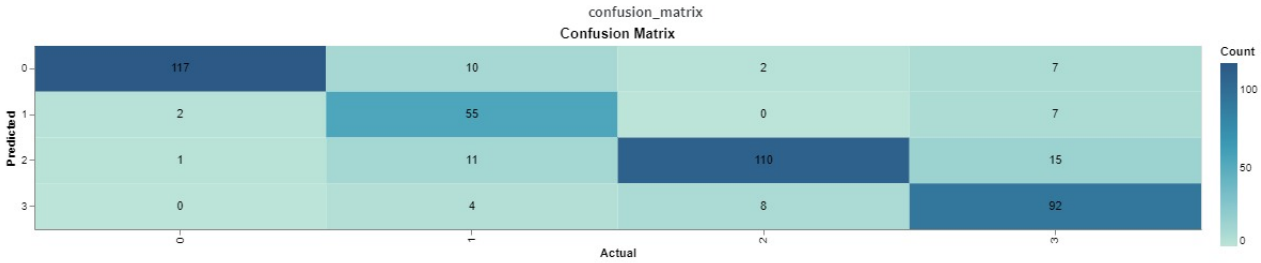

Figure 2: confusion matrix

Table 3: Pre-trained models, token size, number of parameters, and the test results for different tasks in indicGLUE - Source: (Kakwani et al., 2020)

Language model	token size	parameters	test accuracy
XLM-R-base	595 Million	125M	61.09
IndicBERT	**549 Million**	**12M**	**66.66**
bert-base-multilingual-cased		110M	64.62

Table 4: F1 score of different pre-trained models

Pretrained model	MCC	F1 Score
XLM-R-base	0.58048	0.73052
IndicBERT	**0.59426**	**0.73684**
bert-base-multiling-cased	0.58933	0.73474

Table 5: Different hyperparameter used for evaluation

Hyper parameter	values
Learning Rate	1E-5,2E-5,3E-5,4E-5,5E-5
Batch Size	16, 32
Epochs	2, 3, 4

decided to use this model to improve grammar error detection for future experiments.

5.2 Evaluation

We used two standard metrics, namely MCC (Matthew Correlation Coefficient)(Matthews, 1975) and F1 score, to evaluate the model that we trained. Table 4 shows the initial performance of different fine-tuned classification models for the test set. It is evident from the results that the IndicBERT outperforms other pre-trained models. Moreover, since hyper-parameters also affect the results, we experiment with different hyper-parameter combinations to fine-tune the classification model. Table 5 shows fine-tuned values for the set of hyper-parameters. We change the hyper-parameters to get a better F1 score. Initially, the F1 score was 73%. Also, we found significant confusion among number errors and gender errors. The dataset has number error, gender error, person error and error-less sentences. We also found that some sentences have two kinds of errors when we look deeper. Therefore, we defined error precedence to the prioritise error labels as number > gender > person. For instance, Table 6 shows how number error is prioritised over the gender error in the dataset. After this precedence setting, the grammar error detection showed the F1 score of 84%. Figure 2 shows the current confusion matrix among different type of errors. Eventually, we obtained the best results for the combination of learning rate = 3E-05, batch size = 16, and epochs = 4 along with IndicBERT. Equations 1, 2, and 3 show that how we calculated the F1 score from True Positive (TP), False Positive (FP), and False Negative (FN) values.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{374}{374 + (19{+}9{+}27{+}12)} = 0.84 \quad (1)$$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{374}{374 + (3{+}25{+}10{+}29)} = 0.84 \quad (2)$$

$$F_1 = \frac{\text{Precision} \times \text{Recall}}{\text{Precison} + \text{Recall}} = \frac{2 \times 0.84 \times 0.84}{0.84 + 0.84} = 0.84 \quad (3)$$

Table 6: Precedence of errors

Erroneous sentence	number error	gender error	error type
கவிதா வந்தான் . kavitā vantān Kavitha.NOM-**3SgF** come.**3SgM** .	false	true	gender error
தமிழ் மொழி பழமையானவை Tamil moli palamaiyānavai Tamil language.NOM-**3Sg** old.**3Pl** .	true	true	number error

6 Conclusion

We have implemented a baseline application for Tamil grammatical error detection using the state-of-the-art approach. The application outlined here detects grammatical errors related to the person-number-gender agreement between the nominative subject and the verbal predicate in a sentence. We used a multi-lingual pre-trained model to capture the Tamil structures and then fine-tuned it using the grammar error annotated data we created. We found that the IndicBERT model gives better accuracy than other pre-trained models. Our baseline model shows an F1 Score of 84.0% for unseen a test set.

As the next step, we are planning to use *Thamizhi*Morph (Sarveswaran et al., 2021) — A Morphological analyser to create more annotated data to train the grammar checker. The current model relies on the pre-trained model to capture the syntactic information such as subject and predicate. However, this can be obtained using a syntactic parser, and the syntactically parsed data may increase the score. Therefore, as the next step, we will also experiment with a Tamil dependency parser called *Thamizhi*UDp (Sarveswaran and Dias, 2020) to incorporate syntactic information such as subject and predicate information into our datasets to see whether the proposed system can be improved further.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

S Rajendran. 2012. Preliminaries to the preparation of a spell and grammar checker for Tamil. *Language in India*, 2.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2016. A novel hybrid approach to detect and correct spelling in Tamil text. In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–6. IEEE.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2018. Detecting and correcting real-word errors in Tamil sentences. *Ruhuna Journal of Science*, 9(2).

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2019. Detecting and Correcting Grammatical Mistakes due to Subject-Verb Inconformity and Conflicts in Tense Aspects in Tamil Sentences. In *6th Ruhuna International Science and Technology Conference (RISTCON)*.

Kengatharaiyer Sarveswaran and Miriam Butt. 2019. Computational challenges with Tamil complex predicates. In *Proceedings of the LFG19 conference, Australian National University. CSLI, Stanford*, pages 272–292.

Kengatharaiyer Sarveswaran and Gihan Dias. 2020. ThamizhiUDp: A dependency parser for Tamil. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 200–207, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. ThamizhiMorph: A morphological parser for the Tamil language. *Machine Translation*, 35(1):37–70.

J Segar and K Sarveswaran. 2015. Contextual spell checking for Tamil language. In *14th Tamil Internet conference*, pages 1–5.

Keerthana Uthayamoorthy, Kirshika Kanthasamy, Thavarasa Senthaalan, Kengatharaiyer Sarveswaran, and Gihan Dias. 2019. DDSpell-A Data Driven Spell Checker and Suggestion Generator for the Tamil Language. In *2019 19th international conference on advances in ICT for emerging regions (ICTer)*, volume 250, pages 1–6. IEEE.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498. Association for Computational Linguistics.