Deliverable 3

Sofia Y. Ahmed
syahmed@uw.edu

Libbey Brown
egollhof@uw.edu

Rachel Hantz
hantz1rk@uw.edu

Elizabeth Okada
esokada@uw.edu

Abstract

This project report provides information about the system we have developed for our primary task, with improvements made upon our baseline system. Our primary task is multi-class emotion classification in English essays. Our adaptation task will be multi-class emotion classification in Urdu tweets.

1 Introduction

The primary task for our system this quarter is emotion classification in English essays. This task was Track 2 of the shared task for WASSA 2022 (Barriere et al., 2022) and required participants to develop a multi-class classification model which predicts an emotion tag given the text of an essay.

To develop a baseline, we experimented with several combinations of embedding strategies and classification models. None of our systems yielded high performance on the classification task; the macro F1-score for our highest-performing system was only 24.36. This system combined a bag of words embedding strategy with a decision tree classifier.

For this deliverable, we improved upon our baseline in several ways. We enhanced our preprocessing to handle negation. We implemented various methods to deal with the class imbalance in our data. We utilized boosting in conjunction with the classifiers used in our prior deliverable. Finally, we experimented with the use of a pretrained language model to act as a classifier.

Our best-performing system utilized a pretrained Roberta model fine-tuned on a version of our training data which was sampled to improve the balance of class labels. This system achieved a Macro F1 score of 45.88.

2 Task description

2.1 Primary Task

Our primary task is emotion classification for essay texts written in English; this task is Track 2 of the shared task for WASSA 2022 (Barriere et al., 2022). We develop a multi-class classification model which predicts an emotion tag from one of seven categories: Ekman's six basic emotions (joy, sadness, surprise, disgust, anger, or fear) and the neutral tag no-emotion.

The dataset was initially generated by Buechel et al. (2018) and consists of 2655 essays which were written by 403 participants in reaction to reading disturbing news articles. The dataset was downloaded from the WASSA (2022) website. Table 1 describes the original train, development, and test splits.

Train	Dev	Test	Total
1860	270	525	2655

Table 1: Original Dataset Split for Primary Task

Due to the inavailability of the original test data, we opted to obtain to retain the original development data as is, but then select 10% of test data from the original training data. Table 2 describes the new train, development, and test splits. Here, there are 2130 total essays with a 77.3/12.7/10 train/dev/test split.

Train	Dev	Test	Total
1647	270	213	2130

Table 2: New Dataset Split for Primary Task

Participants rated their level of empathy and distress after reading the article and then described their thoughts and feelings in writing. Emotional tags were added to the essays as part of the WASSA 2021 shared task (Tafreshi et al., 2021). Emotion tags were added through prediction models (the specific models used were a gated RNN and a fine-tuned RoBERTA model) and the tags were manually verified by annotators from Amazon Mechanical Turk. Because these essays were generated in response to disturbing news articles, the distribution of emotion tags is not balanced; sadness is overwhelmingly the largest tag, followed by anger. See Table 3 for the distribution of the emotion tags.

	Train	Dev	Test	Total
joy	72	14	10	96
sadness	570	98	77	745
disgust	131	12	18	161
fear	173	31	21	225
anger	312	76	37	425
surprise	145	14	19	178
no-emo	244	25	31	300

Table 3: Emotion Distribution for Primary Task

For evaluation, we use the evaluation script that was provided by the task organizers on the website. We have simplified the evaluation script by removing code for the regression task tracks, which are not part of our project, and also by allowing for arbitrary paths and filenames. We have not modified the function that calculates the scores. Our system outputs a .tsv file with a single column of emotion predictions in string format. The evaluation file compares this output file with the gold standard file and performance is calculated through several measures: Macro F1-Score, Macro Recall, Macro Precision, Micro F1-Score, Micro Recall, Micro Precision, and Accuracy.

2.2 Adaptation Task

For our adaptation task, we plan to classify emotion for tweets in Urdu; this was Task A for the shared task at FIRE 2022 (Butt et al., 2023). As with our primary task, we plan to develop a multi-class classification model which predicts an emotion tag from one of seven categories: Ekman's six basic emotions (joy[1], sadness, surprise, disgust, anger, or fear) and the neutral tag no-emotion. The adaptation differs from our primary task in both language and genre.

The Urdu dataset consists of 9750 annotated tweets. The datatset was downloaded from the FIRE (2022) shared task website. Table 4 lists the split between training and test data. To mirror the primary task, we may create a development set extracted from the training data prior to model development.

Train	Test	Total
7800	1950	9750

Table 4: Dataset Split for Adaptation

The tweets are labeled with emotional tags among Ekman's six basic emotions and the neutral no-emotion tag. Construction of the dataset is detailed in Ashraf et al. (2022)[2]. The authors' goal was to construct a balanced dataset, so Twitter hashtags were used to identify tweets representative of a particular emotion, and then annotators classified the tweets with one or more emotion tags. Note that this dataset differs from the dataset used in our primary task in that each tweet can be tagged with more than one emotion (i.e. suggestive of a multi-label classification task). We will utilize only the tweets which have a single emotion tagged for ease of adaptation. Table 5 shows the distribution of emotion tags for the full Urdu data, prior to selecting only tweets with one emotion tag[3].

	Train	Test	Total
joy	1046	261	1307
sadness	1550	388	1938
disgust	761	190	951
fear	609	152	761
anger	811	203	1014
surprise	1550	388	1938
no-emo	3014	753	3767

Table 5: Emotion Distribution for Adaptation

[1]The original Urdu dataset uses 'happiness' in place of 'joy'. We map the label 'happiness' to 'joy.'

[2]The FIRE 2022 shared task references Ashraf et al. (2022) as a source for their dataset, but the present dataset appears to be a slightly larger sample than the one depicted in Ashraf et al. (2022). Thus, we display counts from the present dataset, not those described by Ashraf et al. (2022).

[3]285 train and 61 test tweets had no emotion tag.

Table 6 lists the split between training and test data after selecting only tweets with one emotion tag.

Train	Test	Total
5463	1389	6852

Table 6: Dataset Split for Adaptation: Only One Emotion Tag per Tweet

Table 7 shows the distribution of Urdu emotion tags after selecting only tweets with one emotion tag.

	Train	Test	Total
joy	744	185	929
sadness	911	230	1141
disgust	20	10	30
fear	159	41	200
anger	89	35	124
surprise	526	135	661
no-emo	3014	753	3767

Table 7: Emotion Distribution for Adaptation: Only One Emotion Tag per Tweet

We will use the same evaluation script as for the primary task.

3 System Overview

For this deliverable, we experimented with two paths: (A) fine-tuning a pretrained Roberta model, and (B) applying improvements to our scikit-learn classifier systems (see Figure 1).

We fine-tuned the pre-trained Roberta model with both with our original (imbalanced) training data and with a class-balanced version of our training data.

The scikit-learn classifier path included a preprocessing step, optional negation handling, several types of embeddings, optional class balancing, and a decision tree or SVM classifier with optional boosting.

Evaluated on the official task metric of Macro F1, the best-performing system was a pretrained Roberta model combined with a class-balancing strategy. The best-performing traditional system was a decision tree classifier run on our emotion-enhanced bag of words vectors and combined with a class-balancing strategy.

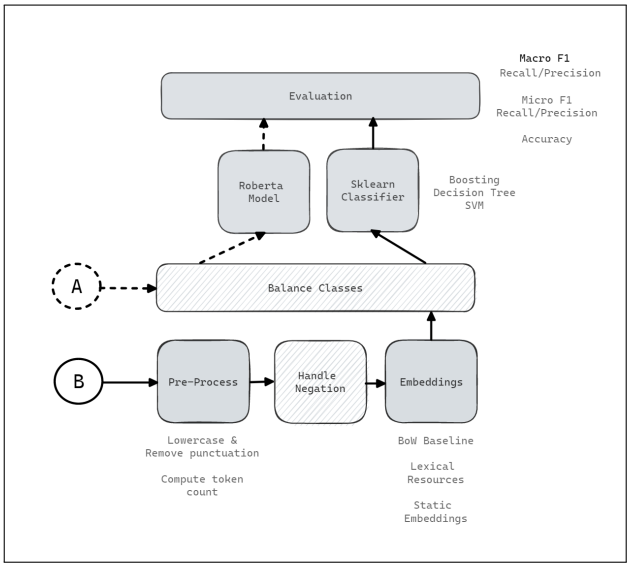

Figure 1: System Overview

4 Approach

4.1 Text Pre-processing

Text pre-processing included tokenization, lowercasing, and removal of punctuation.

We processed negated words using a technique adapted from the method laid out in (Babanejad et al., 2020), in which negation words were removed and the word immediately following the negation was replaced with its antomym. We used the Python package spaCy[4] to create dependency parses of each essay. Tokens with the dependency "negation" were identified and removed; tokens that were the head of the negation tokens were replaced with their antonyms. Antonyms were identified using the Python package spacy-wordnet [5].

4.2 Embeddings

We used various methods to create feature vectors for the essays, including a bag of words baseline, lexical resources, and static embeddings. We describe these methods in the following sections.

4.2.1 Bag of Words

We utilized bag of words to use as a baseline vector representation for our system. We instantiated a vocabulary for the bag of words by sorting the set of tokens in all training instances combined in order to assign each token an index. Then, to vectorize an essay instance, we assigned the frequency of each token in that essay to that token's vocabulary index

[4]https://spacy.io/
[5]https://pypi.org/project/spacy-wordnet/

in the vector. If an essay instance contained a token not found in the vocabulary, it was interpreted as an "unknown" token, and assigned the index for "unknown" for smoothing.

4.2.2 Lexical Resources

We utilized the NRC Word-Emotion Association Lexicon (also known as EmoLex) as a resource to create two additional vector representations that go beyond a baseline bag of words: emotion only vectors and emotion enhanced bag of words vectors (Mohammad and Turney, 2010, 2013). For a set of 14,182 unigrams, EmoLex attributes a binary 0 (not associated) or 1 (associated) for eight emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) and two sentiments (negative and positive). This lexicon was initially and further developed using crowdsourcing.

Emotion only vectors contained 6 features, one for each of Ekman's 6 emotions. For each token in an essay instance that was also present in the emotion lexicon, we summed the binary feature vector returned for each word from EmoLex. If a token was present more than once in the essay instance, the corresponding binary feature vector was summed that many times. A vector containing all zeros, indicates that no tokens in the essay instance had any association with any of Ekman's 6 emotions according to EmoLex. Emotion enhanced bag of words vectors were the concatenation of the bag of words and emotion only vectors.

4.2.3 Static Embeddings

Word2Vec is a model architecture which computes vector representations of words by reconstructing their linguistic context (Mikolov et al., 2013a). A large corpus of words is taken as an input and the model produces a vector space, generally of hundreds of dimensions, in which each word has a corresponding vector. For this system, we utilized pretrained Word2Vec embeddings which were generated by training the model on a corpus of one billion words that came from a dataset of Google News articles (Mikolov et al., 2013b). The pretrained vector representations were obtained through Gensim's data repository (RaRe-Technologies).

Each essay was represented with a centroid vector. First, the vector representation of each word was found; if there was no vector representation of a word it was omitted. The centroid of an essay was the average of the vector representations of all words. This method - calculating the centroid from the average of all words in the essay - was used in (Gennaro and Ash, 2022).

4.3 Class Imbalance

As seen in Table 3, the distribution of emotions is highly skewed amongst the dataset of essays. 35% of essays are tagged as *sadness*, while only 4.5% of essays are tagged *joy*. To handle class imbalance, we implemented and tested three class balancing strategies: random over/under sampling, SMOTE, and SMOTE with removal of Tomek links.

Random over/under sampling found the average class size for the training data, undersampling classes above the average class size and oversampling classes below the average class size so that all classes had the average class size. SMOTE (Synthetic Minority Over-sampling Technique) calculated intermediate vectors between pairs of data points of a minority class and added these synthetic minority class vectors to the training data (Chawla et al., 2002). Tomek links are pairs of opposite class instances that are each other's nearest neighbor. SMOTE with removal of Tomek links first applied the SMOTE algorithm and then undersampled the training dataset by removing any Tomek links (Batista et al., 2003). Both SMOTE strategies inserted synthetic vector instances up to the value of the majority class size. The python library imbalanced-learn was leveraged for all three methods[6].

We selected random over/under sampling as our class balancing strategy as it had the highest performance on our data. We describe our process for selecting this method in Section 5, displaying results of various trials in Table 9.

4.4 Classifiers

Two classical machine learning models were used in our experimentation: Support Vector Machines (SVM) and Decision Trees. SVMs are supervised learning methods that can be used for classification, regression and detecting outliers (Pedregosa et al., 2011). In this case, we used SVM for classification. We first attempted to classify using SVM because it is commonly used in classification problems. Next, we implemented Decision Trees in order to experiment with a different Machine Learning model and see if we could improve the classification predictions from the experiments that used SVM. Deci-

[6]https://imbalanced-learn.org/stable/index.html

sion Trees are also a supervised learning method used for classification and regression, and again, in our case, were used for classification. The Python package scikit-learn was used to implement both of these Machine Learning models. [7]

As an attempted improvement for this deliverable, we wanted to try an ensemble method, which is a technique that uses multiple models and combines them into one for enhanced results. The ensemble method we chose to use was boosting. Boosting can be used in tandem with classical machine learning models, such as Decision Trees, to enhance the results. Therefore, we implemented boosting in tandem with both SVMs and Decision Trees, as those were the machine learning models that we used in our baseline system. Once again, scikit-learn was used for this.

In addition to the classical machine learning algorithms described above, we began using a pretrained language model for this deliverable. We selected the *distilroberta-base* model available from the Hugging Face hub for its relatively small size and ease of use. The transformers library was used to fine-tune this model for a sequence classification task and perform inference. We adopted the default hyperparameters recommended on the Hugging Face website. [8]

5 Results

Without handling for class imbalance or negation, the un-boosted decision tree and SVM classifiers used in our initial system yielded low performance on their best vector configurations: the bag of words baseline and emotion enhanced bag of words vectors respectively (Table 8). For the best decision tree system without class balancing or negation, on average, only 24.3% of predicted emotions are true positives (Macro precision) and only 25.9% of emotions that *should* be predicted are detected (Macro recall). The Macro F1 was 24.4. SVM predictions overly favored the majority class when not handling for class imbalance, greatly skewing the interpretability of Macro precision and producing a very low best Macro F1 of 15.95. To narrow our experimentation space, we did not continue evaluating on SVM.

Table 9 displays the Macro F1 scores that resulted from trials of each class balancing strategy on our six vector types and the decision tree classifier[9]. Random over/under sampling performed best in 3 out of 6 cases–highest with emotion enhanced bag of words vectors using decision trees–and was selected to help improve our system. In the best balancing configuration, on average, 31.82% of predicted emotions are true positives (macro precision), and 32.42% of emotions that *should* be predicted are detected (macro recall). The Macro F1 score of 29.9 improves on our previous best initial system's Macro F1 of 24.4.

	BoW	EmoBoW	W2VPT
initial sys			
SVM	13.7	15.9	7.6
DT	24.4	21.6	20.3
DTb	25.5	22.7	14.4
N+DT	28.1	24.1	14.9
N+DTb	21.9	25.2	13.7
B+DT	26.8	**29.9**	14.6
B+DTb	24.6	22.2	13.9
N+B+DT	26.4	25.9	13.1
N+B+DTb	22.3	24.6	13.2
Roberta			
Roberta	**35.7**		
B+Roberta	**45.9**		

Table 8: **Ablation Study:** Macro F1 scores for initial systems and various improvement component configurations on the development set: *N*: Negation, *B*: Balancing, *DTb*: Decision Tree Boosting. **Bold** text denotes the three highest performing configurations. An underline denotes the highest performing configuration overall. Bag of Words embeddings are used in baseline systems.

Vector	Over/Under	SMOTE	S.Tomek
BoW	**26.81**	25.24	23.9
Emo	14.64	**16.36**	15.9
EmoBoW	**29.9**	23.06	24.67
W2VPT	**14.56**	14.2	14.45

Table 9: Macro F1 scores for class balancing strategies on the development set using *decision tree* classifier. **Bold** text denotes the highest performing balancing strategy for each vector type. An underline denotes the highest performing configuration overall.

Table 8 also displays an ablation study showing the individual impact of each improvement compo-

[7] https://scikit-learn.org/stable/index.html

[8]: https://huggingface.co/docs/transformers/tasks/sequence_classification

[9] Almost all trials on the SVM classifier performed worse than using the decision tree classifier.

nent (Negation, Balancing, Decision Tree Boosting, Roberta) for our system. Overall, training a Roberta model on our own balanced data produces the best system, with a Macro F1 of 45.88. 47.35% of predicted emotions are true positives (Macro precision) and 53.6% of emotions that *should* be predicted are detected (Macro recall). The system is 51.48% accurate (Micro precision/recall/F1). Performance for each emotion class is displayed in Table 10. We see the highest performance for our majority class, *sadness*.

	precision	recall	f1-score	n
anger	0.56	0.25	0.35	76
disgust	0.15	0.75	0.24	12
fear	0.54	0.45	0.49	31
joy	0.47	0.64	0.55	14
neutral	0.39	0.28	0.33	25
sadness	0.82	0.73	0.77	98
surprise	0.39	0.64	0.49	14

Table 10: Per class metrics for highest performing system: Roberta model trained on balanced data.

6 Discussion

Our best-performing system is the Roberta model trained on balanced data, with a Macro F1 score of 45.88. This is a significant improvement upon our baseline system, which had a Macro F1 of only 24.4. This is likely due to the large size of the pre-trained Roberta model compared to a classical machine learning model trained from scratch on our relatively small training data, as well as the Roberta model's use of contextualized embeddings. The class balancing strategy also played a major role: without balancing, the Roberta model achieved a Macro F1 of only 35.66 (compared to 45.88 with balancing).

6.1 Qualitative Analysis

A qualitative analysis of the predictions made by our system reveals some patterns to help us understand the performance of our system.

6.1.1 Correct predictions

Our system consistently makes the correct predictions for majority class labels in which the emotion is stated clearly in the text of the essay, as in the following examples:

- *I felt pretty neutral when i read the article.* (neutral)
- *My feelings after reading the article is that it made me feel a bit sad.* (sadness)
- *After reading the article, i was definitely a bit worried and alarmed.* (fear)

6.1.2 Incorrect predictions

Incorrect predictions made by our system tend to fall in four categories: multiple emotions described in the text of the essay, nuanced expression of emotion, minority class emotions, and poor annotation of the data.

1. Many of the articles appear to have inspired multiple emotions for the participants. For example, some essays contained expressions of both fear and sadness, fear and surprise, or sadness and joy. Our system would often identify an emotion that, while expressed in the text, was not the primary emotion of the essay. In the following example, the gold annotation is surprise, but the author also expresses fear.

   > *"I just read an artivcle about a police shooting. It was different than I expected. Two police officers where shot to death while sitting in their patrol cars. I have gotten so used to reading about them shooting and killing people I was surprised to read about them getting the same treatment. Everyday I am reading something or seeing a video of police shooting people. I am scared to death of getting pulled over. All the police in my area are rookies and do not know me from when I was a ""bad kid"" lol. I would be worried to death getting pulled over by the police seeing how trigger happy they have been lately."*

2. As stated earlier, our system did very well classifying essays which stated clearly "I felt ..." or "This article made me feel..". However, essays which did not contain specific definitions of emotion were more challenging for the system to classify. The following essay, which is labeled with the emotion "disgust", does not provide such a clear definition of the author's emotion.

   > *Can you believe the Canadian government did something some terrible. I would never expect Canada to have*

something so bad on there record, especially given there reputation. Those families deserve justice and any living relatives deserve reparations from the government. This will at least be some justice.

3. Our system had challenges identifying the least frequent emotion, 'joy', even when this emotion was clearly expressed in the text as in the following essay:

   I am glad that they found the missing hiker and it is heartwarming that everything turned out for the best and he was found alive. It was truly amazing that he survived all that time without food or shelter. It goes to show that you should go looking for somebody missing no matter how long it takes because they might still be out there. It is an amazing story that will inspire and bring hope to the world.

4. Finally, there were essays which (in our opinion) were mislabeled by the annotators. While our system assigned what we believe to be the correct label, it did not match with the gold label and so contributed to our error rate. The following essay has the gold label of "sadness", but our team feels it is more accurately represented by the label of "neutral".

   My feelings towards this article is neutral as well. It did not really make me feel anything. Rather yet, it made me feel a bit bored and i kind of dozed off for a bit. I felt like it wasn't anything that would impact me or anything i would think twice about. It made me feel mostly relaxed. I didn't have nay sort of positive or negative emotions towards the article.

7 Ethical Considerations and Limitations

7.1 Ethical Considerations

There are some ethical concerns regarding the collection of this dataset. These essays were written in response to disturbing news articles; as noted earlier, the most prevalent emotion in this dataset is sadness. It is likely that many of the participants had unpleasant reactions to the news articles and may have had emotional harm from participating in the creation of this dataset.

There is inherent risk in classifying human emotions based on text, especially if some significant decision will be made about the author based on the predicted emotion. One example is the detection of crisis in student standardized test essays (Burkhardt et al., 2021). There is significant doubt that machine learning algorithms are able to detect internal emotional states, and such systems also demonstrate racial and gender bias (Boyd and Andalibi, 2023). Added to this is the fact that our task only allows for one emotion to be detected per passage, which leaves little room for nuance in interpreting the results.

7.2 Limitations

While our system performance increased notably for this deliverable, we also left several avenues of experimentation unexplored. We were unable to perform data augmentation to compensate for the small size of our training data, either by using synonym replacement to augment our data for the minority classes or by finding data from another source. Additionally, while we did attempt one ensemble method (boosting), we did not use ensemble methods combining approaches with different models. Finally, we did not tune the recommended hyper-parameters for the Roberta model. Some combination of these techniques may have yielded even more promising results, especially for the essays which were incorrectly classified due to the minority class emotion or the more nuanced expression of emotions.

8 Conclusion

In the development of the initial system, we experimented with different vector representations and classifiers. None of our systems yielded high performance on the classification task, and none of the systems performed better than the baseline bag of words model using the decision tree classifier. Overall, the SVM classifier overwhelmingly predicted the majority class, while the decision tree classifier was generally inaccurate.

In this deliverable, we were able to successfully improve upon our initial system's performance from the previous deliverable. We implemented several improvements to address some of the limitations of our baseline system. These improvements included handling class imbalance, adjusting preprocessing to handle negation, and leveraging new techniques for classification such as boosting as

well as the pre-trained language model RoBERTa. Due to these improvements, our best-performing system is the RoBERTa model that has been trained on our improved balanced dataset. This system yielded a Macro-F1 score that was significantly higher than the Macro-F1 score of our best system from the last deliverable. Thus, through various experiments and tools, we were able to improve upon our initial system in this deliverable. Next, we look forward to adapting our system to work on an Urdu dataset.

References

Noman Ashraf, Lal Khan, Sabur Butt, Hsien-Tsung Chang, Grigori Sidorov, and Alexander Gelbukh. 2022. Multi-label emotion classification of urdu tweets. *PeerJ Computer Science*, 8:e896.

Nastaran Babanejad, Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. A comprehensive analysis of preprocessing for word representation learning in affective tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5799–5810, Online. Association for Computational Linguistics.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. WASSA 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 214–227, Dublin, Ireland. Association for Computational Linguistics.

Gustavo Batista, Ana Bazzan, and Maria-Carolina Monard. 2003. Balancing training data for automated annotation of keywords: a case study. pages 10–18.

Karen L. Boyd and Nazanin Andalibi. 2023. Automated Emotion Recognition in the Workplace: How Proposed Technologies Reveal Potential Futures of Work. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW1):95:1–95:37.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4758–4765, Brussels, Belgium. Association for Computational Linguistics.

Amy Burkhardt, Susan Lottridge, and Sherri Woolf. 2021. A Rubric for the Detection of Students in Crisis. *Educational Measurement: Issues and Practice*, 40(2):72–80.

Sabur Butt, Maaz Amjad, Fazlourrahman Balouchzahi, Noman Ashraf, Rajesh Sharma, Grigori Sidorov, and Alexander Gelbukh. 2023. Emothreat@fire2022: Shared track on emotions and threat detection in urdu. New York, NY, USA. Association for Computing Machinery.

Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357.

FIRE. 2022. Emothreat at fire 2022.

Gloria Gennaro and Elliott Ash. 2022. Emotion and reason in political language. *The Economic Journal*, 132(643):1037–1059.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA. Association for Computational Linguistics.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

RaRe-Technologies. gensim-data.

Shabnam Tafreshi, Orphee De Clercq, Valentin Barriere, Sven Buechel, João Sedoc, and Alexandra Balahur. 2021. WASSA 2021 shared task: Predicting empathy and emotion in reaction to news stories. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–104, Online. Association for Computational Linguistics.

WASSA. 2022. Wassa 2022 shared task.