Deep Learning Based Sentiment Analysis for Malayalam,Tamil and Kannada Languages

Pavan Kumar P.H.V, Premjith B, Sanjanasri J.P and Soman K.P

Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

Abstract

This paper describes the submission of the Amrita_CEN_NLP team to the shared task on Dravidian-CodeMix-FIRE2021. The dataset used in this task is CodeMix text associated with the context of social media. It's most common to notice the comments under Youtube videos, Facebook posts in the CodeMix. In this task, we implemented three different Deep learning-based architectures: Deep Neural Network (DNN), Bidirectional-Long Short Term Memory network (Bi-LSTM), and finally, Convolution Neural network (CNN) combined with a Long Short Term Memory network (LSTM) for predicting various sentiments associated with the Dravidian CodeMix languages(Malayalam, Tamil, Kannada). The data given by organizers is highly imbalanced to handle this issue weightage given to each class weight based on their distribution over data. Our experiments reveal that CNN combined with LSTM, DNN with one hidden layer performs best for Malayalam linguistics and, the BiLSTM layer suits the classification of Tamil and Kannada corpus. After inferring the results obtained on performed experiments, we submitted the results.

Keywords

CodeMix, Multilingual, Tamil, Malayalam, Kannada, Dravidian

1. Introduction

India is a multilingual country [1] where we often spot conversations on social media platforms [2] like YouTube, Facebook and, Twitter in code-mixed text. *Sentiment analysis* [3] is a concept/technique involved in identifying and analyzing the sentiment/mood of people in the social media [4] context. To classify the underlying sentiments of text as positive, negative, mixed feelings, Native, non-Native, we use sentiment analysis [5].

Text that adopts the vocabulary and grammar from multiple languages frames a new structure based on its usage called code-mixed text [6]. This paper discusses the methodology and results submitted to the shared task of sentiment analysis for Malayalam-English, Tamil-English, and Kannada-English languages [7]. We implemented three Deep Neural network architectures for classifying code-mixed text: Convolution Neural Network (CNN) combined with LSTM

FIRE 2021: Forum for Information Retrieval Evaluation, December 17-21, 2020, Hyderabad, India

cb.en.p2cen20020@cb.students.amrita.edu (P. K. P.H.V); b_premjith@cb.amrita.edu (. P. B); jp_sanjanasri@cb.amrita.edu (S. J.P); kp_soman@amrita.edu (S. K.P)

https://www.linkedin.com/in/pavan-kumar-phv/ (P. K. P.H.V); https://www.amrita.edu/faculty/b-premjith (. P. B); https://www.researchgate.net/profile/Sanjanasri-Jp (S. J.P); https://www.amrita.edu/faculty/kp-soman (S. K.P)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3159/T6-17.pdf

(CNN-LSTM) [8], Bidirectional-Long Short Term Memory (Bi-LSTM) network [9], and Deep Neural Network(DNN) with one hidden layer.

The remaining sections of the paper consist of, Section:2 details the work done in this area, Section:3 explains the dataset used in the shared task, Section:4 discusses the methodology followed in conducting experiments, Section:5 details the list of experiments and results. Finally, the paper concludes with Section:6.

2. Literature Review

B. R Chakravarthi et al. [10] created a golden standard corpus for the code-mixed dataset in Malayalam–English language. The authors collected data from YouTube comments after preprocessing, manually labeled the data with the help of annotators. B.R. Chakravarthi et al. used Logistic regression (LR), Support vector machine (SVM), Decision tree (DT), Random Forest (RF), Multinomial Naive Bayes (MNB), K-Nearest Neighbours (KNN) as machine learning techniques and, Dynamic Meta-Embeddings (DME), Contextualized DME(CDME), One Dimensional Convolution Neural Network(1D-CNN), Bidirectional Encoder Representations for Transformers (BERT) as Deep Learning techniques for defining a baseline method for sentiment analysis. Except for SVM rest, all the machine learning Models had detected the various classes in the data. Due to the usage of pre-trained embeddings in deep learning Models, CDME and DME are thriving to identify all the classes and, 1D-CNN shows better F1-score, precision, recall, and macro-average.

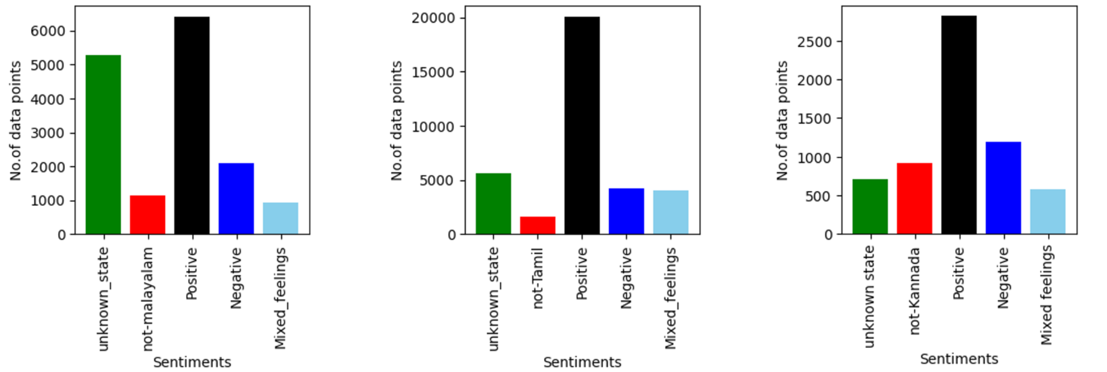
In 2020, Soumya S & Pramod K.V conducted sentiment analysis on unilingual Malayalam tweets [11] using various machine learning techniques combined with different features embeddings for tweets of positive and negative classes. They used SVM, NB, and Random Forest (RF) machine learning techniques for classification of tweets and found that RF gives significant accuracy along unigram with Sentiwordnet by considering negation word as a feature.

Manju Venugopalan & Deepa Gupta performed sentiment analysis [12] on the binary classification of Twitter data using SVM and Decision Tree (J48) classifiers. The authors measured the performance of the SVM and J48 Model by comparing them with the unigram Model performance and, they found that J48 and SVM classifier outperformed when compared with the unigram Model.

T. Tulasi Sasidhar et al. [13] had used deep learning techniques to perform sentiment analysis on Hindi-English code-mix data. They perceived that the CNN-Bi-LSTM Model had achieved the best performance compared to other Models with an F1-score of 70.32%. A similar Model with some slight variations is used in this shared task, where the details of the Model are explained in the section 4.2.

3. Dataset Description

The dataset used in the shared task [14] contains bilingual and native texts of three different languages, Malayalam-English [10], Tamil-English [15] and, Kannada-English [16]. Figure 1 illustrates the distribution of data over classes, and the split of the dataset in conducting the experiments are mentioned in Table 1.



(a) Malayalam-English Train Dataset **(b)** Tamil-English Train Dataset **(c)** Kanada-English Train Dataset

Figure 1: Distribution of train dataset over each language.

Table 1
Description of class labels and their train, validation, and test split of the corresponding languages.

Language	Class	Train Dataset	Validation Dataset	Test Dataset
Malayalam-English	unknown_state Positive Negative Mixed_feelings not-malayalam	15888	1766	1962
Tamil-English	unknown_state Positive Negative Mixed_feelings not-Tamil	35656	3962	4402
Kanada-English	unknown state Positive Negative Mixed feelings not-Kannada	6212	691	768

4. Methodology

This section explains the methodology followed in conducting experiments and the Models submitted to the shared task.

4.1. Preprocessing

Dataset [14] used in the shared task is a mix of the Dravidian(Malayalam, Tamil & Kannada) and English language of social media corpus [17], which contains lots of special Characters, emojis, URLs, and hashtags. These entities affect the performance of the Model accuracy. To remove all such entities from the corpus [18], we implemented the preprocessing stage.

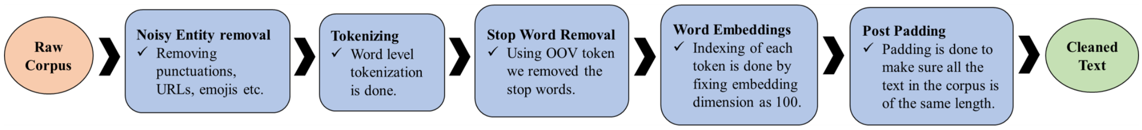


Figure 2: Stages in preprocessing

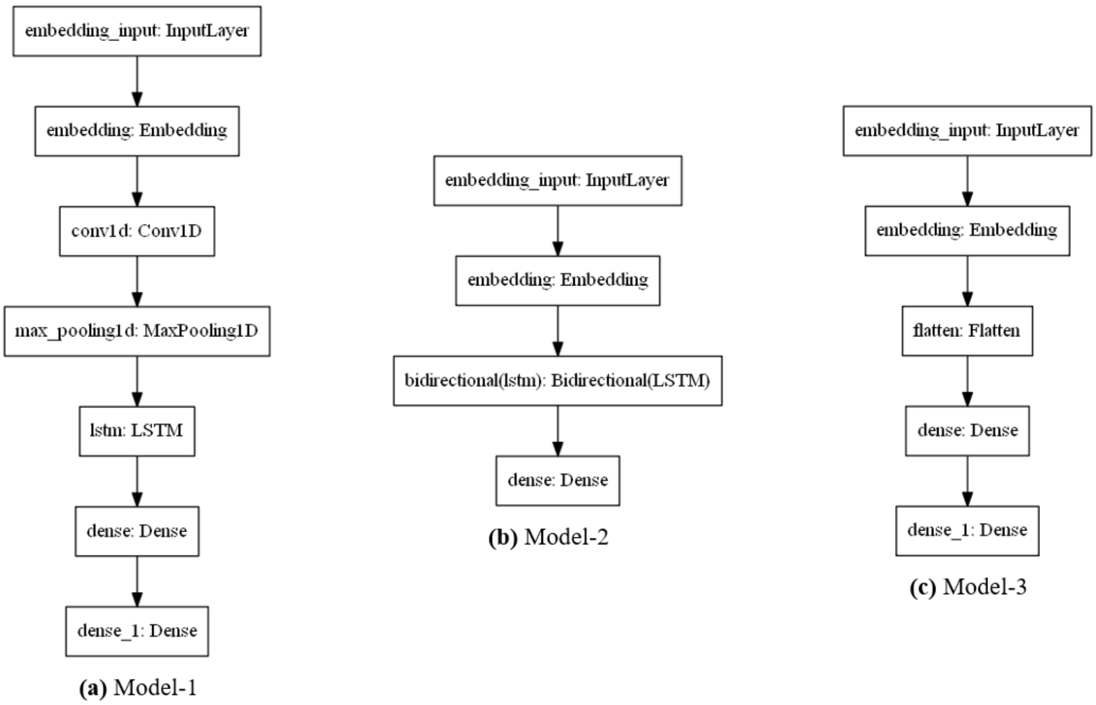


Figure 3: Illustration of all three Models used in conducting experiments

4.2. Description on Models

Experiments had conducted on the dataset using various Models of deep neural network architectures. Model-1 illustrated in Figure 3 contains embedding layer, 1D-CNN, 1D Max Pooling, Long Short Term Memory (LSTM), a hidden layer and finally, a dense layer. Model-2 contains an embedding layer, a Bidirectional-Long Short Term Memory network (Bi-LSTM) and, a dense layer. Model-3 contains an Embedding layer, a Flatten, a hidden and, a Dense layer.

Each Model illustrated in Figure 3 follows a set of sequential steps before feeding into the network. After preprocessing data, the extracted features as embedded vectors for each sentence in the corpus are feed forwarded as inputs to the network.

Dataset used in the shared task is highly imbalanced. The concept of class weights [19] is applied to overcome this issue by computing the Individual class weights using equation 1. Classes labels with more data points get minimum weight, and with fewer gets maximum weight

$$C_w = \frac{\sum_{c=1}^{n} N_c}{N_c} \tag{1}$$

In the above equation-(1),

$C_w \rightarrow$ Class Weights, $\sum_{c=1}^{n} N_c \rightarrow$ Sum of all the sentences in the corpus $N_c \rightarrow$ Number of sentences in each class c.

4.3. Hyperparameter tuning

Table 2
Hyperparameter values and the optimal values used in Model-2&3

	Hyperparameter	Values	Optimal Value
Model-2	Embedding dimension	50, 100	100
	embeddings_initializer	uniform, orthogonal, constant	orthogonal
	embeddings_regularizer	L1, L2	L1
	Number of neurons in LSTM layer	16, 32, 64, 128, 256	32
	Activation Function at hidden layer	Sigmoid, RELU	RELU
	Activation Function at Output layer	Softmax	Softmax
	Optimizer	Adam	Adam
	Loss function	Sparse Categorical Crossentropy, Categorical Crossentropy	Categorical Crossentropy
	learning Rate	0.1, 0.01, 0.001	0.01
	Batch size	16, 32, 64, 80, 128, 132, 256	128
Model-3	Embedding dimension	50, 100	100
	Number of neurons in hidden layer	16, 32, 64, 128, 256	128
	Activation Function at hidden layer	Sigmoid, RELU	RELU
	Activation Function at Output layer	Softmax	Softmax
	Optimizer	Adam	Adam
	Loss function	Sparse Categorical Crossentropy, Categorical Crossentropy	Categorical Crossentropy
	learning Rate	0.1, 0.01, 0.001	0.01
	Batch size	16, 32, 64, 80, 128, 132, 256	64

Hyperparameter tuning was conducted based on improvements in Accuracy, Precision, Recall and, AUC values. Table 2 shows the hyperparameter values and the optimal values used for conducting experiments on Model-3, Which was the best performing Model.

5. Experiments and Results

We used three different deep neural network Models illustrated in Figure 3 to conduct the shared task experiments[1]. Model-1 contains a 1D-CNN, Max Pooling, LSTM layer, and a fully connected dense layer; Model-2 had one Bi-LSTM layer followed by a dense layer; Model-3 had a hidden layer and one fully connected dense layer. The experimental results on the training dataset of all three Models on the selected hyperparameters are in Table 3,4,5, and the validation performance is in Table 6. The best-performing Model metrics values are highlighted in bold font.

DNN with one Hidden layer achieve better classification than Model-1 and Model-2 on the Malayalam-English language. BiLSTM with the mentioned hyperparameters in Tabel 2 performs better than Model-1 and Model-3 on the Kannada-English CodMix. For the Tamil-English corpus based on training and testing performance and the metric values, we go for Model-2.

Table 3
Training performance on Malayalam-English Dataset for various Models

Model	Accuracy	Precission	Recall	AUC
Model-1	0.925	0.8297	0.7866	0.9633
Model-2	**0.9482**	**0.8881**	**0.848**	**0.9806**
Model-3	0.8428	0.8571	0.2545	0.7657

Table 4
Training performance on Tamil-English Dataset for various Models

Model	Accuracy	Precission	Recall	AUC
Model-1	0.9732	0.9473	0.9171	0.9919
Model-2	0.8439	0.7037	0.3778	0.8389
Model-3	**0.9905**	**0.9787**	**0.9737**	**0.9972**

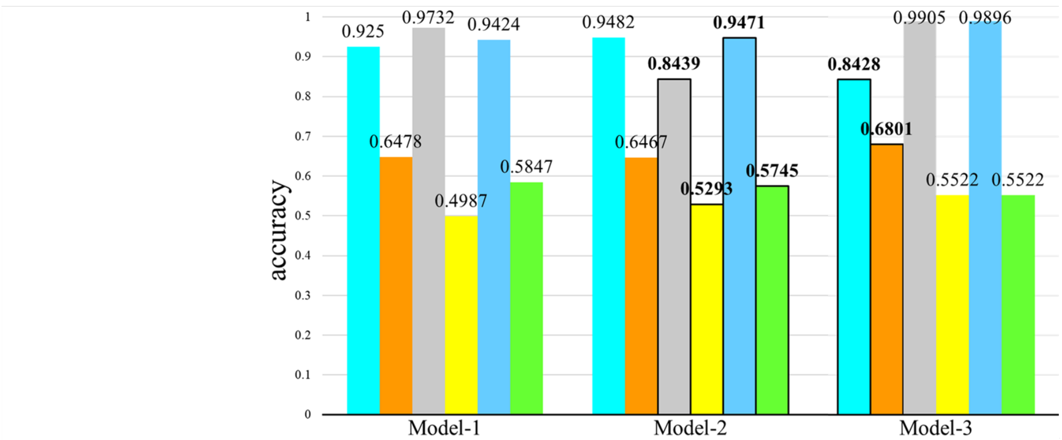
Table 5
Training performance on Kannada-English dataset for various Models

Model	Accuracy	Precission	Recall	AUC
Model-1	0.9424	0.8741	0.8316	0.9769
Model-2	0.9471	0.8823	0.8489	0.9811
Model-3	**0.9896**	**0.9762**	**0.9719**	**0.9992**

[1]https://github.com/phvpavankumar/Sentiment-Analysis-for-Malayalam-Tamil-and-Kannada-Languages

Table 6
Testing Performance of all the three Models

Language	Malayalam - English			Tamil - English			Kannada - English		
Model	**Precission**	**Recall**	**F1 Score**	**Precission**	**Recall**	**F1 Score**	**Precission**	**Recall**	**F1 Score**
Model-1	0.5854	0.6432	0.6077	0.4397	0.5072	0.4384	0.5007	0.5248	0.5085
Model-2	0.5797	0.6346	0.5995	**0.4232**	**0.5072**	**0.441**	**0.5062**	**0.5455**	**0.5193**
Model-3	**0.6303**	**0.6304**	**0.627**	0.43	0.4631	0.4408	0.4855	0.5126	0.4552



	Model-1	Model-2	Model-3
Malyalam-English Train accuracy	0.925	0.9482	**0.8428**
Malyalam-English Test accuracy	0.6478	0.6467	**0.6801**
Tamil-English Train accuracy	0.9732	**0.8439**	0.9905
Tamil- EnglishTest accuracy	0.4987	**0.5293**	0.5522
Kannada-English Train accuracy	0.9424	**0.9471**	0.9896
Kannada-English Test accuracy	0.5847	**0.5745**	0.5522

Figure 4: Testing Performance of all the three Models

6. Conclusion

In this paper, we discussed the submission of a shared task by team Amrita_CEN_NLP for Dravidian-CodeMix-FIRE2021. We did sentiment analysis for three Dravidian code-mixed languages, Malayalam, Tamil and, Kannada. We used three different deep learning Models: Model-1 had a 1D-CNN layer, Maxpooling layer, LSTM, a fully connected dense layer. Model-2 had one Bi-LSTM layer, Model-3 had only one fully connected thick layer for conducting experiments. After training three embedding Models on datasets several times, optimal hyperparameters we listed and the results obtained from Model-3 were much better when compared with Model-1 and Model-2 in Malayalam-English linguistics. Model-2 suits good for Kannada-English and Tamil-English linguistics.

References

[1] B. Krishnamurti, Dravidian languages (2020). URL: https://www.britannica.com/topic/Dravidian-languages.

[2] S. Suryawanshi, B. R. Chakravarthi, Findings of the shared task on troll meme classification in Tamil, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 126–132. URL: https://aclanthology.org/2021.dravidianlangtech-1.16.

[3] N. Choudhary, R. Singh, I. Bindlish, M. Shrivastava, Sentiment analysis of code-mixed languages leveraging resource rich languages, CoRR abs/1804.00806 (2018). URL: http://arxiv.org/abs/1804.00806. arXiv:1804.00806.

[4] S. Banerjee, B. Raja Chakravarthi, J. P. McCrae, Comparison of pretrained embeddings to identify hate speech in indian code-mixed text, in: 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 21–25. doi:10.1109/ICACCCN51052.2020.9362731.

[5] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on sentiment analysis for dravidian languages in code-mixed text, in: Forum for Information Retrieval Evaluation, 2020, pp. 21–24.

[6] B. R. Chakravarthi, P. K. Kumaresan, R. Sakuntharaj, A. K. Madasamy, S. Thavareesan, P. B, S. Chinnaudayar Navaneethakrishnan, J. P. McCrae, T. Mandl, Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[7] S. Banerjee, A. Jayapal, S. Thavareesan, Nuig-shubhanker@dravidian-codemix- fire2020: Sentiment analysis of code-mixed dravidian text using xlnet, in: FIRE, 2020.

[8] K. Sreelakshmi, B. Premjith, S. Kp, Amrita_cen_nlp@ dravidianlangtech-eacl2021: Deep learning-based offensive language identification in malayalam, tamil and kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 249–254.

[9] B. Premjith, K. Soman, Deep learning approach for the morphological synthesis in malayalam and tamil at the character level, Transactions on Asian and Low-Resource Language Information Processing 20 (2021) 1–17.

[10] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.

[11] S. S., P. K.V., Sentiment analysis of malayalam tweets using machine learning techniques, ICT Express 6 (2020) 300–305. URL: https://www.sciencedirect.com/science/article/pii/S2405959520300382. doi:https://doi.org/10.1016/j.icte.2020.04.003.

[12] M. Venugopalan, D. Gupta, Exploring sentiment analysis on twitter data, in: 2015 Eighth International Conference on Contemporary Computing (IC3), 2015, pp. 241–247. doi:10.1109/IC3.2015.7346686.

[13] T. T. Sasidhar, B. Premjith, K. Sreelakshmi, K. P. Soman, Sentiment analysis on hindi–english code-mixed social media text, 2021. doi:10.1007/978-981-33-4543-0_65.
[14] R. Priyadharshini, B. R. Chakravarthi, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, Overview of the dravidiancodemix 2021 shared task on sentiment detection in tamil, malayalam, and kannada, in: Forum for Information Retrieval Evaluation, FIRE 2021, Association for Computing Machinery, 2021.
[15] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.
[16] A. Hande, R. Priyadharshini, B. R. Chakravarthi, KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection, in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 54–63. URL: https://www.aclweb.org/anthology/2020.peoples-1.6.
[17] B. R. Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, J. P. McCrae, A. Hande, R. Ponnusamy, S. Banerjee, C. Vasantharajan, Findings of the Sentiment Analysis of Dravidian Languages in Code-Mixed Text 2021, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[18] S. Ayvaz, M. Shiha, The effects of emoji in sentiment analysis, International Journal of Computer and Electrical Engineering 9 (2017) 360–369. doi:10.17706/IJCEE.2017.9.1.360-369.
[19] B. Premjith, K. Soman, Amrita_cen_nlp@ wosp 3c citation context classification task, in: Proceedings of the 8th International Workshop on Mining Scientific Publications, 2020, pp. 71–74.