

Context Aware Emotion Detection from Low Resource Urdu Language using Deep Neural Network

MUHAMMAD FARRUKH BASHIR, Riphah International University, Pakistan

ABDUL REHMAN JAVED, Air University, Pakistan

MUHAMMAD UMAIR ARSHAD, National University of Computer and Emerging Sciences, Pakistan

THIPPA REDDY GADEKALLU, Vellore Institute of Technology, India

WASEEM SHAHZAD, National University of Computer and Emerging Sciences, Pakistan

MIRZA OMER BEG, National University of Computer and Emerging Sciences, Pakistan, Pakistan

Emotion detection (ED) plays a vital role in determining individual interest in any field. Humans use gestures, facial expressions, voice pitch, and choose words to describe their emotions. Significant work has been done to detect emotions from the textual data in English, French, Chinese, and other high-resource languages. However, emotion classification has not been well studied in low-resource languages (i.e., Urdu) due to the lack of labeled corpora. This paper presents a publicly available Urdu Nastalique Emotions Dataset (*UNED*) of sentences and paragraphs annotated with different emotions and proposes a deep learning (DL) based technique for classifying emotions in the *UNED* corpus. Our annotated *UNED* corpus has six emotions for both paragraphs and sentences. We perform extensive experimentation to evaluate the quality of the corpus and further classify it using machine learning and DL approaches. Experimental results show that the developed DL-based model performs better than generic machine learning approaches with an F1 score of 85% on *UNED* sentence-based corpus and 50% on *UNED* paragraph-based corpus.

CCS Concepts: • **Computing methodologies** → **Language resources**.

Additional Key Words and Phrases: Emotion Detection (ED), Urdu Nastalique Emotions Dataset (UNED), Annotated UNED corpus

1 INTRODUCTION

Emotions are real feelings correlated with circumstances, mood, and relationships with others. Emotion detection (ED) has recently gained increased attention from researchers in Natural Language Processing (NLP) because of numerous business, political, and psychology applications. Humans use gestures [10], facial expressions [54], vocal pitch [16, 27, 33] and words [9, 12, 63] to communicate their emotions.

ED in high-resource languages like English has been extensively studied, but not in low-resource languages like Urdu. Moreover, 300 million people globally speak Urdu, with over 100 million speaking it as their mother

Authors' addresses: Muhammad Farrukh Bashir, farrukh.bashir@riphah.edu.pk, Faculty of Computing, Riphah International University, Islamabad, Pakistan; Abdul Rehman Javed, abdulrehman.cs@au.edu.pk, Department of Cyber Security, Air University, Islamabad, Pakistan; Muhammad Umair Arshad, umair.arshad@nu.edu.pk, Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan; Thippa Reddy Gadekallu, thippareddy.g@vit.ac.in, School of Information Technology and Engineering, Vellore Institute of Technology, Tamil Nadu, India; Waseem Shahzad, waseem.shahzad@nu.edu.pk, Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan; Mirza Omer Beg, omer.beg@nu.edu.pk, Department of Computer Science, National University of Computer and Emerging Sciences, Pakistan, Islamabad, Pakistan.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

2375-4699/2022/1-ART1 \$15.00

<https://doi.org/10.1145/3460392>

tongue. Pakistan's official language, Urdu, is an Indo-Aryan language. Furthermore, Urdu is often used in India and Bangladesh. These people use Urdu to express their emotions on numerous social media platforms [1, 19, 44, 52]. Due to Urdu's complicated morphology, ED from Urdu text is more complex than in English. Urdu has been influenced by several languages, including Arabic, Turkish, English, Persian, and Sanskrit. The complexity of the language can be linked to the terms that have been borrowed from other languages and are present in its vocabulary [55]. The morphological behavior of Urdu is similar to Hindi. Adopted words in Urdu follow the grammatical rules of their parent language, and the structure of a sentence is influenced. In Urdu, verbs and nouns have more than 40 forms [45], thus it is not a trivial task to analyze the Urdu language for further exploration.

The wide range of uses of ED has made it intriguing. In real life, Urdu ED can be used to correlate employee performance with customer reviews based on emotional sentiment. ED can be used to build sentiment-based search engines [18], realistic chat-bots, emotional expression analysis [43], emotional topic discovery and emotional question answering [27], determining the gender of writer [4], automatic emotion word extraction [36], and understanding social interactions [20, 23, 31].

When it comes to purchasing a product, emotion takes precedence over price and functionality. ED can be used to promote things intelligently to enhance sales and target people based on their emotional responses to similar products [5]. ED for Urdu can also be embedded in browser extensions to assist in automatic emotional text predictions and analysis. It can be used for suggestive emoji insertion while composing emails. ED can be further used to categorize books, novels, stories, websites, and blogs to recommend reading material according to the user's emotional state of mind [48]. An example of the Urdu, its Roman Urdu, and English Translation is shown in Figure 1.

★
Example 1:
Urdu Text: میں تم سے بہت ناراض ہوں
Roman Script: Main tum say bohat naraz hon.
Translation: I'm so angry with you.
★
Example 2:
Urdu Text: آج میں بہت خوش ہوں
Roman Script: Aaj main bohat khush hon.
Translation: Today I am very happy.
★

Fig. 1. Comment in Urdu with its Roman Urdu and English translation

Detecting emotions in textual dialogues is difficult due to the absence of facial expressions and voice modulations [7]. In the past, emotion corpora have been built for several languages such as English, Japanese, Chinese, Arabic, and Spanish [27]. Still, for Urdu, limited work has been done to recognize the sentiments of Urdu sentences. Sentiment Analysis is used for finding the polarity (positive/negative) of sentences, but ED is a specialized sentiment analysis task. Sometimes, emotion and sentiment are used as synonyms. However, according to the definition, Emotion is defined as a strong feeling, for example, love, anger, fear, happiness, or sadness, but sentiment defines the general polarity such as positive, negative, or neutral [40]. Many lexicons and sentence-based annotated corpora are available for Urdu sentiment analysis. However, there is no annotated emotion corpus

for the Urdu language. Due to this unavailability, ED has not been explored for the Urdu language, to the best of our knowledge. There is a need for a benchmark corpus to build, evaluate, and compare ED systems.

Several emotions can be expressed using text, but psychologists have focused on basic emotion theories. Keltner *et al.* [28] worked with six basic emotions: happy, love, anger, fear, surprise, and disgust. Earlier, Plutchik [42] proposed eight basic emotions, including anticipation and trust. Our work focuses on five common emotions, which are: happy(H), sad(S), fear(F), love(L), and anger(A). If none of these emotions exist in a particular example, it is classified as neutral.

Researchers adopt several approaches for emotion classification, such as lexicon-based and machine learning-based approaches. Lexicon-based approaches require an independent dictionary of words and their emotion mappings. Lexicon-based approaches do not give satisfactory results for sentences containing multiple emotions, as shown in examples E1-E3. The complexity of opinions may be observed in these examples, which demonstrate how lexicon-based techniques fail to deal with them.

(E1.) Wo Buhat ghusay (A) aur mayoosi (A) ka shikar ha:

وہ بہت زیادہ غصے اور مایوسی کا شکار ہے

They were very angry (A) and frustrated (A).

Description of E1: It holds two keywords, both keywords belong to the same emotion category, which is related to anger. The relevancy of these keywords with the same emotion category is helpful for ED using a lexicon-based approach.

(E2.) Yahan pr axhy (H) km pr b saza (S) milti ha:

یہاں اچھے کام پر بھی سزا ملتی ہے۔

Here, good deeds also go punished.

Description of E2: There are two adjectives to this sentence. The first adjective's emotional orientation is happy, and the second adjective is sad. Lexicon-based approaches assign neutral emotion (equal number of the lexicon of both emotions H-1, S-1). However, the actual emotion marked by the annotator is *sad*.

(E3.) Kafi dinon bad bryani kha kr bht kush(H) aur purjosh (H) tha lakin adil ky ravayye nay mujay gamgeen (S) kr dia:

کافی دنوں بعد بریانی کھا کر میں بہت خوش اور پر جوش تھا لیکن عادل کے رویے نے مجھے غمگین کر دیا۔

I was happy and energetic eating biryani after a long time, but Adil's behavior made me sad.

Description of E3: It holds two happy adjectives and one sad adjective: the Lexicon-based approach assigns happy emotion (maximum number of happy words). However, the actual emotion marked by the manual annotator is sad.

There are three types of machine learning approaches. The first type is supervised machine learning, which requires labeled data for training and testing. The second strategy is unsupervised, which does not require labeled data for training and testing. A third method is a semi-supervised approach that utilizes both labeled and unlabeled data. In general, supervised learning techniques are chosen over unsupervised and semi-supervised techniques for ED due to their high specificity. [47].

In practice, ED can be classified at different levels like document level, sentence level, and paragraph-level [43]. In this paper, our focus is on the building of *UNED* corpus for paragraph-based ED and sentence-based ED. This work is being carried out in phases. We collect data from various blogs, newspapers, novels, books, and short tales. In the second phase, the data is manually annotated by certain Urdu annotators. In the third phase, we

utilize this labeled *UNED* corpus to perform classification. Our whole sentence-based corpus has 52,000 samples, whereas our entire paragraph-based corpus contains 2,000 samples of six distinct classes.

1.1 Contributions

The significant contributions are as follows:

- A large-scale collection and manual annotation of the 52,000 sentence-based and 2,000 paragraph-based labeled *UNED* corpus of Urdu ED. The subset of the Urdu Nastalique Emotions Dataset [*UNED*] is freely available to the research community in order to extended the research on this topic ¹.
- Emotion detection of Urdu language dealing with five common emotions of happy, sad, fear, anger, and love, as opposed to sentiment analysis of Urdu language.
- To ensure corpus quality, data annotation quality validation should be performed by applying inter-annotator agreement.
- Construction of word embedding for *UNED* corpus and propose a deep learning-based model for classifying emotions in the *UNED* corpus.
- Results show that the proposed model performs better than generic machine learning approaches with an F1 score of 85% on *UNED* sentence-based corpus and 50% on *UNED* paragraph-based corpus.

1.2 Paper Organization

The rest of the paper is set out as follows: Section 2 comprises the related work about different ED corpus and their classification approaches. Section 3 shows the overview of our corpus generation, annotation process, and the ED techniques applied to the annotated corpus. Section 4 shows the experimental setup detail. Section 5 shows our results and analysis from various approaches. In the last section, we conclude our work and describe future work.

2 RELATED WORK

There are numerous types of emotion, each representing a particular human emotion. These distinct human emotions also affect human behavior. In the literature, we discussed the work of the ED corpus, which contains textual data from a variety of languages, including English, Arabic, Urdu, and a range of others.

Jinkun *et al.* [14] created an automated corpus of Chinese emotional text from written dialogue. Cheng *et al.* [15] concentrated on numerous user corpora from Chinese microblogs, and they took into account the emotion causal relation, which improves annotation accuracy. Yang *et al.* [62] conformed to the fundamental rules for data annotation as outlined by a linguistic expert and achieved high-quality annotations. Quan *et al.* [43] built a Chinese corpus for expression analysis, and to assure the annotation's credibility, the corpus was annotated separately by three distinct annotators. To analyze the emotion in metaphorical expressions in Chinese language Zhang *et al.* and Dongyu *et al.* [64, 65] constructed corpus based on metaphor.

Minato *et al.* [35, 36] constructed a emotion corpus for **Japanese language** and concluded that machine learning based approaches give better recall than lexicon based approaches. Takashi *et al.* [61] proposed a technique to extract features for emotion classification of Japanese language. Sato *et al.* [50] work on emotion classification using deep learning model for efficient estimation of intensities. Jung *et al.* and Do *et al.* [18, 26] proposed a corpus-based approach for **Korean language** emotion classification. To confirm the annotation's credibility, a single example was annotated by three distinct individuals. Do *et al.* [18] create an automated emotion lexicon and proposed a feature engineering technique that was helpful to enhance the performance of machine learning-based algorithms.

¹<https://github.com/farrukhbashir/UNED>

Tocoglu *et al.* [58] build a corpus of ED for **Turkish language**. It contains six basic emotions and shows that SVM gave better results instead of others. Tocoglu *et al.* [57] then generated corpus and applied machine learning algorithm using Weka tool. They have achieved 86% accuracy on this dataset. Kuijper *et al.* [30] created an additional training data for **Spanish language** by translating data from other languages. Vijay *et al.* [60] creates an emotion corpus for **English and Hindi** mixed text data. The SemEval competition has performed a lot of effort from the last few years for ED classification. The corpora used in these competitions became a standard benchmark for ED Corpora for **Arabic, English, Spanish languages** [2, 12, 30].

Researchers employed a variety of techniques to extract emotion from textual data. The four most prevalent approaches are keyword-based, Lexicon-based, machine learning-based, and hybrid approaches.

The initial effort in determining textual emotions focuses on keyword detection. This approach recognizes classes by utilizing a set of specified keywords. Samonte *et al.* [49] used spotting for keywords to identify emotions from textual data. They obtained low accuracy due to a short keyword database. Mohammad and Turney [37] have suggested a lexicon-based approach in 2013. They considered those words that occur consistently in the Google N-Gram corpus. In Urdu lexicon-based approach Rehman and Bajwa [45] used lexicon for sentiment analysis. They translated an English-language dictionary into Urdu. However, lexicon-based approaches are limited to local minima, making them ineffective for large-scale ED problems.

Ali *et al.* [56] had developed a lexicon for the purpose of sentiment analysis. They extracted "sentiunit" using a shallow parsing approach. They had given the term "sentiunit" to the terms that denote sentence polarity. Negation processing is crucial in sentiment analysis since it inverts the sentence's polarity. There are no, never, and not negation instances, but some other cases like a prefix (e.g., bay faida) and implicitly (absence of negation). Aslam *et al.* [55] have proposed a phrase level negation handling for sentiment analysis of Urdu language. Emotions are extracted in response to numerous psychological depression variables such as negativity and uncertainty. Identifying these variables is the most confusing issue that psychiatrist faces when treating a patient. Kodati *et al.* [17] proposed a deep learning models with global vector representations (GloVe) embeddings to capture text sequences of data. M. Ishtiaq *et al.* [6] they also present a lexicon-based approach for sentiment analysis on 26,614 sentences. Krishnan *et al.* [29] provides a Naive Bayes based ED approach. Zhao *et al.* [66] addressed CNN's approach to identification of emotions in text dialogue without the use of keyword extraction and segmentation. Badaro *et al.* [11] worked on Arabic ED and applied different machine learning approaches with multiple pre-processing steps. Smetanin and Sergey [53] use Bi-directional LSTM. They achieved a 72% F1 score.

Bouhekif *et al.* [13] utilized a deep learning technique with a variety of various model combinations. The final prediction was made using the probabilities associated with the three models. They had obtained a 74% F1 score. Mukhtar and Khan [38, 39] applied a machine learning approach for polarity analysis on the Urdu language. They achieved an 83.2% accuracy with the lexicon-based approach and a 67% accuracy with KNN. There is no need for labeled corpus in an unsupervised approach. Using word embedding, Mirko *et al.* [32] developed an unsupervised model for detecting emotions. Zheng *et al.* [21] provide a semi-supervised ED approach. Fathy *et al.* [22] present a hybrid model for ED based on ontology and semantic similarity of keywords. They used semantic similarity to detect emotion when the ontology base did not match. They had obtained a 76% F1 score.

Existing Bag-of-Terms (BOW)-based approaches were ineffective for complex phrases, including more positive than negative words and vice versa. Furthermore, Hassan and Shoaib [24] used the segmentation approach for determining the sub opinion information within Urdu text. Three steps should be included in this procedure. The initial step was sentence segmentation, and the second was the computation of the polarity score, and the third was the identification of sentiment polarity.

While significant work on ED is being done in English and other languages, preliminary research is being done in the resource-constrained Urdu language. The majority of researchers solved the Urdu classification problems using basic classifiers. While researchers are attempting to determine the polarity of phrases in the

Urdu language, no well-known study has been conducted to date to detect emotions. In this paper, we generate a labeled *UNED* corpus for ED and describe a system for identifying emotions in Urdu text via a deep learning approach.

3 PROPOSED METHODOLOGY FOR CORPUS GENERATION AND CLASSIFICATION

The overview of the proposed framework is shown in Figure 2 with each component of the framework explained in the following content. The overall system works by scrapping raw data (Urdu texts) from different sources and cleaning the data for duplicates, links, and so on. After the data is cleaned, annotators provide it for annotation, which assigns emotion labels to data. Further, we split into the training and testing sets, and the features are extracted from the data using either one of the three feature engineering techniques, namely count vectorization, Tf-Idf, and word embedding.

After feature extraction, machine learning and deep learning models are trained on the training data and then tested on the test data. The model is evaluated, and hyperparameters are tuned to get optimal results.

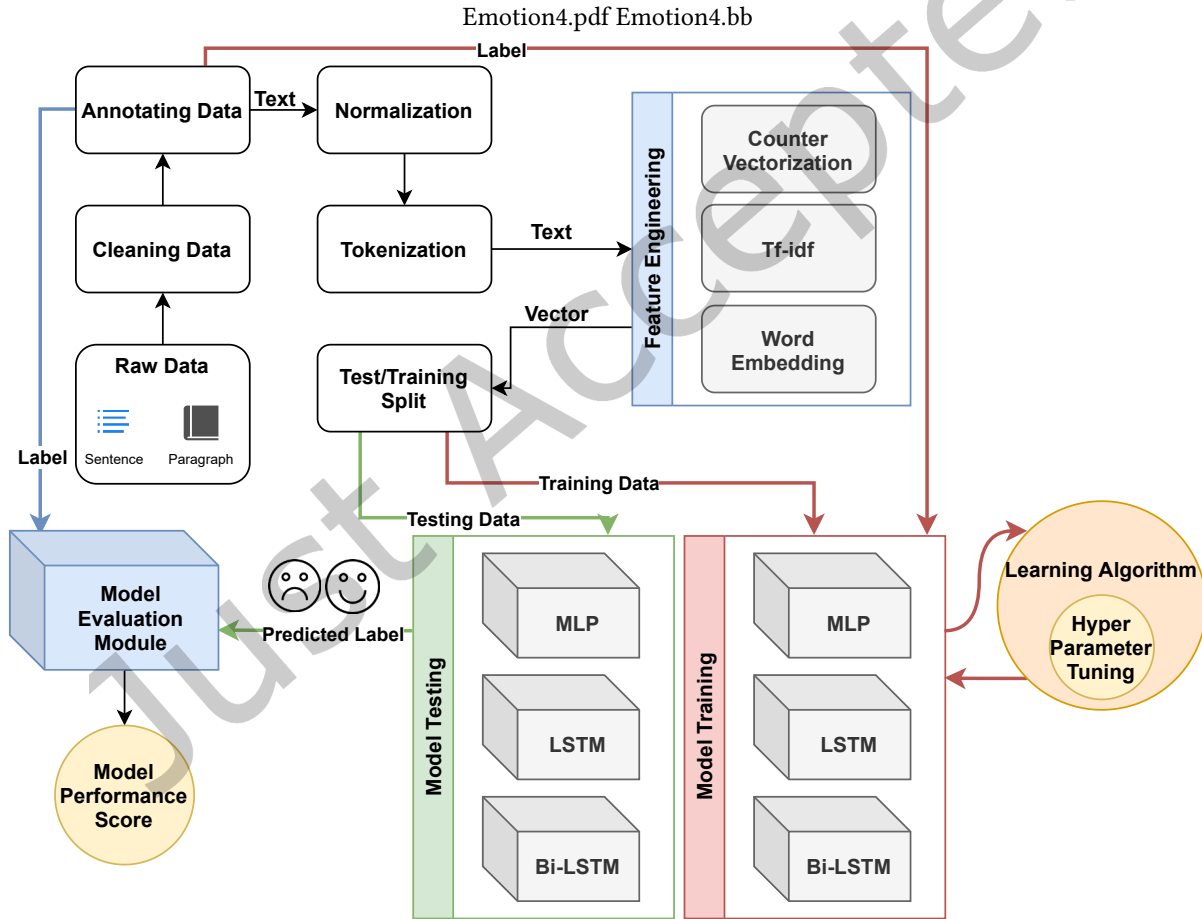


Fig. 2. Proposed work flow diagram for Urdu emotion detection on UNED corpus

3.1 Data Collection

There is a huge amount of corpus available in other languages to get better results in classification, but there is no publicly available labeled Urdu emotion-based corpus to the best of my knowledge. In this paper, we created Urdu Nastalique Emotions Dataset ² (*UNED*) for Urdu ED. While generating an emotion-based corpus, the important thing is to target those websites and blogs where emotions exist. We crawl various websites and collect an Urdu text corpus. The crawled data belong to various domains such as news, entertainment, sports, blogs, stories, novels, and many others where emotions are higher. The data is crawled from different social websites like Nawa-i-Waqt ³, hamariweb ⁴, BBC Urdu ⁵, Urdu Books websites and Urdu point ⁶ as shown in Table 1. We select these different sources based on the following criteria.

- The available text on these resources is formed by native Urdu speakers.
- All writers have a different background to ensure the diversity of multi-domain text.

Table 1. Statistics of Urdu raw data collected from various sources

Source	Sentences	Tokens	Vocabulary
Urdu Planet	4,978,629	79,231,431	572,253
BBC Urdu	341,726	9,676,253	98,346
Urduvoa	98,332	1,752,353	45,819
Urdu Library	97,384	1,721,537	44,273
Minhaj Books	84,142	1,563,521	40,974
Urdu Point	99,138	1,963,678	45,338
Nawai Waqt	94,227	1,876,876	44,346
Jang news	88,347	989,224	39,998
Wikipedia	1,223,461	32,473,556	327,889
Other	91,987,986	239,235,425	2,973,778
Total	107,093,283	2,270,482,854	9,233,014

Small sentences are difficult to categorize due to their lack of context. As the length of the statement increases, the probability of various emotions usually increases. As a result, each side has its advantages and disadvantages. In this paper, we construct a *UNED* dataset for both Urdu sentences and paragraphs. The following constraints are involved in creating the *UNED* corpus based on sentence-based data.

- Firstly, we split sentences based on dash/dot, which exists at the end of the sentence.
- The length of one sentence in a sentence-based corpus is fixed to 90 words.
- The sentences which have a length of fewer than three words are discarded.

The following constraints are involved in creating the *UNED* corpus based on paragraph-based data.

- For paragraph, we split sentences based on dash/dot, which exists at the end of the sentence.
- The length of the paragraph-based corpus is fixed to 5 sentences and 450 words.
- The sentences which have a length of fewer than 15 words are discarded.

²<https://github.com/farrukhbashir/UNED>

³<https://www.nawaiwaqt.com.pk>

⁴<https://hamariweb.com/>

⁵<https://www.bbc.com/urdu>

⁶<https://www.urdupoint.com>

Because in this information, the confidentiality of user credentials is a primary concern, so only data that was publicly available on the internet have been recorded. There was no Personally Identifiable Information in the collected data.

We get many unrelated sentences when we split based on dashes, e.g., only dashes in one sentence. We apply fixed position restrictions and many others to fix such problems. In the pre-processing step, we remove the sentences consisting of poetry and Arabic text due to minimum examples of such kind of data.

3.2 Dataset Annotation

After obtaining crawled data, it is necessary to annotate it, as the crawled data is unlabeled. A group of diverse volunteers who are native Urdu speakers is chosen to annotate data. We have 30 different annotators, each of whom attends a different college or university and resides in a different city around Pakistan. The majority are students, educators, and domain experts with extensive knowledge of Urdu literature. The average age of annotators is between 21 and 25, but some are as old as 27 to 51. We use five basic emotion categories suggested by Paul Ekman [28] which are Happy, Sad, Anger, Fear, and Love. Each annotator is assigned random sentences and paragraph examples, and they are guided to classify each example into one emotion category. Some sentences have no emotion category; instead of finding the emotion category for those sentences, we gave the facility of neutral class. The annotators were given the following guidelines:

- Assign a single emotion to each example from the given above five emotions.
- If an example does not fall in any of the above emotion categories, it is annotated as *Neutral*.
- If an example contains multiple emotions, it is assigned a label of that emotion expressed with a higher intensity.

3.2.1 Emotional Keyword Generation. Most of the examples belonged to sad emotion at the beginning of annotation and minimum examples of other emotion categories. To find sentences belonging to other emotions, we adopt the keyword-based data generation technique [49]. There is no well-known list of Urdu emotional keywords to the best of our knowledge, with the proper guideline of a domain expert. We successfully found almost 50 keywords for each emotion category. Then we extract sentences and paragraphs based on those keywords. Additionally, distribute them to annotators for annotation. We ensure that the sentences have diversity. Many of the words are related to emotions, while some have a neutral meaning. After complete annotation of the corpus, we cross-check the quality of these emotional keywords against those samples containing only one emotion keyword. We have achieved 90% accuracy; the 10% that are inaccurate are those in which negation appears within the word structure.

3.2.2 Data Annotation Quality Validation. Data validation is essential to obtain accurate performance. We use three distinct annotators to perform annotation on a single example. In this validation process, there are three possible points of agreement. If three annotators establish consensus and two annotators assign the same emotion category while the third annotator provides a different category, the final emotion category is assigned based on a majority vote. Reassessment is undertaken if all three annotators assign three unique emotion categories to a single example. Those examples are discarded from the dataset if an identical result is obtained. There are various inter-annotator agreement (IAA) evaluation measures like Cohen's kappa, Fleiss' kappa, Krippendorff's alpha [8]. We choose Fleiss's Kappa for inter-annotator agreement, as Fleiss's kappa coefficient indicates the degree of agreement between more than two annotators who classified N items into C mutually exclusive categories. Where C is Happy, Angry, Sad, Love, and Neutral, n is the total number of sentences. Fleiss' kappa can have multiple annotators, and it gives more accurate results where more than two annotators annotate data. In the case of Fleiss' kappa, every item is not necessarily annotated by each annotator. We perform Fleiss's *kappa inter-annotator agreement*, by computing k using following Equation 1:

$$K = \frac{P(a) - P(e)}{1 - P(e)} \quad (1)$$

Where the factor $1 - P(e)$ gives the degree of agreement that is attainable above chance, $P(a) - P(e)$ gives the degree of the agreement achieved above chance. If the annotators annotated each data point the same, they are incomplete agreement then $\kappa = 1$. If there is no similar data point of different annotators, then there is no agreement among the annotators, and κ is less than or equal to zero. Table 2 shows our inter-annotator agreement and disagreement on our dataset.

Table 2. Inter-Annotator agreement and disagreement between different annotators.

Label	Agreement
Agreement	61.37%
Disagreement	38.63%
Total	100%

We obtain a value of 61.37. It shows the best possible level of agreement among annotators. What is the meaning of a number produced by Fliess κ ? What does that Specific value mean? According to Anthony et al. [59] commonly cited scales are shown in Figure 3. In our case IAA is 79.37% as shown in Table 2, which is a substantial agreement according to Landis and kooch 1977 and Green 1997 Kappa interpretation.

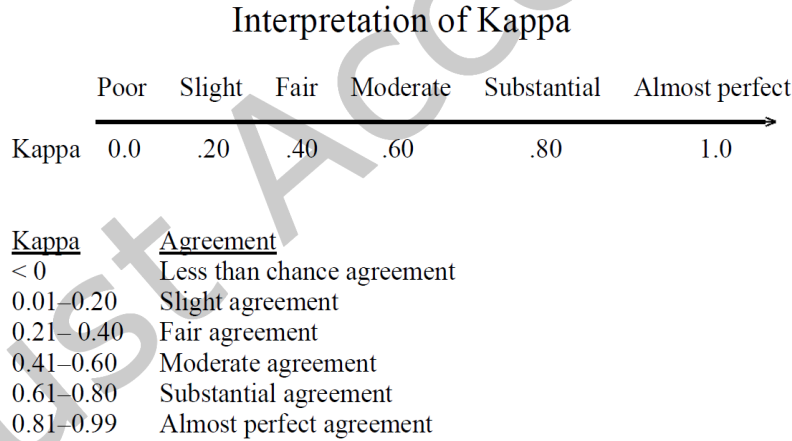


Fig. 3. Inter-annotator agreement scale to measure the suitability of provided manual for annotations.

3.2.3 Corpus Statistics. We made two kinds of the corpus, i.e., sentence base and paragraph-based corpus. Our annotated sentence-based Corpus contains 52,000 sentences, and paragraph-based corpus contains 2000 examples. There are 15 tokens in one sentence and 270 tokens in one paragraph in our annotated corpus. The complete sentence-based corpus contains 49020 tokens, and the paragraph-based corpus has 21,648 tokens. Moreover, complete detail of our original cleaned corpus labeled corpus with no. of sentences w.r.t each class is shown in Table 3.

Table 3. Cleaned corpus statistics

Whole Corpus Statistics	
Count of Sentences	200000
Count of Total words	3,758,000
Unique Word Count	30,000
Average no of words in sentence	18
Sentence Corpus Statistics	
Count of Sentences	52,000
Total number of tokens	947,500
Unique words count	27,590
Average number of words in sentence	15
Maximum number of words in sentence	20
Paragraph Corpus Statistics	
Count of Paragraph	2000
Total number of tokens	556,000
Unique words count	24,460
Average number of words in Paragraph	270
Maximum number of words in Paragraph	380
Emotion Category Statistics (Sentence)	
Total Number of Emotions	6
Count of Happy sentences	10760
Count of Sad sentences	14568
Count of Anger sentences	1327
Count of Love sentences	10427
Count of Fear sentences	1100
Count of Neutral sentences	13818
Emotion Category Statistics (Paragraph)	
Total Number of Emotions	6
Count of Happy paragraphs	430
Count of Sad paragraphs	598
Count of Anger paragraphs	99
Count of Love paragraphs	377
Count of Fear paragraphs	126
Count of Neutral paragraphs	370

After full annotation, we set a different count of examples belonging to different emotions as shown in Figure 4 and Figure 5. Total frequency of labelled corpus of sentences and total frequency of labelled corpus of paragraph shown in Table 3.

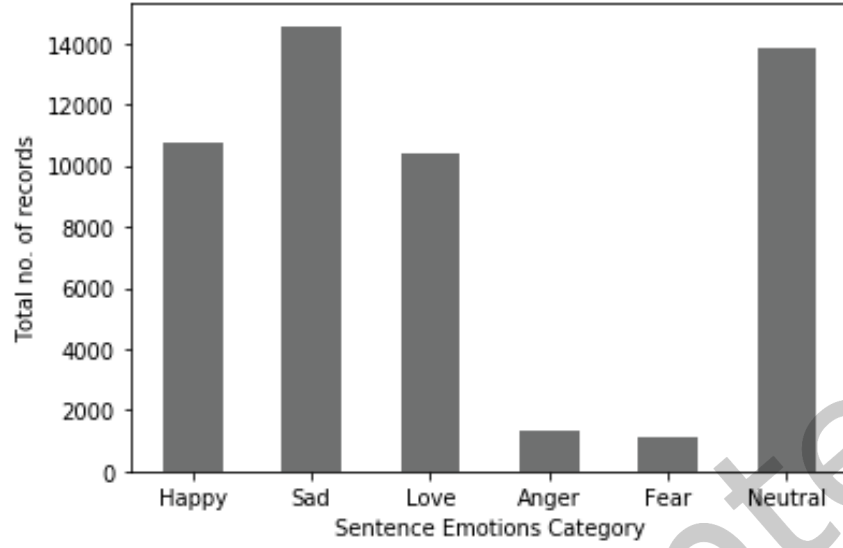


Fig. 4. Frequency of sentence based total labelled corpus

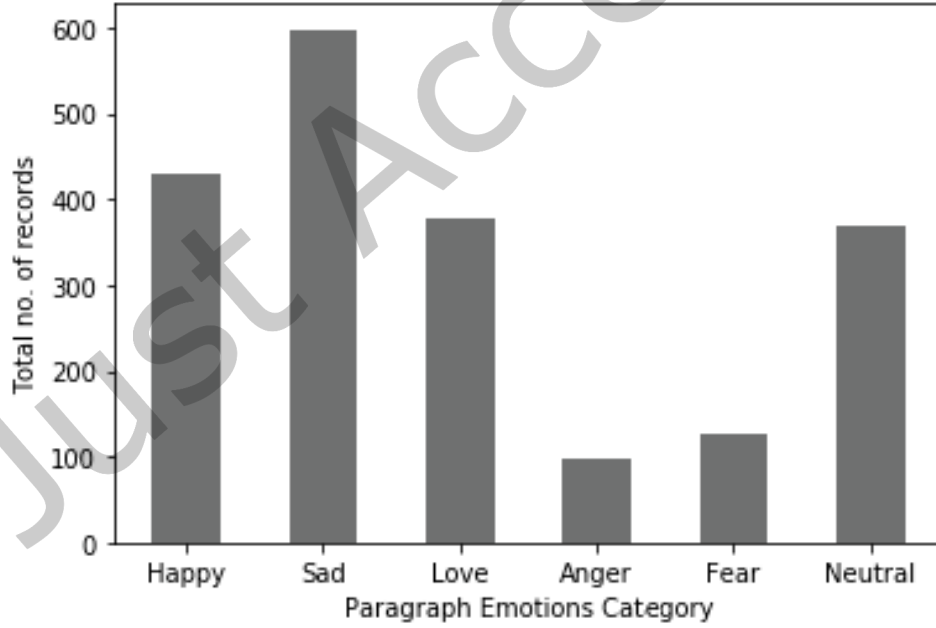


Fig. 5. Frequency of paragraph based total labelled corpus

3.2.4 Data Analysis. When we examine the data regarding the vocabulary of emotions, it was observed that the most frequent happy words are [Shandar - شاندار - Glorious], [Mazak - مزاق - Joke], [Shafqat - شفقت - Compassion], [Fakhar - فخر - Pride], [Purjosh - پر جوش - Excellent], [Dilchasb - دلچسب - Interest], [Anaam - انعام - Prize], [Dilkash - دلکش - Attractive], [Puritmad - پراعتماد - Confidence], [Hosla Afzai - حوصلہ افزائی - Encouragement], [Aman - امن - Peace], [Fateh - فتح - Victory], [Faida - فائدہ - Benefit].

The sad words are [Moat - موت - Death], [Preshan - پریشان - Anxiety], [Dhamakha - دھماکہ - Explosion], [Gham - غم - Sadness], [Bimar - بیمار - Sickness], [khudkashi - خودکشی - Suicide], [Takhleef - تکلیف - Trouble], [Nuqsan - نقصان - Loss], [Burai - برائی - Evil], [Badqismat - بد قسمت - Unfortunate], [Zulam - ظلم - Cruelty], [Rona - رونا - Weep], [Soag - سوگ - Mourning].

The love words are [Ishq - عشق - Love], [Tareef - تعریف - Admire], [Jazba - جذبہ - Passion], [Hamdard - ہمدرد - Sympathy], [Romanwi - رومانوی - Romantic], [Meetha - میٹھا - Sweet], [Mehboob - محبوب - Beloved], [Khobsurat - خوبصورت - Beautiful], [Mukhlis - مخلص - Sincere], [Dilbar - دلبر - Friendly], [Rahmdil - رحم دل - kind hearted], [Qabil Purtash - قابل پرستش - Adorable], [Rehmat - رحمت - Blessed].

The anger words are [Jarhana - جارہانہ - Aggressive], [Tashaddud - تشدد - Violence], [Antiqam - انتقام - Retaliation], [Nafrat Angaiz - نفرت انگیز - Hateful], [Pagal - پاگل - Mad], [Mushtail karna - مشتعل کرنا - Provoke], [Ghadab - غضب - Rage], [Larai - لڑائی - Fight], [Adawat - عداوت - Hostility], [Thapard marna - تھپڑ مارنا - Cuffing], [Toheen Amaiz - توہین آمیز - Insulting], [Dhokha - دھوکہ - Cheat], [Dhamkhi - دھمکی - Menace].

The fear words are [Khof Zadah - خوف زدہ - Scared], [Dehshat Zada - دہشت زدہ - Alarmed], [Mashkook - مشکوک - Suspicious], [Khoof - خوف - Terror], [Khoofnaq - خوفناک - Dreadful], [Ghair Mustahikm - غیر مستحکم - Unsettled], [Adm Itmad - عدم اعتماد - Distrustful], [Kanpna - کانپنا - Shudder], [Buzdil - بزدل - Timid], [Adam Tahaffuz - عدم تحفظ - Insecure], [Dhachkha Lagna - دھچکا لگنا - Shocked], [Uljan - الجھن - Confused], [Byanak - بھیانک - Grisly].

3.2.5 Dataset sample. Dataset sample is attached in Table 4.

3.2.6 Data Normalization. The fundamental goal of normalizing Urdu text is to represent it in a consistent format. As previously stated, Urdu is a synthesis of several languages. As a result, some characters were assigned to different Unicode subsets for the same script, resulting in divergence. Certain characters combine to form a compound character. For example, alif-madd can be composed of two characters (Unicode 0627+0653) and one

Table 4. Annotated UNED sentence-based corpus samples

Example	Emotion Class
مجھے تم پر بہت غصہ ہے	Anger
میں غصے میں وہاں سے واپس آگیا	Anger
تم بہت بد تمیز ہو چلے جاؤ یہاں سے	Anger
بہت دکھ ہوا ان کی وفات کا سن کر	Sad
مجھے میچ ہارنے کا بہت دکھ ہے	Sad
مجھے بہت افسوس ہوا	Sad
اپنی جیت کی خوشی میں اس نے مجھے دعوت دی	Happy
مجھے آپ کی شادی کی خبر سن کر بہت خوشی ہوئی	Happy
میں آج بہت خوش ہوں کے تم پاس ہو گئے	Happy
تم سناؤ کیسے ہو	Neutral
آج سے اس کام کو شروع کرتے ہیں	Neutral
کلی میں لاہور جاؤں گا	Neutral

character (Unicode: 0622) [25]. All of these characters are allotted with a unique Unicode. We use Unicode standard normalization forms as suggested in this paper [25].

3.3 Feature Extraction Techniques

Feature extraction is performed using three techniques, *count vectorization*, *tf-idf* and *word embeddings*.

The complete proposed algorithm for ED from Urdu sentences is shown in Algorithm 1. Four procedures form this algorithm: the first is data generation, the second is data annotation, the third is a classification of examples, and the fourth and final method is the main procedure. In main, both functions, i.e., *data_annotation* and *classification*, are called, which returns the particular emotion. Line 1 shows the data generation procedure; in this procedure, we target different websites for scraping Urdu text, as shown in line 2. Lines 3 to 4 represent the text splitting process, and line 5 represents the fixed-length restriction of words in a sentence. If there are only three or more than 90 words in a sentence, it is discarded. Lines 11 to 15 represent the labeled data process collected from different annotators. Line 17 is the classification procedure definition. In this procedure, we perform the pre-processing phase, which is involved segmentation of words, text normalization, and duplication sentence removal, after this pre-processed data is passed to the neural network for further classification.

These are sparse vectors containing 0s for all the words absent from the example. **Count Vectorization:** is used to construct vectors with the exact dimensions as our vocabulary. If a word appears in the sentence or

Algorithm 1 ALGORITHM FOR EMOTION EXTRACTION FROM SENTENCES

```

1: procedure DATA_GEN(website)
2:   data  $\leftarrow$  scrap_data(website)
3:   if ((data == ' - ')) then
4:     text  $\leftarrow$  split(data)
5:   if ((text_length > 3)&&( text_length <= 90)) then
6:     sentence  $\leftarrow$  text
7:   else
8:     discard(text)
9:   return sentences
10: procedure DATA_ANNOTATION()
11:   url  $\leftarrow$  website_url
12:   Sentences  $\leftarrow$  DATA_GEN(url)
13:   sentence  $\leftarrow$  labels(Sentences)
14:   return sentences
15: procedure CLASSIFICATION(sentence)
16:   for each s  $\in$  sentence do
17:     dataframe  $\leftarrow$  normalizeText(s)
18:     Unique_text  $\leftarrow$  Duplicate_Removal(dataframe)
19:     Seg_Word  $\leftarrow$  tokenize(dataframe)
20:     Emotions  $\leftarrow$  NN(wordToVec_Emb(Seg_Word))
21:   return Emotions
22: procedure MAIN()
23:   input website_url
24:   DATA_ANNOTATION()
25:   CLASSIFICATION(sentence)

```

paragraph, one is placed to the dimension corresponding to that word. Every time that word is used again, the count goes up by one.

Tf-idf: $Tf - idf$ represents term frequency and inverse document frequency. This weight is a mathematical measure used to evaluate precisely how frequent and important a term is in a piece of text. $Tf - idf$ is calculated by multiplying two metrics: the frequency of a word in a document and its inverse frequency about other words.

Word Embedding: Machine does not understand the human written textual data. We use word embedding to convert the Urdu textual data into a numerical vector representation that the machine can understand.

Mostly used word embedding are *mikolov 2013* [34]. There are two variants of it, a continuous bag of words and skip-gram. These word2vec are mostly available for English and other languages, but there are no well-known embedding for Urdu text related to emotions. We build our embedding. We train word embedding on our dataset to make it more applicable to our case (Urdu Sentences containing emotions) due to the absence of any good dataset. If a word was not included in the list, we allocate it as suggested by Pandya [41] with a uniform distribution at $[-0.25, 0.25]$; since the outcomes of uniform distribution are equally likely; every variable has the same probability of occurring. Each variable has an equal opportunity to appear with this distribution, but there may be an endless number of points.

We use the *gensim* library in python to train word2vec. Word2vec is also known as word embedding. We use the skip-gram model on ten lacs yelp. To train word2vec, we examine different hyperparameters, such as context

window, min word count, and word dimensions. These parameter values significantly affect the quality of the word vectors. We use some parameters with their default values, which we adjust as given below.

- Word dimension: It is used to control the size of our embedding. We are using the 300 dimensions.
- Min word count: It is used to control the word count, the minimum number allowed in the word2vec model.
- Number of workers: It is used in the training process to specify the number of threads required to achieve an adequate level of parallelism.
- Context: It is the size of the context window. We are utilizing a context window with a size of 5.
- Downsampling: we use it to control the frequent words in the text corpus.

All terms are tokenized based on space and punctuation marks to handle unidentified words. Then we measure the word frequency of these words, and the term "UNK" replaces all such words with a frequency of less than 6.

3.4 Deep Learning-Based Classification

We apply different Deep learning models for our tasks like MLP, LSTM, and BiLSTM to model ED from the text as a classification task. We need to assign happy, sad, anger, love, and fear class to an example.

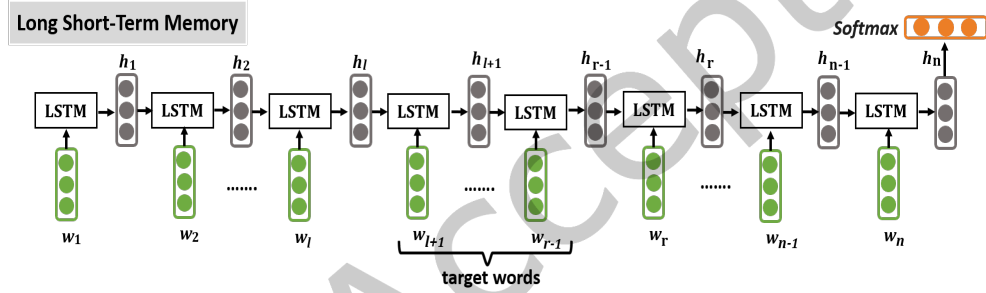


Fig. 6. The basic Long Short-term Memory (LSTM) Approach. x stands for word in a sentence whose length is n , $\{x_{l+1}, x_{l+2}, \dots, x_{r-1}\}$ represents target words, whereas $\{x_1, x_2, \dots, x_l\}$ represents previous context words, $\{x_r, \dots, x_{n-1}\}$ are following context words.

3.4.1 Network architecture: This section describes a long short-term memory (LSTM) model for ED. We used a basic variant of LSTM. LSTMs are state-of-the-art in the area of sentiment analysis and ED. They are capable of multi-level inference and preserving long-distance dependencies between words. The representation of the sentence can naturally be considered a feature for predicting sentence emotion. We train our LSTM classifier for ED in Urdu text and hyper-tuned it for improving its performance.

In particular, each word is mapped onto a real-valued vector, also known as word embedding. Within a word embedding matrix, all term vectors are stacked as $L_w \in \mathbb{R}^{d \times |V|}$, where d stands for the dimension of word vector and $|V|$ is vocabulary size. An example of the model is shown in Figure 6. LSTM is a variation of the recurrent neural network (RNN) that transforms the current word vector x_t with the output vector of the previous time-step a_{t-1} while passing on long term dependencies between words calculated using Equation 2. In addition, the RNN transition function is a linear layer followed by a point-wise non-linear layer such as the hyperbolic tangent (\tanh) function.

$$a_t = \tanh(W \cdot [a_{t-1}; x_t] + b) \quad (2)$$

where $W \in \mathbb{R}^{d \times 2d}$, $b \in \mathbb{R}^d$ and d is the word vector dimension. However, RNN has a gradient vanishing or exploding, where gradients over long sequences can disappear or grow exponentially. Several researchers have used a different variation of the LSTM cell for transformation that improves the preservation of semantic correlations in a series. The LSTM cell includes three additional gates over regular RNN: an input gate, a forget gate, and an output gate. Such gates optimally remember vector input, forget history and produce vector output. LSTM cell is calculated using below Equations 3, 4, 5, 6, 7, and 8.

$$\tilde{c}_t = \tanh(W_c \cdot [a_{t-1}; x_t] + b_c), \quad (3)$$

$$i_t = \sigma(W_i \cdot [a_{t-1}; x_t] + b_i), \quad (4)$$

$$f_t = \sigma(W_f \cdot [a_{t-1}; x_t] + b_f), \quad (5)$$

$$o_t = \sigma(W_o \cdot [a_{t-1}; x_t] + b_o), \quad (6)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}, \quad (7)$$

$$h_t = \tanh(c_t) \odot o_t, \quad (8)$$

where \odot is used for element-wise multiplication, σ is the *sigmoid* function, whereas $W_i, b_i, W_f, b_f, W_o, b_o$ are parameters of the LSTM gates. The activation for each time step is calculated for each word in the sentence, and the final activation vector can be considered the sentence's encoding. This encoding is fed into a softmax layer with as many neurons as classes. This layer classifies the sentence as happy, angry, sad, love, fear, and neutral. Softmax is computed as given by the Equation 9:

$$\text{softmax}(c_i) = \frac{\exp(c_i)}{\sum_j \exp(c_j)}, \quad (9)$$

Where j is the number of emotions being classified.

We apply an LSTM network model for ED. To convert all sentences into vector form, we use our trained word2vec model. In our case, an Adam optimizer is used. Softmax as an activation layer has been used because there are more than two classes. Softmax is used for multi-class classification using probabilities, and classes with higher probability will be assigned to that example. Different parameter tuning of optimizer and activation is given in Table 5.

Table 5. Different optimizer and activation tuning

Parameter	Candidates	Best parameter
Optimizer	sgd, RMSprop, Adam	Adam
Activation Layer	tanh, relu, sigmoid, SoftMax	SoftMax

As we know, the model's performance is highly affected by the hyperparameter of the LSTM network. To make our model more efficient for ED, we tune that parameter with different values. The detail of these different values is given in Table 6.

Table 6. Different hyperparameter tuning

Parameter	Ranges	Best parameter
max_nb_words	10k -50k	40k
embedding dim	300	300
batch size	32 -500	64
lstm dim	32 -800	64
drop out	0.2 -0.5	0.5
learning rate	0.1 -0.009	0.0003

3.4.2 Training Model. Initially, the dataset is split into training and testing. We use 80% data for training and 20% data for testing. The output of our data is changed into one-hot encoding. As we defined previously, those outputs are actually in 6 categories. We assign 1 for happy, 2 for sad, 3 for love, 4 for anger, 5 for fear, and 6 for neutral class.

Primarily, we set Keras’s sequential model because it allows creating the model layer by layer. In the training model, we use multiple hidden layers. In the first layer, we use a memory unit of size 32, and in the second hidden layer, we define the size of the memory unit as 64. We set the dropout value to 0.2 and the recurrent dropout value to 0.3. The dense layer is fully connected with the hidden layer. We use a sigmoid activation function to predict the output within six different categories that are already defined. We use Adam optimizer to boost the process and optimize this categorical_cross-entropy. As we know, the LSTM model takes time during training. After some time, we save the model with the name model.h5 and load this model for further training. We use the F1 score as an evaluation metric for each epoch.

3.4.3 Testing of our trained model. We validate our trained model using the separated 20% test data. The test examples are converted into vectors in the testing phase using our generated embedding of 300 dimensions. After testing, the predicted label is mapped with an actual label. We achieve an F1 score of 85% with our model on sentence-based *UNED* corpus and achieve 50% on the paragraph-based *UNED* corpus.

4 EXPERIMENTAL SETUP

We use the viper i7 machine for experimentation. This system has 16GB RAM, a 7th generation and 3.40GHZ processor, and with Windows operating system. Our code runs using Jupyter Notebooks IDE with Python 3.0 version. We conduct word embedding training on a Microsoft Azure account with 56GB of RAM and a Windows operating system. We use three techniques for extraction of features as discussed above, i.e., TF-IDF vectorizer [3], BoW approach [51] and word embedding approach [46]. Different Machine Learning algorithms are explored: Support Vector Machine (SVM), K-Nearest Neighbor(KNN), Random Forest(RF), Multi-layer Perceptron, and LSTMs. Using training and test splits, each classifier was trained and tested separately for the *UNED* corpus.

4.1 Evaluation measures

This study performed the analysis using the most commonly used evaluation metrics, i.e., Accuracy, Precision, Recall, and F1 score as calculated in Equation 10, 11, 12 and 13.

An ED system’s Accuracy (*A*) is defined as the proportion of the total number of correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

Precision (P) is the measure of the accuracy of emotion prediction resulting from my model against the manually labeled dataset.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (11)$$

Recall (R) is the overall coverage of the model. In other words, it is the ratio of correctly predicted sentences to all sentences in the actual class.

$$Recall = \frac{Positive}{TruePositive + FalseNegative} \quad (12)$$

F1 score is the weighted average of both precision and recall. Usually, the F1 score is more beneficial than accuracy when irregular class distribution occurs.

$$F1\ score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (13)$$

5 RESULTS AND DISCUSSIONS

We create four different subsets of the *UNED* sentence-based corpus with different numbers of examples for testing purposes, i.e., 12000, 19000, 36000, and 52000 sentences.

5.1 Sentence Based Dataset Evaluations

Figure 7 shows the result by applying the proposed model on 12000 examples. In this evaluation, we prepare 3000 examples of sad, 3000 examples of happy, 2000 examples of love, 3500 examples of neutral. Both fear and anger comprise a total of 500 examples. The red line shows the F1 score on training data, and the green line shows the F1 score on validation data. We achieve the 50% F1 score on this subset of the dataset. At the initial level, the F1 score increases on both the training and validation sets. However, after a particular epoch, the F1 score on the validation set does not increase, and training remains stable.

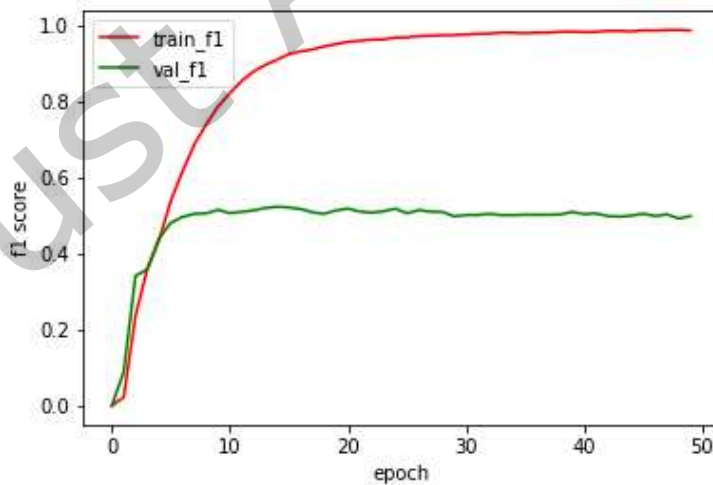


Fig. 7. Results on 12000 instances

Figure 8 shows the results of 19,000 examples. In this evaluation, we have 5,500 examples of sad emotion, 5,000 examples of happiness, 3,000 examples of love, 5,000 examples of neutral. Both fear and anger comprise a total of 500 examples. The red line shows the F1 score on training data, and the green line shows the F1 score on validation data. We got a 65% F1 score on this subset of the dataset. The number of examples present in this subset still failed to provide a significant result, so we extended the number of samples in this subset of the dataset.

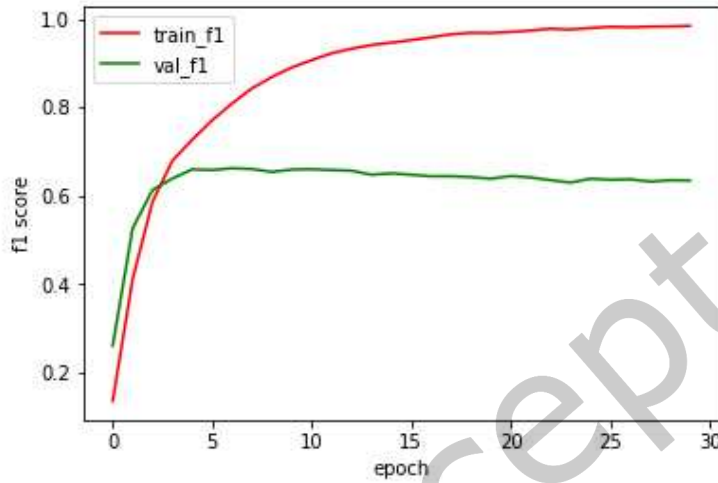


Fig. 8. Results on 19,000 instances

In Figure 9, it shows the result of 36,000 examples. In this evaluation, we have 10,000 examples of sad emotion, 5,000 examples of happy, 10,000 examples of love, 9,000 examples of neutral, 1,000 examples of anger, and 1,000 examples of fear. The red line shows the F1 score on training data, and the green line shows the F1 score on validation data. We achieve a 77% F1 score on this subset of the dataset, which is better than previously. We further extend the number of examples for getting more enhanced results.

In Figure 10, the result shows 52,000 examples. This evaluation has 13,200 examples of sad emotion, 12,000 examples of happy, 12,000 examples of love, 11,300 examples of neutral, 1,500 examples of anger, and 2,000 examples of fear. The red line shows the F1 score on training data, and the green line shows the F1 score on validation data. We achieve an 85% F1 score on this subset of the dataset. After analyzing the previous evaluation, we conclude that with the addition of examples, the results of our model improved, as shown in Figure 8. We achieve an 65% F1 score on 19,000 examples, and on 52,000 examples as shown in Figure 10, we achieve an 85% F1 score.

5.1.1 Comparison of models by using Count Vectorization approach:

In this section, we used Count Vectorization for feature extraction with a different subset of the *UNED* sentence based corpus, i.e., 12,000, 19,000, 36,000, and 52,000. Further, we compare our results with the other machine learning techniques such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multi-Layer Perceptron (MLP), Ada Boost, and K-Nearest Neighbour (KNN). The detailed precision, recall, and F1 score are shown in Figure 11. We observe that the F1 score gradually increases by increasing the number of examples.

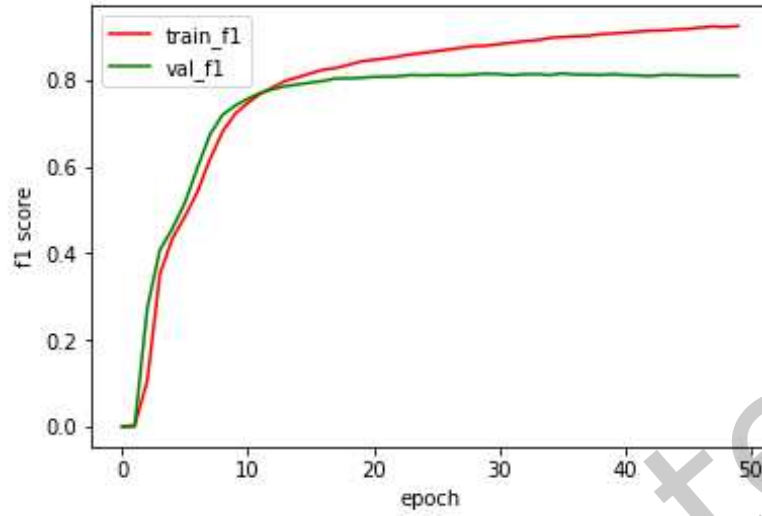


Fig. 9. Results on 36,000 instances

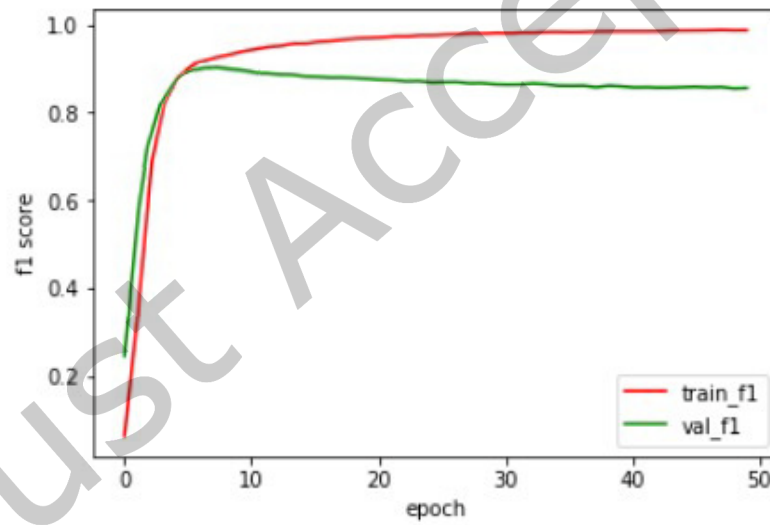


Fig. 10. Results on 52,000 instances

However, after 36,000 examples, as there are fewer sentences for two classes, i.e., fear and anger, the F1 score for the several classifiers decreases.

5.1.2 Comparison of models by using TF-IDF approach:

In this section, we use TF-IDF for feature extraction with a different subset of the *UNED* sentence-based corpus. Further, we compare our results with other machine learning techniques such as Support Vector Machine (SVM),

Decision Tree (DT), Random Forest (RF), Multi-Layer Perceptron, Ada Boost, and K-Nearest Neighbour (KNN). The detailed precision, recall, and F1 score are elaborated in Figure 12. We observe that, by an increasing number of examples, the F1 score gradually increases, except for the case of SVM.

5.1.3 Comparison of models by using Word Embedding approach:

In this section, we use our trained word embedding for feature extraction with a different subset of the *UNED* sentence based corpus, i.e., 12,000, 19,000, 36,000, and 52,000. Further, we compare our results with other machine learning techniques such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multi-Layer Perceptron (MLP), Ada Boost, and K-Nearest Neighbour (KNN). These comparisons are shown in Table 7. The detailed precision, recall, and F1 score are elaborated in Figure 13 and Figure 14. We observe that, as the number of examples increases, the F1 score gradually increases. However, after 36000 examples, the F1 score for some classifiers, such as decision trees and random forests, decreases due to the imbalanced data of classes. As a result of our findings, we can conclude that both LSTM and Bi-LSTM perform better than other classifiers because they collect more information for classification when word embeddings are used. Due to the absence of a large dataset like Google News or Wikipedia, we assume that the primary cause for average performance in our case was the lack of dataset as Google Glove and other embeddings have. Thus, our findings for Urdu text-based ED are still state of the art in the deep learning work.

Table 7. Comparison of Count Vectorizer, TF-IDF and Word2Vec approach on UNED sentence based dataset for 52K examples

Technique	Algorithm	Accuracy	Precision	Recall	F1
BOW	SVM	0.83	0.82	0.83	0.83
	KNN	0.62	0.62	0.66	0.62
	Multilayer perceptron	0.70	0.71	0.77	0.71
	Random Forest	0.82	0.81	0.81	0.81
	Decision Tree	0.64	0.64	0.64	0.62
	Ada Boost	0.84	0.84	0.85	0.84
Tf-idf	SVM	0.47	0.5	0.47	0.43
	KNN	0.42	0.39	0.43	0.40
	Multilayer perceptron	0.45	0.49	0.46	0.45
	Random Forest	0.44	0.36	0.44	0.38
	Decision Tree	0.34	0.34	0.35	0.34
	Ada Boost	0.31	0.25	0.32	0.25
WE-300	SVM	0.78	0.79	0.79	0.79
	KNN	0.63	0.66	0.63	0.63
	Multilayer perceptron	0.81	0.81	0.80	0.80
	Random Forest	0.74	0.75	0.74	0.74
	LSTM	0.85	0.82	0.88	0.85
	Bi-LSTM	0.853	0.84	0.87	0.853
	Decision Tree	0.54	0.55	0.55	0.55
	Ada Boost	0.68	0.69	0.69	0.69

5.2 Paragraph Based Dataset Evaluation

In this section, we discuss our results on *UNED* paragraph-based corpus. In Figure 15, we present our results produced on 2,000 examples. In this evaluation, we have 400 examples of sad emotion, 400 examples of happy, 400 examples of love, 400 examples of neutral. Both fear and anger comprise a total of 400 examples. The red line shows the F1 score on training data, and the green line shows the F1 score on validation data. We achieve a 50% F1 score on this *UNED* paragraph-based corpus. We see that the F1 score increases initially on both the training and validation sets. Due to the lack of additional generalization capabilities, the validated set stops growing after a certain epoch. The *UNED* paragraph based corpus is smaller than a *UNED* sentence based corpus. That is why on a sentence-based dataset, our model trained efficiently and provided a better result.

5.2.1 Comparison of models by using count vectorization approach: In this section, we use Count Vectorization for feature extraction and compare our results with other machine learning techniques such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multi-Layer Perceptron (MLP), Ada Boost, and K-Nearest Neighbour (KNN). The results with detailed precision, recall, and F1 score is shown in Figure 16. We achieve a 41% F1 score using SVM, 36% F1 score using DT, 39% F1 score using RF, 36% F1 score using AdaBoost, 39% F1 score using MLP classifier, and 35% F1 score with KNN. The possible reason for the unsatisfactory results could be that we have fewer examples in the paragraph-based *UNED* corpus, but it is the first initiative towards paragraph-based Urdu ED tasks.

5.2.2 Comparison of models by using TF-IDF approach:

In this section, we use TF-IDF for feature extraction and compare our results with other machine learning techniques such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron, Ada Boost, and K-Nearest Neighbour (KNN). We achieve a 43% F1 score using SVM, 34% F1 score using DT, 38% F1 score using RF, and 25% F1 score using AdaBoost, 45% F1 score using MLP classifier, and 40% F1 score with KNN. The results with detailed precision, recall, and F1 score is elaborated in Figure 17.

5.2.3 Comparison of models by using Word Embedding approach: In this section, we used our trained word embeddings for feature extraction and compared our proposed approach results for the *UNED* paragraph-based dataset with other states of the art machine learning techniques such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multi-Layer Perceptron (MLP), Ada Boost, and K-Nearest Neighbour (KNN). The results are shown in Table 8, while detailed precision, recall, and F1 score is elaborated in Figure 18. We note that, due to the scarcity of paragraph-based data, we cannot attain a high score. Further, we can conclude that when word embeddings are used, LSTM and Bi-LSTM both perform better than other classifiers.

6 CONCLUSION AND FUTURE WORK

Numerous studies on ED have been performed in English and various other languages. However, the Urdu language is yet to be studied for ED. There is a lack of a publicly available dataset for Urdu ED. Therefore, we have proposed a solution for Urdu ED. We present Urdu Nastalique Emotions Dataset (*UNED*) for emotion recognition from Urdu sentences and paragraphs. We propose an LSTM-based approach for Emotion Detection and classification. Further, we compare our results with other states of the art machine learning approaches. Compared to existing approaches, the proposed approach outperforms them all by a substantial margin. Additional datasets for sentence and paragraph-based ED can be developed in the future. Fear and Anger emotions are represented in a small number of sets in the *UNED* corpus. In the future, we aim to collect relevant instances of these emotions. There are a variety of semi-supervised approaches that could be used in the future.

Table 8. Comparison of Count Vectorizer, TF-IDF and Word2Vec approach on paragraph dataset

Technique	Algorithm	Accuracy	Precision	Recall	F1 score
BOW	SVM	0.42	0.43	0.43	0.42
	KNN	0.35	0.36	0.35	0.33
	Multilayer perceptron	0.38	0.38	0.38	0.35
	Random Forest	0.45	0.41	0.45	0.39
	Decision Tree	0.37	0.36	0.37	0.36
	Ada Boost	0.34	0.34	0.35	0.30
Tf-idf	SVM	0.47	0.5	0.47	0.43
	KNN	0.42	0.39	0.43	0.40
	Multilayer perceptron	0.45	0.49	0.46	0.45
	Random Forest	0.44	0.36	0.44	0.38
	Decision Tree	0.34	0.34	0.35	0.34
	Ada Boost	0.31	0.25	0.32	0.25
WE-300	SVM	0.30	0.10	0.31	0.15
	KNN	0.12	0.25	0.13	0.13
	Multilayer perceptron	0.12	0.21	0.13	0.14
	Random Forest	0.11	0.11	0.12	0.10
	LSTM	0.50	0.48	0.51	0.50
	Bi-LSTM	0.50	0.49	0.50	0.50
	Decision Tree	0.15	0.23	0.15	0.16
	Ada Boost	0.13	0.20	0.14	0.14

REFERENCES

- [1] Ahmad Abbasi, Abdul Rehman Javed, Chinmay Chakraborty, Jamel Nebhen, Wisha Zehra, and Zunera Jalil. 2021. ElStream: An Ensemble Learning Approach for Concept Drift Detection in Dynamic Social Big Data Stream Learning. *IEEE Access* 9 (2021), 66408–66419.
- [2] Malak Abdullah and Samira Shaikh. 2018. Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning. In *Proceedings of the 12th international workshop on semantic evaluation*. 350–357.
- [3] Akiko Aizawa. 2003. An information-theoretic perspective of tf-idf measures. *Information Processing & Management* 39, 1 (2003), 45–65.
- [4] Kholoud Alsmearat, Mohammed Shehab, Mahmoud Al-Ayyoub, Riyadh Al-Shalabi, and Ghassan Kanaan. 2015. Emotion analysis of arabic articles and its impact on identifying the author's gender. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 1–6.
- [5] Nourah Alswaidan and Mohamed El Bachir Menai. 2019. KSU at SemEval-2019 Task 3: Hybrid Features for Emotion Recognition in Textual Conversation. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 247–250.
- [6] Kamran Amjad, Maria Ishtiaq, Samar Firdous, and Muhammad Amir Mehmood. 2017. Exploring Twitter news biases using urdu-based sentiment lexicon. In *2017 International Conference on Open Source Systems & Technologies (ICOSST)*. IEEE, 48–53.
- [7] M. U. Arshad, M. F. Bashir, A. Majeed, W. Shahzad, and M. O. Beg. 2019. Corpus for Emotion Detection on Roman Urdu. In *2019 22nd International Multitopic Conference (INMIC)*. 1–6.
- [8] Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34, 4 (2008), 555–596.
- [9] P Ashokkumar, Siva G Shankar, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. 2021. A two-stage text feature selection algorithm for improving text classification. *ACM Transactions on Asian and Low-Resource Language Information Processing* 20, 3 (2021).
- [10] Egils Avots and Gholamreza Anbarjafari. 2019. Multimodal Database of Emotional Speech, Video and Gestures. In *Pattern Recognition and Information Forensics: ICPR 2018 International Workshops, CVAUI, IWCF, and MIPPSNA, Beijing, China, August 20-24, 2018, Revised Selected Papers*, Vol. 11188. Springer, 153.
- [11] Gilbert Badaro, Obeida El Jundi, Alaa Khaddaj, Alaa Maarouf, Raslan Kain, Hazem Hajj, and Wassim El-Hajj. 2018. Ema at semeval-2018 task 1: Emotion mining for Arabic. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. 236–244.

- [12] Yves Bestgen. 2019. CECL at SemEval-2019 Task 3: Using Surface Learning for Detecting Emotion in Textual Conversations. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 148–152.
- [13] Abdessalam Boucekif, Praveen Joshi, Latifa Boucekif, and Haithem Afli. 2019. EPITA-ADAPT at SemEval-2019 Task 3: Detecting emotions in textual conversations using deep learning models combination. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 215–219.
- [14] Jinkun Chen, Cong Liu, and Ming Li. 2017. Automatic emotional spoken language text corpus construction from written dialogs in fictions. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 319–324.
- [15] Xiyao Cheng, Ying Chen, Bixiao Cheng, Shoushan Li, and Guodong Zhou. 2017. An emotion cause corpus for chinese microblogs with multiple-user structures. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17, 1 (2017), 6.
- [16] Giovanni Costantini, Iacopo Iaderola, Andrea Paoloni, and Massimiliano Todisco. 2014. Emovo corpus: an italian emotional speech database. In *International Conference on Language Resources and Evaluation (LREC 2014)*. European Language Resources Association (ELRA), 3501–3504.
- [17] Kodati Dheeraj and Tene Ramakrishnu. 2021. Negative emotions detection on online mental-health related patients texts using the deep learning with MHA-BCNN model. *Expert Systems with Applications* (2021), 115265.
- [18] Hyo Jin Do and Ho-Jin Choi. 2015. Korean twitter emotion classification using automatically built emotion lexicons and fine-grained features. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*. 142–150.
- [19] Raissa Yapan Dougnon, Philippe Fournier-Viger, Jerry Chun-Wei Lin, and Roger Nkambou. 2015. Accurate Online Social Network User Profiling. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 264–270.
- [20] Raissa Yapan Dougnon, Philippe Fournier-Viger, Jerry Chun-Wei Lin, and Roger Nkambou. 2016. Inferring social network user profiles using a partial social graph. *Journal of intelligent information systems* 47, 2 (2016), 313–344.
- [21] Changde Du, Changying Du, Jinpeng Li, Wei-long Zheng, Bao-liang Lu, and Huiguang He. 2017. Semi-supervised Bayesian deep multi-modal emotion recognition. *arXiv preprint arXiv:1704.07548* (2017).
- [22] Samar Fathy, Nahla El-Haggag, and Mohamed H Haggag. 2017. A hybrid model for emotion detection from text. *International Journal of Information Retrieval Research (IJRR)* 7, 1 (2017), 32–48.
- [23] Zhiwei Guo, Keping Yu, Yu Li, Gautam Srivastava, and Jerry Chun-Wei Lin. 2021. Deep learning-embedded social internet of things for ambiguity-aware social recommendations. *IEEE Transactions on Network Science and Engineering* (2021).
- [24] Muhammad Hassan and Muhammad Shoaib. 2018. Opinion within opinion: segmentation approach for urdu sentiment analysis. *Int. Arab J. Inf. Technol.* 15, 1 (2018), 21–28.
- [25] Muhammad Humayoun, Rao Muhammad Adeel Nawab, Muhammad Uzair, Saba Aslam, and Omer Farzand. 2016. Urdu Summary Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. 796–800.
- [26] Younghee Jung, Kinam Park, Taemin Lee, Jeongmin Chae, and Soonyoung Jung. 2017. A corpus-based approach to classifying emotions using Korean linguistic features. *Cluster Computing* 20, 1 (2017), 583–595.
- [27] Sawit Kasuriya, Thanaruk Theeramunkong, Chai Wutiwintachai, and Piyawat Sukhummek. 2019. Developing a Thai emotional speech corpus from Lakorn (EMOLA). *Language Resources and Evaluation* 53 (march 2019), 1–39. <https://link.springer.com/article/10.1007/s10579-018-9428-9>
- [28] Dacher Keltner. 2004. Ekman, emotional expression, and the art of empirical epiphany. *Journal of Research in Personality* 38, 1 (2004), 37–44.
- [29] Hema Krishnan, M Sudheep Elayidom, and T Santhanakrishnan. 2017. Emotion Detection of Tweets using Naïve Bayes Classifier. *Emotion* (2017).
- [30] Marloes Kuijper, Mike van Lenthe, and Rik van Noord. 2018. Ug18 at semeval-2018 task 1: Generating additional training data for predicting emotion intensity in spanish. *arXiv preprint arXiv:1805.10824* (2018).
- [31] Jiyoung Lee and Yun Jung Choi. 2018. Understanding social viewing through discussion network and emotion: A focus on South Korean presidential debates. *Telematics and Informatics* 35, 5 (2018), 1382–1391.
- [32] Mirko Mazzoleni, Gabriele Maroni, and Fabio Previti. 2017. Unsupervised learning of fundamental emotional states via word embeddings. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–6.
- [33] Mohamed Meddeb, Hichem Karray, and Adel M Alimi. 2017. Building and analysing emotion corpus of the Arabic speech. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*. IEEE, 134–139.
- [34] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
- [35] Junko Minato, David B Bracewell, Fuji Ren, and Shingo Kuroiwa. 2006. Statistical analysis of a Japanese emotion corpus for natural language processing. In *International Conference on Intelligent Computing*. Springer, 924–929.
- [36] Junko Minato, David B Bracewell, Fuji Ren, and Shingo Kuroiwa. 2008. Japanese Emotion Corpus Analysis and its Use for Automatic Emotion Word Identification. *Engineering Letters* 16, 1 (2008).
- [37] Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence* 29, 3 (2013), 436–465.

- [38] Neelam Mukhtar and Mohammad Abid Khan. 2018. Urdu sentiment analysis using supervised machine learning approach. *International Journal of Pattern Recognition and Artificial Intelligence* 32, 02 (2018), 1851001.
- [39] Neelam Mukhtar, Mohammad Abid Khan, Nadia Chiragh, and Shah Nazir. 2018. Identification and handling of intensifiers for enhancing accuracy of Urdu sentiment analysis. *Expert Systems* 35, 6 (2018), e12317.
- [40] Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. 2014. Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing* 5, 2 (2014), 101–111.
- [41] Rutvija Pandya and Jayati Pandya. 2015. C5.0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications* 117, 16 (2015), 18–21.
- [42] Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*. Elsevier, 3–33.
- [43] Changqin Quan and Fuji Ren. 2009. Construction of a blog emotion corpus for Chinese emotional expression analysis. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 1446–1454.
- [44] Ebin Deni Raj, Gunasekaran Manogaran, Gautam Srivastava, and Yulei Wu. 2020. Information granulation-based community detection for social networks. *IEEE Transactions on Computational Social Systems* 8, 1 (2020), 122–133.
- [45] Zia Ul Rehman and Imran Sarwar Bajwa. 2016. Lexicon-based sentiment analysis for Urdu language. In *2016 sixth international conference on innovative computing technology (INTECH)*. IEEE, 497–501.
- [46] Xin Rong. 2014. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738* (2014).
- [47] Ali Saeed, Rao Muhammad Adeel Nawab, Mark Stevenson, and Paul Rayson. 2019. A Sense Annotated Corpus for All-Words Urdu Word Sense Disambiguation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18, 4 (2019), 40.
- [48] Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne, and Reda Alhajj. 2018. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining* 8, 1 (2018), 28.
- [49] Mary Jane C Samonte, Hector Irvin B Punzalan, Richard Julian Paul G Santiago, and Peter Joshua L Linchangco. 2017. Emotion detection in blog posts using keyword spotting and semantic analysis. In *Proceedings of the 3rd International Conference on Communication and Information Processing*. ACM, 6–13.
- [50] Kazuki Sato and Tomonobu Ozaki. 2019. Estimation of Emotion Type and Intensity in Japanese Tweets Using Multi-task Deep Learning. In *Workshops of the International Conference on Advanced Information Networking and Applications*. Springer, 314–323.
- [51] Abhinav Sethy and Bhuvana Ramabhadran. 2008. Bag-of-word normalized n-gram models. In *Ninth Annual Conference of the International Speech Communication Association*.
- [52] Neel Shah, Gautam Srivastava, David W Savage, and Vijay Mago. 2020. Assessing Canadians health activity and nutritional habits through social media. *Frontiers in public health* 7 (2020), 400.
- [53] Sergey Smetanin. 2019. EmoSense at SemEval-2019 Task 3: Bidirectional LSTM Network for Contextual Emotion Detection in Textual Conversations. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 210–214.
- [54] Mohamed Soltani, Hamed Zarzour, and Mohamed Chaouki Babahenini. 2018. Facial emotion detection in massive open online courses. In *World Conference on Information Systems and Technologies*. Springer, 277–286.
- [55] Afraz Zahra Syed, Muhammad Aslam, and Ana Maria Martinez-Enriquez. 2011. Sentiment analysis of urdu language: handling phrase-level negation. In *Mexican International Conference on Artificial Intelligence*. Springer, 382–393.
- [56] Afraz Z Syed, Muhammad Aslam, and Ana Maria Martinez-Enriquez. 2014. Associating targets with SentiUnits: a step forward in sentiment analysis of Urdu text. *Artificial intelligence review* 41, 4 (2014), 535–561.
- [57] Mansur Alp Tocoglu and Adil Alpkocak. 2014. Emotion extraction from turkish text. In *2014 European Network Intelligence Conference*. IEEE, 130–133.
- [58] Mansur Alp Tocoglu and Adil Alpkocak. 2018. TREMO: A dataset for emotion analysis in Turkish. *Journal of Information Science* 44, 6 (2018), 848–860.
- [59] Anthony J Viera, Joanne M Garrett, et al. 2005. Understanding interobserver agreement: the kappa statistic. *Fam med* 37, 5 (2005), 360–363.
- [60] Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus Creation and Emotion Prediction for Hindi-English Code-Mixed Social Media Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. 128–135.
- [61] Takashi Yamazaki and Minoru Nakayama. 2017. Extracting Acoustic Features of Japanese Speech to Classify Emotions.. In *FedCSIS Communication Papers*. 141–145.
- [62] Liang Yang and Hongfei Lin. 2012. Construction and application of Chinese emotional corpus. In *Workshop on Chinese Lexical Semantics*. Springer, 122–133.
- [63] Wisha Zehra, Abdul Rehman Javed, Zunera Jalil, Habib Ullah Khan, and Thippa Reddy Gadekallu. 2021. Cross corpus multi-lingual speech emotion recognition using ensemble learning. *Complex & Intelligent Systems* (2021), 1–10.
- [64] Dongyu Zhang, Hongfei Lin, Liang Yang, Shaowu Zhang, and Bo Xu. 2018. Construction of a Chinese Corpus for the Analysis of the Emotionality of Metaphorical Expressions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*

- (*Volume 2: Short Papers*). 144–150.
- [65] Dongyu Zhang, Hongfei Lin, Puqi Zheng, Liang Yang, and Shaowu Zhang. 2018. The Identification of the Emotionality of Metaphorical Expressions Based on a Manually Annotated Chinese Corpus. *IEEE Access* 6 (2018), 71241–71248.
- [66] Jialiang Zhao and Qi Gao. 2017. Annotation and Detection of Emotion in Text-based Dialogue Systems with CNN. *arXiv preprint arXiv:1710.00987* (2017).

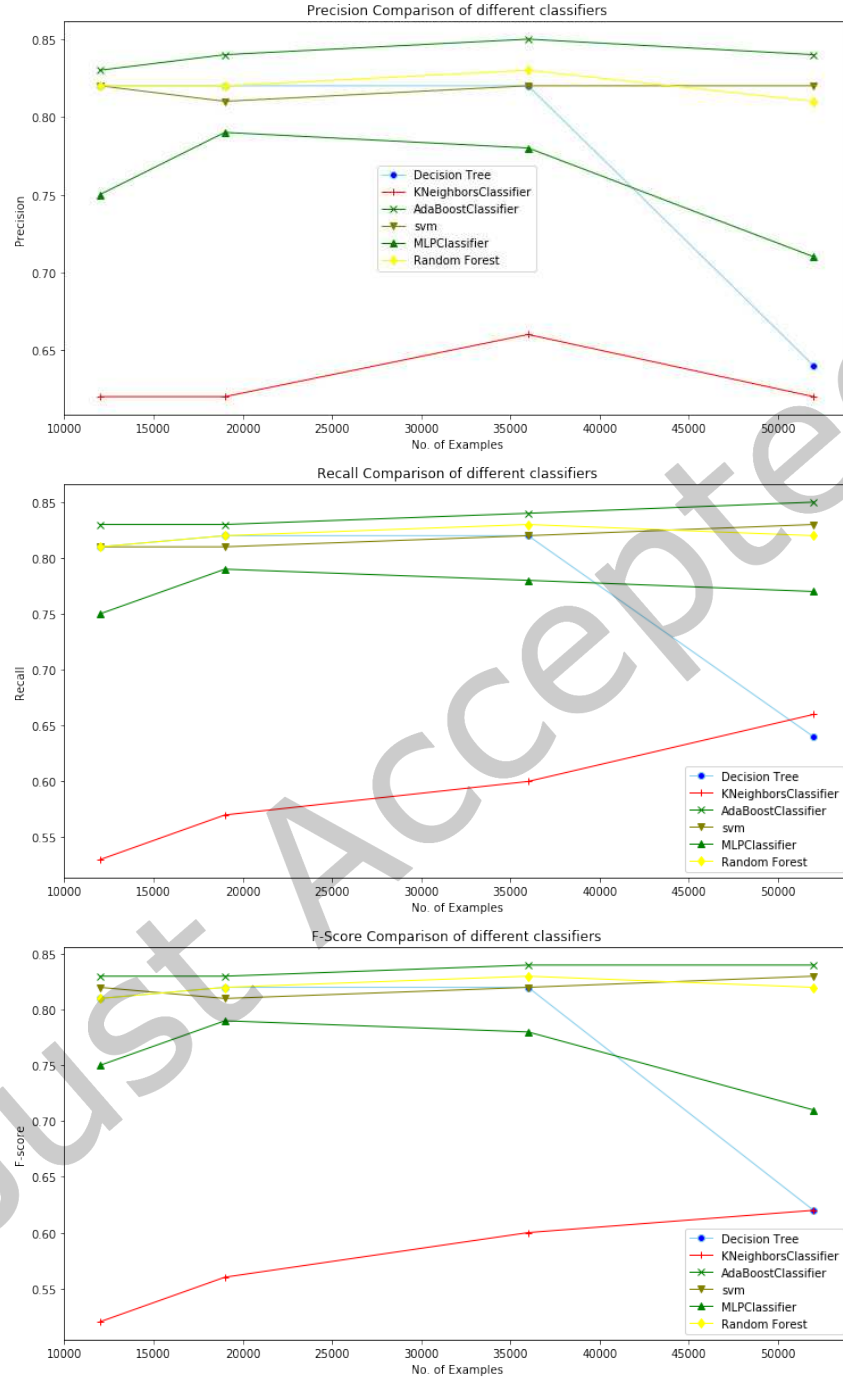


Fig. 11. Precision, Recall and F1 score comparison of all classification algorithms for UNED sentence based dataset using Count Vectorization

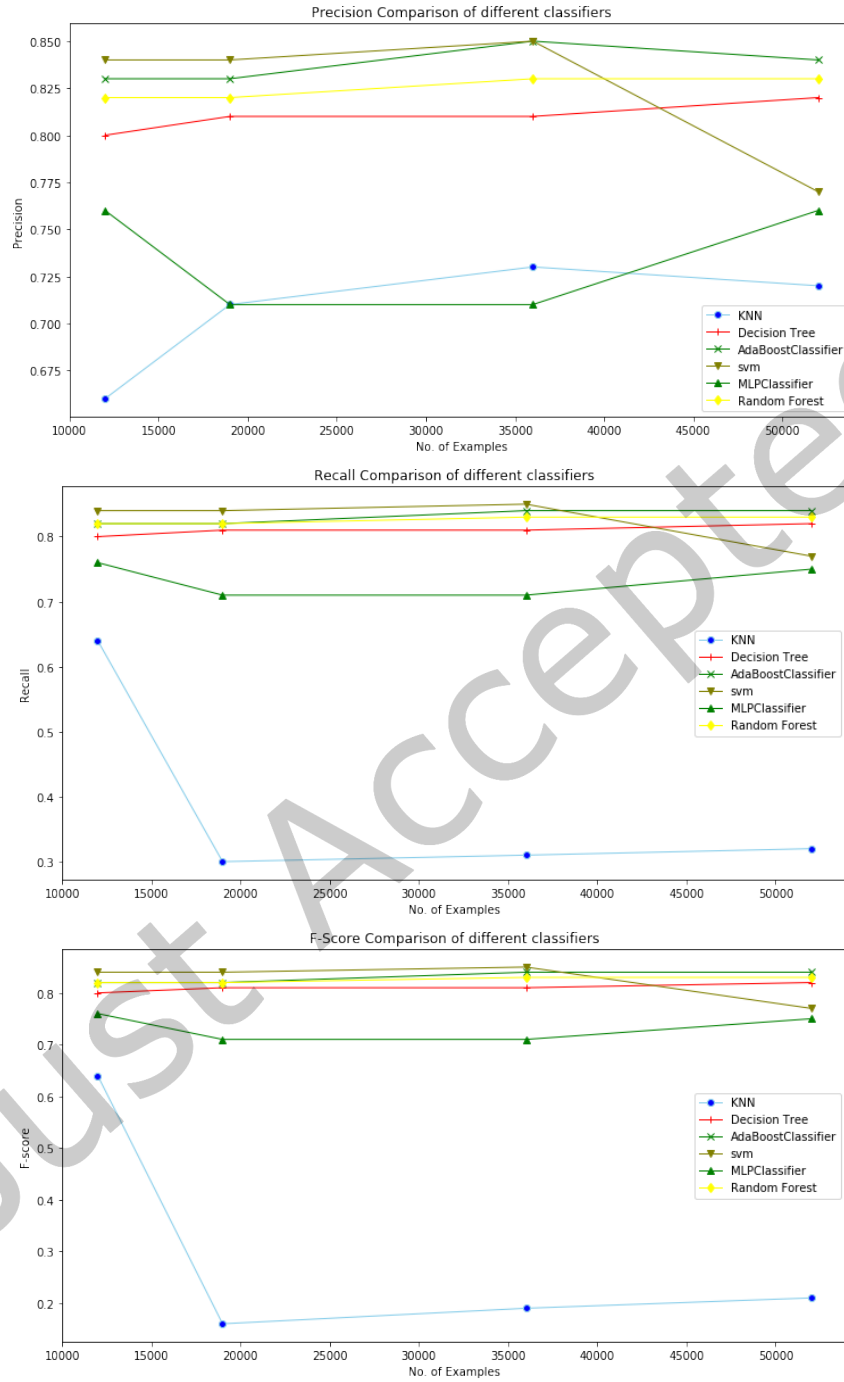


Fig. 12. Precision, Recall and F1 score Comparison of all classification algorithms for UNED sentence based dataset using Tf-idf

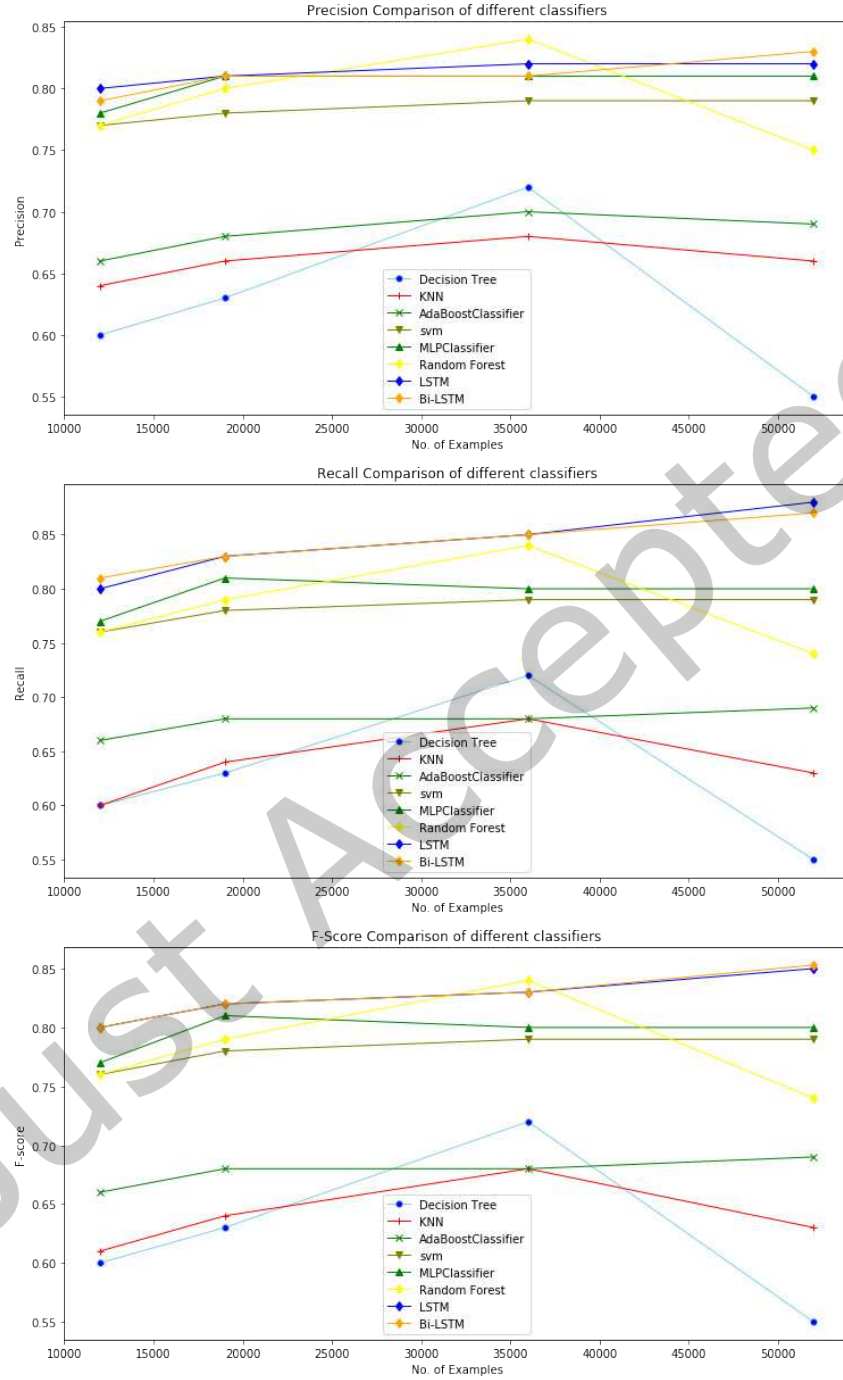


Fig. 13. Precision, Recall and F1 score comparison of all classification algorithms for UNED sentence based dataset using Word2vec

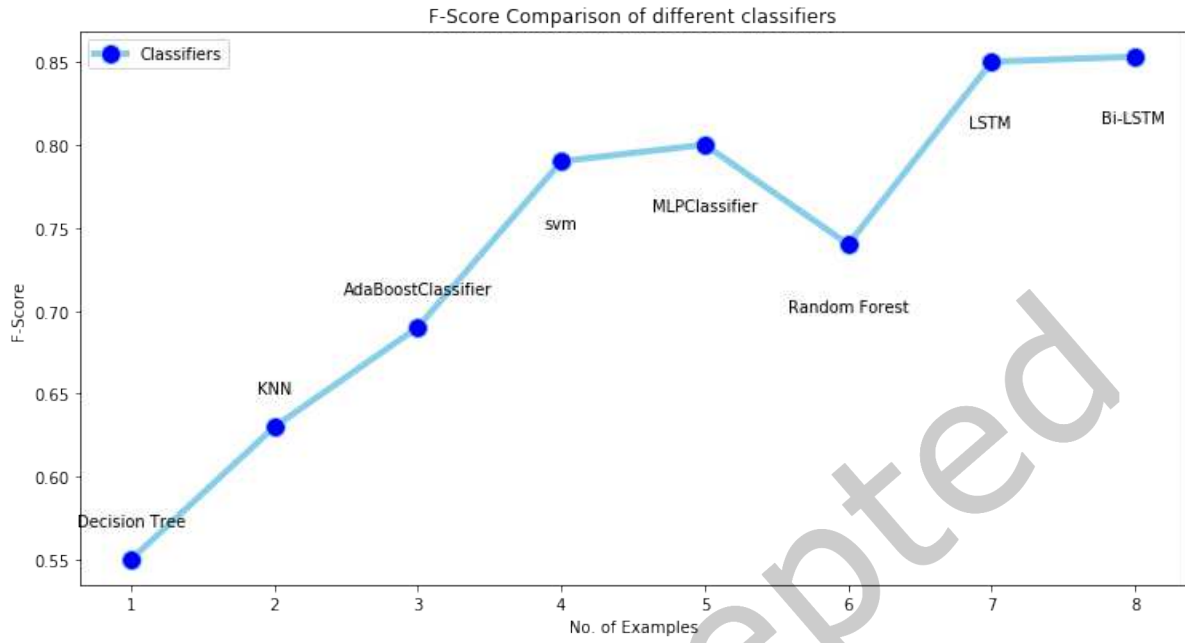


Fig. 14. Comparison with other classification algorithms for UNED sentence based dataset

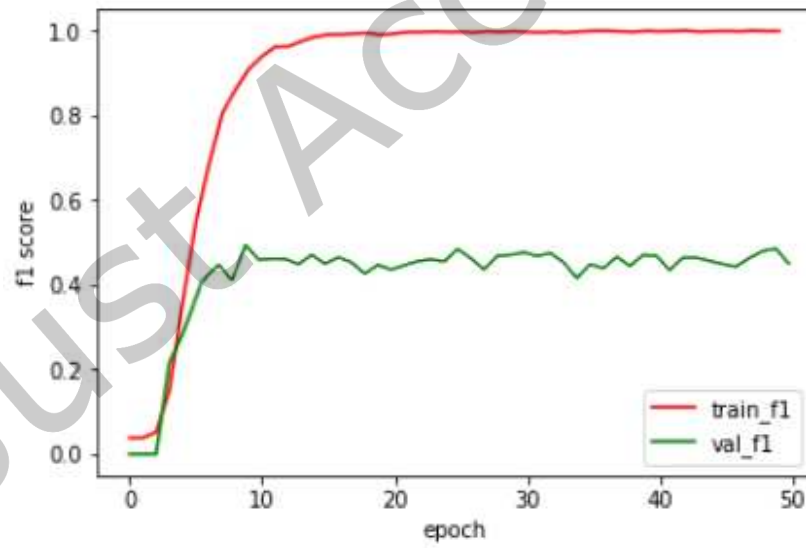


Fig. 15. Results with classification algorithm for UNED paragraph based dataset

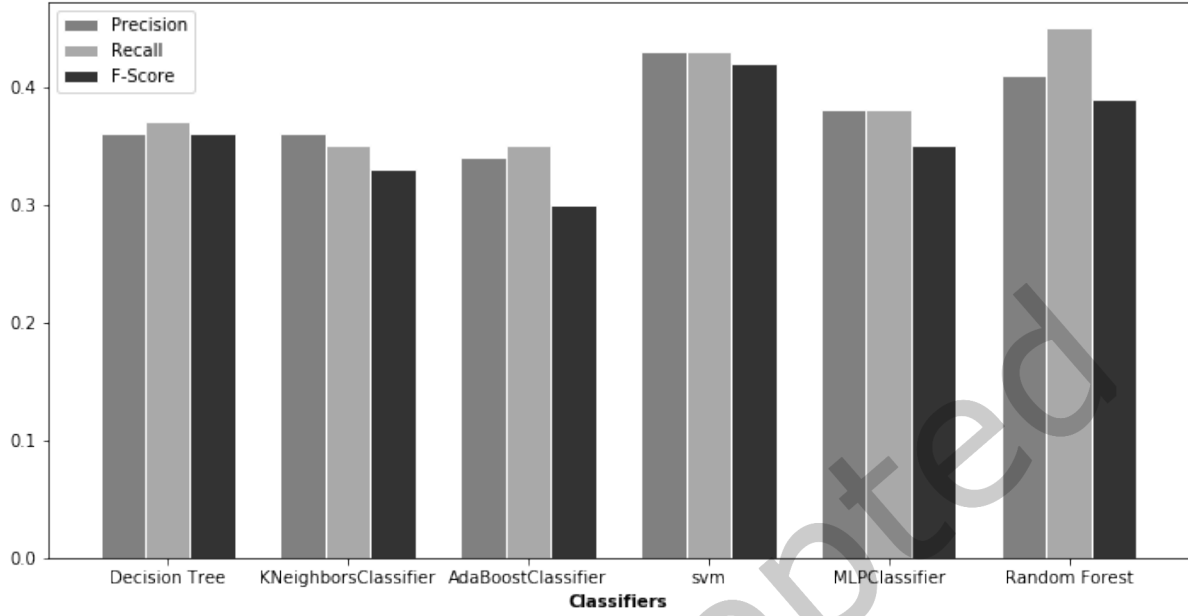


Fig. 16. Comparison with other classification algorithms using Count-Vectorization for UNED paragraph dataset

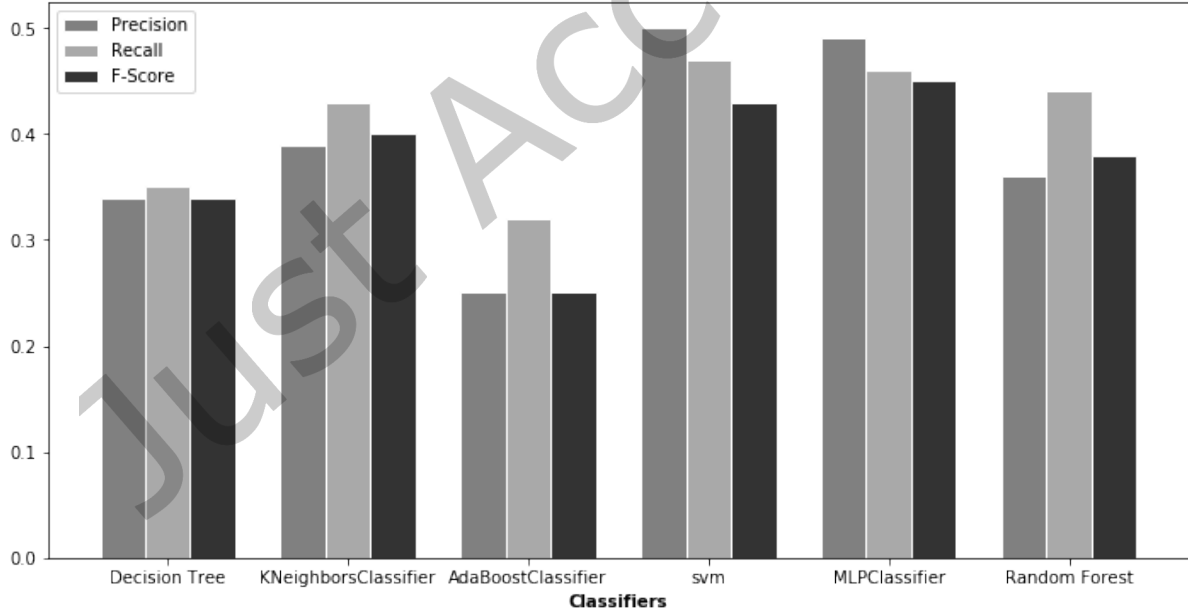


Fig. 17. Comparison with other classification algorithms using Tf-idf for UNED paragraph dataset

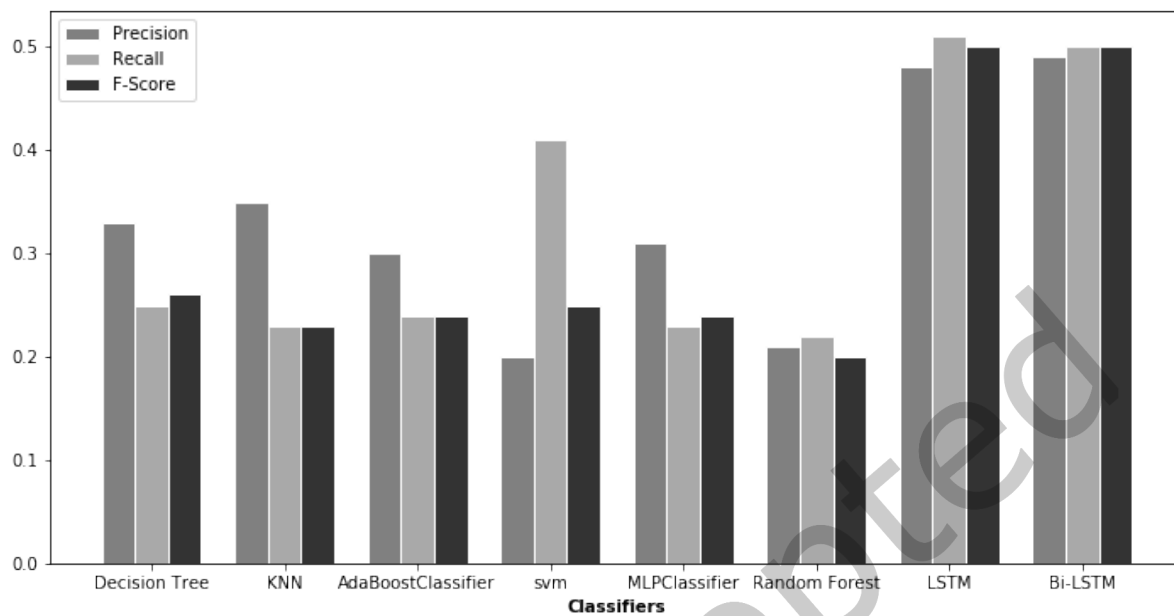


Fig. 18. Comparison with other classification algorithms using Word2vec for UNED paragraph dataset