Multi-Label Emotion Classification in Urdu

Dejah Madhusankar, Avanthika Karthikeyan and Bharathi B

Department of CSE, Sri Siva Subramaniya Nadar College of Engineering, Tamil Nadu, India

Abstract

With the massive growth and widespread usage of social media platforms, the rates of its misuse and its corresponding impact on society have seen an exponential rise in numbers. The comfort of anonymity and wide reach offered by social media has led to the convenient spread of hatred and incitement to threats, that are often targeted against particular users and communities. Thus identifying hate speech, threats and intense emotions in the digital arena has gained attention recently. This is also the aim of the EmoThreat: Emotions and Threat Detection in Urdu 2022 Challenge. In this paper, we describe a few traditional machine learning models and deep neural networks submitted by our team Aces for Task A: Multi-label emotion classification in Urdu. The models tested include Classifier Chains, MLKNN, RNN and LSTM Networks implemented with a combination of feature extraction methods such as the Count Vectorizer and TF-IDF, as well as embedding models like Word2Vec and FastText. Each model has been discussed in detail in Section 4 after a brief overview of the dataset adopted, in Section 3. Out of these tested permutations, the Classifier Chains model with TF-IDF vectorization proved to give the most promising results, which has been detailed in Section 5.

Keywords

Emotion Classification in Urdu, Classifier Chains, RNN, TF-IDF, Neural Networks, Multi-label emotion detection, fastText, MLKNN

1. Introduction

The right to free speech and expression on global platforms has inadvertently led to the generation of numerous digital posts containing hateful, sensitive and abusive content. These vulgar narratives, often-times targeted against certain individuals and communities worsen users' experience from communication via such media, while other posts contain actual threats that put users in danger. Today major social-media platforms such as Google, Meta, YouTube and Twitter are taking significant efforts to resolve this issue by detecting and censoring objectionable content before it can instigate disruptions and chaos in society. With the Urdu language having more than 230 million speakers worldwide[1], a massive amount of user data gets generated on an everyday basis. Content moderation of such enormous data is difficult to achieve solely through manpower. This paper thereby explores various Machine Learning (ML) algorithms and models for generating multi-label classification of emotions in Urdu text,

Forum for Information Retrieval Evaluation, December 16-20, 2022, India

dejahmadhushankar@gmail.com (D. Madhusankar); karthiavanthika@gmail.com (A. Karthikeyan); bharathib@ssn.edu.in (B. B)

https://www.linkedin.com/in/dejah-madhusankar/ (D. Madhusankar); https://github.com/Avanthika-K/Multi-Label-Emotion-Classification-in-Urdu (A. Karthikeyan); https://www.ssn.edu.in/staff-members/dr-b-bharathi/ (B. B)

0000-0002-0877-7063 (D. Madhusankar); 0000-0001-7116-9338 (A. Karthikeyan); 0000-0001-7279-5357 (B. B)

in a shared task called EmoThreat hosted by the Forum for Information Retrieval Evaluation, 2022, at the Indian Statistical Institute, Kolkata, overview of which is cited here [2],[3]. Our team submitted five runs for Task A - Multi-label Classification of Emotions in Urdu and has been ranked 8 in the leaderboard.

2. Related Work

With social media as a major platform for blogging and sharing information impacting millions of users' lives every day, detecting and identifying intense emotions and objectionable content becomes a very important task. A lot of research has been conducted over the years on various aspects of the said issue keeping in mind the significance it holds on today's digital society. Over the years, individuals and organizations have come together and proposed various different models.

For instance, in [4] the researchers have adopted traditional, deep learning and transformer-based Machine Learning (ML) approaches along with different kinds of text representation for emotion classification, while the authors of [5] and [6] propose traditional machine learning techniques to categorize text as abusive and not abusive. Similar work was carried out by the authors of [7] where the data set containing tweets in the English language was labeled and categorized as hate speech, offensive language or neither, leading to the authors concluding their work as stated - "We find that racist and homophobic tweets are more likely to be classified as hate speech but that sexist tweets are generally classified as offensive. Tweets without explicit hate keywords are also more difficult to classify." Following this, the authors of [8] have implemented content-based methods for Multi-label classification of tweets, using various classifiers such as Binary Relevance, Classifier Chain, and Label Combination. This work had been undertaken with the literary text [9] providing a backbone to the significance of binary-relevance-based methods for multi-label classification, where the authors have argued that the binary-relevance method has significantly much to offer especially in terms of scalability to large data sets.

On a much recent scale, the researchers of [10] have explored different text representations, namely count-based and fastText pre-trained word embeddings for Urdu, and have considered various machine and deep learning algorithms for evaluation. The latest work of the authors of [11] proficiently fine-tunes the Multilingual BERT(mBERT) model for Urdu sentiment analysis that uses four different text representations to train classifiers. Amidst the general concerns and obstacles faced in developing accurate ML classifiers, it has been majorly realized that the technicalities of the language involved plays a crucial role in the processing of text. Researchers from Korea have put forth an analysis of the difficulties that come along with identifying Korean swear words accurately. Their study, in their words "proposes a method of discriminating profanity using a deep learning model that can grasp the meaning and context of words after separating Hangul into the onset, nucleus, and coda."

Table 1
Categorical data split for train data set

Label	Anger	Disgust	Fear	Sadness	Suprise	Happiness	Neutral
Count	811	761	609	2190	1550	1046	3014

3. Datasets

The dataset used is adapted from Task A of EmoThreat: Emotions and Threat Detection in Urdu, FIRE 2022. The Urdu text in question has been generated using the Nastalīq Urdu script dataset and consists of tweets made by various users on Twitter. The labels adopted for the text classification are based on Ekman's six basic emotions and neutrality. These six basic emotions apart from Neutral are: Anger, Disgust, Fear, Sadness, Surprise and Happiness. The data set consists of 8 columns, representing the 7 labels of emotion followed by the Urdu tweet in the last column. There is a total of 7,800 rows, each of which corresponds to an independent tweet and its corresponding classification label - marked as '1' in the appropriate column if the emotion is detected, or '0' otherwise. Table 1 shows the distribution of the data set.

4. Implementation and Experiments

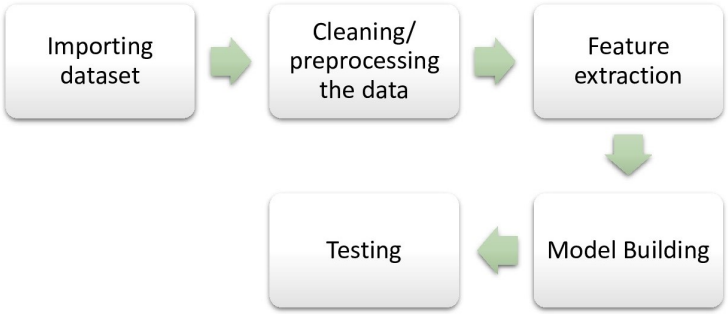

Figure 1: Flowchart describing the workflow

4.1. Models

4.1.1. Classifier Chains

Classifier chains is a machine learning model used for problem transformation in multi-label classification scenarios. It uses |L| (number of labels) binary classifications where each classification is linked along a chain and deals with the binary relevance problem associated with its corresponding label L_j ∈ L[12] Classification starts at the beginning and it propagates along the chain, wherein each classifier learns and predicts the binary association of that particular label.

4.1.2. MLkNN

The Multi-Label k-Nearest Neighbors (MLkNN) algorithm works similarly to the K-Nearest Neighbors (KNN) method, which is a traditional non-parametric supervised learning approach. MLKNN builds on the kNN approach to find the nearest examples to a test class and proceeds to use Bayesian inference to select appropriate labels for classification. It starts by identifying the k-nearest neighbors, followed by the identification of the corresponding labels for the instance using the information collected from those neighbors. k-values in the range of 10 to 40 were passed and better results were seen when the value approached 20. In our model, we have used the default smoothing parameters, ignoring the first neighbors which had values of 1.0 and 0 respectively.

4.1.3. Simple RNN

RNN stands for Recurrent Neural Networks, which is a class of neural networks with loops in them, allowing information to persist for a period of time. This neural architecture thus allows inputs to be taken from previous outputs, enabling patterns to be dependent on previously extracted patterns instead of treating each data point as an individual entity. The RNN model employed we have built here uses a sigmoid activation layer along with a binary cross-entropy loss function and Adam optimizer.

4.1.4. LSTM Network

Long Short-Term Memory (LSTM) is a special recurrent neural network (RNN) architecture that was built to hold an edge over standard feed-forward neural networks and RNN because of their trait of selectively remembering patterns for a longer duration of time. LSTMs use feedback connections which distinguishes them from traditional feed forward neural networks. This property enables LSTM models to process sequences of data without treating each sequence point as an independent entity, but rather, retaining useful metadata about the previous sequence points to help with the computation and processing of new data points. We used LSTM-based neural network classifiers built with the Keras toolkit, along with fastText Urdu word embedding to classify our Urdu text. We used word tokens, Embedding layer (300 dimensions), input length (10001 words) as inputs to the LSTM (128 units) layer and *sigmoid* activation function for the output layer. In this pipeline, we used categorical-cross entropy as the loss function and the Adam optimizer for parameter optimization.

4.2. Feature Extraction

4.2.1. CountVectorizer

CountVectorizer is a tool provided by the scikit-learn library in Python, used to convert text into a vector based on the frequency (count) of each word that occurs in the entire text. This is used as a feature extraction method for text classification problems. Making use of frequencies, it converts a group of text documents to a matrix of token counts. Here, the MLKNN model uses CountVectorizer and takes max_features as 1000 with max_df value of 0.85.

4.2.2. TF-IDF

TF-IDF, short for Term Frequency-Inverse Document Frequency is also a scikit-learn library feature provided by Python for feature extraction in text processing. It converts a collection of raw documents to a matrix of TF-IDF features. Term frequency(TF) refers to the frequency of a word that occurs in a document while Inverse Document Frequency(IDF) implies the number of times the document contains the word in the corpus. TF-IDF is nothing, but the multiplication of TF and IDF. It works similarly to CountVectorizer but gives more importance to the words and their relevance. This enables us to remove the words that are less important for our analysis, thereby simplifying the model building by reducing the input dimensions.

4.2.3. Word2Vec

Word2Vec is a technique to efficiently create word embedding which makes use of shallow neural networks. The effectiveness of Word2Vec is due to its ability to gather and group together vectors of similar words. The embedding matrix was generated using the Word2Vec model, having a model vocabulary's size of 85868. The embedding layer was done using Word2Vec embedding matrix and used a vocabulary size of 5000.

4.2.4. FastText

FastText is an open-source, free library used for efficient learning of word embedding and text classifications. It is a very fast NLP library created by Facebook's AI Research (FAIR) lab. It allows the training of both supervised and unsupervised representations of sentences. The implemented model utilizes Urdu word embeddings which are represented as vectors having a dimension of 300.

5. Results

The performance parameters of the various models tested are presented in the tables below. Table 2 presents the model performance results shown while training and Table 3 displays the results shown by test data. Considering the training data set, a performance accuracy of around 60% was achieved using the Classifier chains model and the simple RNN model. Overall, in the test data set, the Classifier Chains model gave the best results, with a performance accuracy of around 43%. This can possibly be attributed to its property of combining the computational efficiency of the widely-known Binary Relevance method for multi-label classification, while still being able to take the label dependencies into account for classification. It also retains the advantages of the binary method including low memory and run-time complexity. Moreover, in the recent decade, studies exploring the underlying theory and working behind classifier chains have made many improvements to the training procedures, such that this method remains among the state-of-the-art options for multi-label learning of text data. [13]

The least performing model here is MLKNN, though there has been a visible improvement in metrics when different feature extraction algorithms and parameter tuning were applied. The reasons behind its underperformance appear to be inconclusive. Yet we believe that the

Table 2
Cross validation scores of the proposed system

Model	Feature Extraction	Accuracy	Weighted F1
Classifier Chains	TF-IDF	0.59	0.62
Simple RNN	Word2Vec	0.597	0.56
MLKNN	Count Vectorizer	0.55	0.60
MLKNN	TF-IDF	0.57	0.66
Lstm	fastText	0.40	0.22

Table 3
Performance of the proposed system using test data

Model	Feature Extraction	Accuracy	Macro F1
Classifier Chains	TF-IDF	0.426	0.381
Simple RNN	Word2Vec	0.052	0.114
MLKNN	Count Vectorizer	0.354	0.371
MLKNN	TF-IDF	0.052	0.114
Lstm	fastText	0.189	0.24

pre-processing of text and the size of training data used always play a much bigger role than expected. MLKNN is moreover known to be computationally expensive because the algorithm stores all of the training data and is sensitive to irrelevant features. This could potentially lead to errors during classification, which hypothetically explains its high performance with the training data and poor performance with the test data.

6. Conclusions

In this paper, we have presented solutions to Task A of EmoThreat: Emotions and Threat Detection in Urdu, at FIRE 2022. In this work, 4 different ML models along with 4 feature extraction methods were adopted and tested for identifying the seven categories of emotions (six basic emotions along with, *neutral*), and thereby classify various kinds of texts or tweets. Prediction accuracy, F1 macro and micro scores were used as evaluation metrics with the F1 score as the key parameter. Although the results presented are adequately good, we believe that they could potentially be improved with better training and cross-validation methods. It can also be enhanced further by exploring the linguistic features of the Urdu language for better feature extraction. The possibility of other ML models outperforming the proposed systems remains open, and a higher performance accuracy can be fore-looked by employing fine-tuned parameters. For future works, transformer-based models and other deep learning approaches can be applied with fine-tuned parameters to improve emotion detection and classification.

Acknowledgments

Thanks to the Department of CSE, SSN College of Engineering, Kalavakkam, India, for providing us with the opportunity, awareness, and guidance for this task. Special thanks to the organizers of FIRE 2022 for providing us with the necessary data set and for bringing together individuals nationwide to work on this project.

References

[1] Ethnologue, What are the top 200 most spoken languages?, Available at https://www.ethnologue.com/guides/ethnologue200 (2018/03/03), ????

[2] S. Butt, M. Amjad, F. Balouchzahi, N. Ashraf, R. Sharma, G. Sidorov, A. Gelbukh, Overview of EmoThreat: Emotions and Threat Detection in Urdu at FIRE 2022, in: CEUR Workshop Proceedings, 2022.

[3] S. Butt, M. Amjad, F. Balouchzahi, N. Ashraf, R. Sharma, G. Sidorov, A. Gelbukh, EmoThreat@FIRE2022: Shared Track on Emotions and Threat Detection in Urdu, in: Forum for Information Retrieval Evaluation, FIRE 2022, Association for Computing Machinery, New York, NY, USA, 2022.

[4] N. Ashraf, L. Khan, S. Butt, H.-T. Chang, G. Sidorov, A. Gelbukh, Multi-label emotion classification of urdu tweets, PeerJ Computer Science 8 (2022) e896.

[5] B. B. J. B. T. M. K A. Karthikraja, Aarthi Suresh Kumar, Abusive and threatening language detection in native urdu script tweets exploring four conventional machine learning techniques and mlp806-812, FIRE 2021 Working Notes (2021).

[6] M. Amjad, N. Ashraf, A. Zhila, G. Sidorov, A. Zubiaga, A. Gelbukh, Threatening language detection and target identification in urdu tweets, IEEE Access 9 (2021) 128302–128313.

[7] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the international AAAI conference on web and social media, volume 11, 2017, pp. 512–515.

[8] I. Ameer, N. Ashraf, G. Sidorov, H. Gómez Adorno, Multi-label emotion classification using content-based features in twitter, Computación y Sistemas 24 (2020) 1159–1164.

[9] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2009, pp. 254–269.

[10] L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, A. Gelbukh, Urdu sentiment analysis with deep learning methods, IEEE Access 9 (2021) 97803–97812.

[11] L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, Multi-class sentiment analysis of urdu text using multilingual bert, Scientific Reports 12 (2022) 1–17.

[12] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Machine learning 85 (2011) 333–359.

[13] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains: a review and perspectives, Journal of Artificial Intelligence Research 70 (2021) 683–718.