Amrita_CEN at SemEval-2022 Task 6: A Machine Learning Approach for Detecting Intended Sarcasm using Oversampling

Aparna K Ajayan, Krishna Mohanan, Anugraha S, Premjith B, and Soman K.P
Centre for Computation Engineering and Networking (CEN)
Amrita School of Engineering, Amrita Vishwa Vidyapeetham, India
b_premjith@cb.amrita.edu

Abstract

This paper describes the submission of the team Amrita_CEN to the shared task on iSarcasm Eval: Intended Sarcasm Detection in English and Arabic at SemEval 2022. The sarcasm detection task was formulated as a classification problem and modelled using machine learning classifiers. We used K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naïve Bayes, Logistic Regression, Decision Tree, and the Random Forest ensemble method. In addition, the class imbalance problem in the dataset was addressed using a feature engineering technique. We submitted the predictions by SVM, Logistic Regression and Random Forest ensemble based on the performance during training.

1 Introduction

Sarcasm is an ironic form showing a disparity between the actual and intended meaning of the text affecting the decision-making process. These are reflected in our day-to-day communication with each other happening in social media forums. Twitter exhibits rich sarcasm phenomena, thereby encouraging automatic sarcasm detection methods and removing such tweet data. Due to the sociocultural aspects of sarcastic communication, the majority of the sarcasm detection work has been focused only on the English language (Oprea and Magdy, 2020b), and only a limited amount of work was done in other languages such as Arabic (El Mahdaouy et al., 2021).

Identification of sarcastic comments from social media contexts is essential since the author and the receiver are at various places. Therefore, exchanging conversations may sometimes lead to a negative meaning of the text that even the author has not meant to convey. Moreover, the data stream for sarcasm does not exhibit any static structure like specific tags in the form of #sarcasm, and #irony (Ptáček et al., 2014) (Khodak et al., 2018). This event can lead to noisy labels due to several reasons, as outlined by (Oprea and Magdy, 2020b). Other works reported on the topic mainly depend on manual labelling, provided with manually annotated sarcasm labels. In (Oprea and Magdy, 2020b) the authors pointed out that manual labelling represents the author annotation in contrast with the intention of the authors.

The sarcasm prediction on Twitter that influences Machine Intelligence is a challenging task (Khare et al., 2022). It can be achieved with the help of the Natural Language Processing (NLP) approach, and many recent works on automatic sarcasm detection have focused on Twitter data as it primarily requires an understanding of the human expressions, language, and emotions expressed via textual or non-textual content (Kumar et al., 2021). Therefore, the goal of the SemEval shared task is to facilitate the development of machine learning models that can detect sarcasm from tweets. The shared task consists of two subtasks:

- Subtask A: For a given text, determine whether it is sarcastic or non-sarcastic.
- Subtask B (English only): A binary multi-label classification task for a given a text, determine which ironic speech category it belongs.

In this paper, we describe the machine learning models designed for solving the problems given in iSarcasm shared tasks (Abu Farha et al., 2022). The performance of the models was evaluated using the F1-score. The models submitted achieved the following scores: 0.4966 in English, 0.6127 in Arabic and 0.0567 F1-score in subtasks A and B, respectively.

2 Literature Review

The majority of the published works developed for the text sarcasm detection used datasets that were annotated using a weak supervision method, where the texts were regarded as sarcastic only if they met

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 834 - 839

preset criteria, including specific tags like sarcasm and irony (Oprea and Magdy, 2020a) (Ptáček et al., 2014) (Khodak et al., 2018). In (Oprea and Magdy, 2020b), S.V Opera and W Magdy reported that labelling using a weak supervision method could lead to noisy labels. Other works on this topic were based on manual labelling, where the human annotators are given the role of labelling the texts (Filatova, 2012) (Riloff et al., 2013) (Abercrombie and Hovy, 2016). The disadvantage of such a labelling procedure is that it represents the perception of the annotator, which may differ from the author's intention (Oprea and Magdy, 2020b).
In addition to the above-mentioned method, a significant majority of works on sarcasm detection were centered exclusively on the English language (Oprea and Magdy, 2019) (Campbell and Katz, 2012) (Riloff et al., 2013) (Joshi et al., 2016) (Amir et al., 2016) (Rajadesingan et al., 2015) (Bamman and Smith, 2015). It is because of its sociocultural aspects on sarcastic communication (Oprea and Magdy, 2020b), leading to the uncertainty that, the models trained on English could generalize to other languages. All the reported works on sarcasm detection in other languages such as Arabic (Karoui et al., 2017) (Ghanem et al., 2019) (Abbes et al., 2020) (Farha and Magdy, 2020) were relied on the afore-mentioned labelling techniques.

3 Dataset and Task Description

The dataset comprises tweets in English and Arabic. There are two subtasks in English and one in Arabic. Tweets in the English dataset were categorized into two: Sarcastic and Non-sarcastic. It contains 3,467 instances of tweets and ten columns containing the attributes (id, tweet, sarcastic, rephrase, sarcasm, irony, satire, understatement, overstatement, rhetorical question). The objective of task-1 is to determine whether a given text is sarcastic or not. Task-2 is a multi-label classification that aims to classify a tweet into different ironic speech categories, such as Sarcasm, Irony, Satire, Understatement, Overstatement, and Rhetorical questions. The shared task-1 in Arabic focused on categorizing a tweet into sarcastic or non-sarcastic, similar to task-1 in English. The Arabic dataset contains 2,601 instances of tweets and five attributes (id, tweet, sarcastic, rephrase, dialect). Table 1 describes the datasets used for task-1 and task-2 in English and task-1 in Arabic.

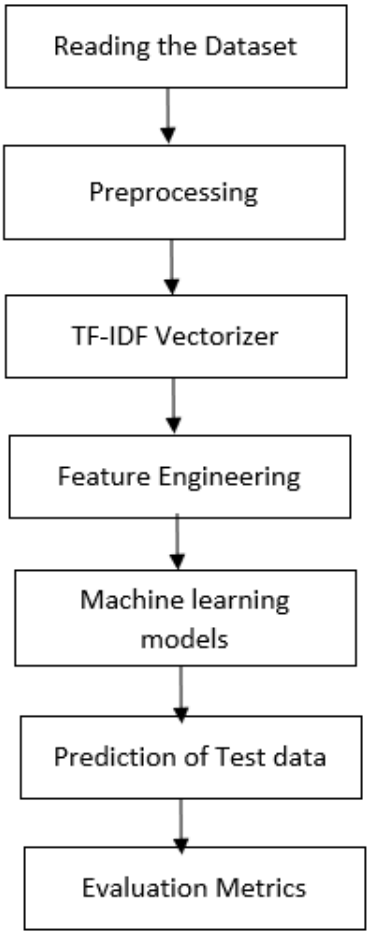

Figure 1: Workflow of the Model

4 System Overview

This section discusses the overview of the models submitted to the shared tasks. The flow of the model building is illustrated in Figure 1.

4.1 Data preprocessing

The shared task was provided with two kinds of input

(a) *Task-1*: text file contain tweets provided with its label, rephrased form and also the irony of the same for both English and Arabic language.

(b) *Task-2*: English text file considered for task 1 is used for irony identification in csv format.

The "Tweet" column from the datasets (Tasks 1 and 2) contains tweets, which must be preprocessed before extracting features for model creation. The preprocessing steps include tokenization, lemmatization, stop word removal and represented tweets

Dataset properties	Task-1 English	Task-2 English	Task-1 Arabic
No. of rows	3,467	3,467	2,601
No. of classes	2	6	2
No. of words	22,623	22,623	38,885
Vocabulary size	5,509	5,509	16,226
Maximum tweet length	72	72	31

Table 1: Description of the dataset used for Task-1 and Taks-2 in English and Task-1 in Arabic

as vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm (HB et al., 2016).The preprocessing of the tweets was carried out by using the functions available in the NLTK][1] library, whereas the sklearn TfidfVectorizer() [2] helps to vectorize the tweets.

4.1.1 Tokenization

Tokenization is the first step that we executed in preprocessing. Here, the tweet from the user is split into tokens for the ease of feature extraction.

4.1.2 Lemmatization

Lemmatization refers to correctly identifying the base form of a word and converting it into the meaningful base form considering the context.

4.1.3 Stopword removal

Stop word removal is performed to remove the most commonly occurring words in the tweet, such as pronouns and articles. A similar operation was performed on Arabic data by collecting a publicly available stopword list.

4.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a feature extraction method for vectorizing a sentence or tweet. The TF-IDF vector can be obtained for a sentence by computing Equation 1 for each word in that sentence.

$$TF - IDF(t, D) = TF(t, D) \times IDF(t) \quad (1)$$

Where the Term Frequency

$$TF(t) = \frac{N(t)}{T} \quad (2)$$

and Inverse Document Frequency

$$IDF(t) = log\frac{n}{df(t)} \quad (3)$$

where,
t is the word in a tweet, $N(t)$ is the number of times word t occurs in a document, T is the number of words in a document, n is the total number of sentences/tweets in the dataset, and $df(t)$ is the number of documents in which the term t appears.

4.3 SMOTE

SMOTE (Synthetic Minority Oversampling Technique) (Chawla et al., 2002) is an oversampling method for solving the class imbalance problem in the dataset. It resolves the problem by increasing the number of data points in the minority class with synthetically generated random data points. It is achieved by randomly selecting one or more k nearest neighbours of each minority class. The process can be initiated using the following steps:

1. Given the minority class S, for each $y \in S$, the nearest k-neighbours of y are obtained using Euclidean distance of y and every other elements in S.

2. Sampling rate T is given according to the proportion of imbalance. For each $y \in S$, T elements are selected randomly from nearest k-neighbours. And the set S_1 is made.

3. For every $y_k \in S_1$, $k = 1, 2, 3..., T$, the formula for generating new example (y')is,

$$y' = y + rand(0, 1) * |y - y_k| \quad (4)$$

The SMOTE algorithm was implemented using the SMOTE function available in the imblearn Python package[3].

4.4 Model development

We utilized K-Nearest neighbour (KNN) (Guo et al., 2003), Support Vector Machine (SVM) (Soman et al., 2009), Naïve Bayes (Huang and Li,

[1] https://www.nltk.org/

[2] https://scikit-learn.org/stable/modules/generated/sklear.feature_extraction.text.TfidfVectorizer.html

[3] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

2011), Decision Tree (Priyam et al., 2013) and Random Forest (Premjith et al., 2019) ensemble method for developing the models for various subtasks in iSarcasm. The procedures for model development for different tasks are given in the following subsections.

4.4.1 Sub task1: Sarcasm Identification

We built a binary classifier to determine whether the given tweet is sarcastic or not. Therefore, for the same purpose, we applied machine learning classifiers to the processed train data. For English tweet data, encouraging results were obtained by Decision tree and logistic regression. The decision tree is a particular type of probability tree that makes the decision about the process (Rahaman et al., 2021), and Logistic Regression is used for predicting the categorical dependent variable using a given set of independent variables (Sarsam et al., 2020). SVM and Random forest classifiers obtained the best performance for Arabic data. The Random Forest classifier reduces the bias due to overfitting and class imbalance between tweets. Bouazizi and Ohtsuki (Bouazizi and Ohtsuki, 2016) used logistic regression to label the data as sarcastic or non-sarcastic.

4.4.2 Sub task2: ironic speech category Identification

A multi-label classifier was developed for this task to determine the ironic speech category of the tweets. We applied a multi labelled classifier strategy with fitting one classifier per target, allowing multiple target variable classifications. The primary purpose behind this class is to extend estimators enabling estimation of a series of target functions mentioned in the dataset, which are trained using a single predictor matrix to predict a series of responses. We implemented a classification model using Logistic Regression, and a decision tree for the same as mentioned above (Sarsam et al., 2020)-(Rahaman et al., 2021).

4.5 Evaluation Metrics

The trained models were evaluated using macro F1-score, Precision, Recall and Accuracy. Accuracy is given by the ratio of the total number of correct predictions to the measure of total predictions done by the model, regardless of correct or incorrect predictions. Precision defines the actual positive among the predicted positive. The recall is a measure of the correctly classified total number of positives. Moreover, F1-score is the harmonic mean of precision and recall. Macro-average is defined as the average of precision, recall, and F1-score in different classes.

5 Experimental Setup

We implemented the models using Python version 3. The training data is split into train and validation sets for confirming the best performing model. In the Arabic sarcasm identification model using the SVM classifier (subtask 1), we used a range of gamma values $(0.1, 1, 10, 100)$ and c regularization parameter values $(0.1, 1, 10, 100)$ and changed the kernel type to RBF, linear and polynomial to see how the accuracy and F1-score vary. In random forest classifier different, n_estimators value $(10, 100, 1000)$ and the maximum features are given to $sqrt$, $log2$ to see the changes (Premjith and Kp, 2020). The English tweet irony detection model (subtask 2) is a multi-class classification problem and implemented using a multioutput classifier set to multilabel.

The model performance was analyzed using macro F1-score obtained using the sklearn metrics along with the accuracy, precision and recall (Pedregosa et al., 2011) of the trained model.

6 Result

All the subtasks were evaluated on the macro-average F1-scores of each information unit. We fixed the best performing models by using cross-validation. The Random Forest classifier and SVM obtained the best F1-scores for English task 1 and Arabic, respectively. In subtask 2, Logistic Regression gave the higher F1-score. We were officially ranked 23rd in task1 English with an F1-score of 0.4966 and accuracy of 56.71% using the Random Forest classifier and ranked 20th in Arabic with 0.6127 of F1-score and 79.21% accuracy using SVM binary classifier. In subtask 2, we were ranked 14th with a macro F1-score of 0.0567 using the Logistic Regression model. The obtained result from our model among all participating teams are shown in table 2, 3 below.

7 Conclusion

This paper presents the submission of Amrita_CEN towards the SemEval 2022 Task 6 competition named " iSarcasmEval - Intended Sarcasm Detection in English and Arabic ". A total of six machine learning algorithms were used, including five

Metrics	Task-1 English	Task-1 Arabic
F1-Sarcastic	0.3052	0.3490
F1-score	0.4966	0.6127
Precision	0.5550	0.6050
Recall	0.6121	0.6246
Accuracy	0.5671	0.7921

Table 2: Result for Subtask 1 English and Arabic

Metrics	Task-2 English
Macro-average F-score	0.0567
F1-score Sarcasm	0.2180
F1-score irony	0.0293
F1-score satire	0.0461
F1-score understatement	0.0074
F1-score overstatement	0.0245
F1-score rhetorical question	0.0150

Table 3: Result for Subtask 2 English

classical ML models and one ensemble technique. The class imbalance problems were dealt with by oversampling technique called SMOTE, and for evaluation, macro F1-score were considered for both the subtasks. The model trained using Random forest, SVM and logistic regression performed well among the subtasks given, and the results were submitted using the same.

8 Acknowledgements

We wish to express our sincere gratitude to our facualty Dr Sowmya V for her help throughout this work and the research underlying it. Your guidance, willingness to share the vast knowledge and expertise made us to understand this work and its manifestations in great depth and helped us to complete the assigned task on time.

References

Intissar Abbes, Yousra Hallem, and Nadia Taga. 2020. Second-hand shopping and brand loyalty: The role of online collaborative redistribution platforms. *Journal of Retailing and consumer Services*, 52:101885.

Gavin Abercrombie and Dirk Hovy. 2016. Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. In *Proceedings of the ACL 2016 student research workshop*, pages 107–113.

Ibrahim Abu Farha, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany. Association for Computational Linguistics.

David Bamman and Noah Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 574–577.

Mondher Bouazizi and Tomoaki Otsuki Ohtsuki. 2016. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:5477–5488.

John D Campbell and Albert N Katz. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*, 49(6):459–480.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 334–339, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2020. From arabic sentiment analysis to sarcasm detection: The arsarcasm dataset. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 32–39.

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Lrec*, pages 392–398. Citeseer.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. Idat at fire2019: Overview of the track on irony detection in arabic tweets. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 10–13.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. 2003. Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer.

Barathi Ganesh HB, M Anand Kumar, and KP Soman. 2016. Distributional semantic representation in health care text classification. In *FIRE (Working Notes)*, pages 201–204.

Yuguang Huang and Lei Li. 2011. Naive bayes classification algorithm based on small sample set. In *2011 IEEE International conference on cloud computing and intelligence systems*, pages 34–39. IEEE.

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1006–1011, Austin, Texas. Association for Computational Linguistics.

Jihen Karoui, Farah Benamara, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. 2017. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 1, pages 262–272.

Arpita Khare, Amisha Gangwar, Sudhakar Singh, and Shiv Prakash. 2022. Sentiment analysis and sarcasm detection of indian general election tweets.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Melbourne, Australia. Association for Computational Linguistics.

Akshi Kumar, Shubham Dikshit, and Victor Hugo C Albuquerque. 2021. Explainable artificial intelligence for sarcasm detection in dialogues. *Wireless Communications and Mobile Computing*, 2021.

Silviu Oprea and Walid Magdy. 2019. Exploring author context for detecting intended vs perceived sarcasm. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2859, Florence, Italy. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2020a. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online. Association for Computational Linguistics.

Silviu Vlad Oprea and Walid Magdy. 2020b. The effect of sociocultural variables on sarcasm communication online. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–22.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

B Premjith and Soman Kp. 2020. Amrita_cen_nlp@ wosp 3c citation context classification task. In *Proceedings of the 8th International Workshop on Mining Scientific Publications*, pages 71–74.

B Premjith, KP Soman, M Anand Kumar, and D Jyothi Ratnam. 2019. Embedding linguistic features in word embedding for preposition sense disambiguation in english—malayalam machine translation context. In *Recent Advances in Computational Intelligence*, pages 341–370. Springer.

Anuja Priyam, GR Abhijeeta, Anju Rathee, and Saurabh Srivastava. 2013. Comparative analysis of decision tree classification algorithms. *International Journal of current engineering and technology*, 3(2):334–337.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 213–223.

Arifur Rahaman, Ratnadip Kuri, Syful Islam, Md Javed Hossain, and Mohammed Humayin Kabir. 2021. Sarcasm detection in tweets: A feature-based approach using supervised machine learning models. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 12(6).

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 97–106.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 704–714.

Samer Muthana Sarsam, Hosam Al-Samarraie, Ahmed Ibrahim Alzahrani, and Bianca Wright. 2020. Sarcasm detection using machine learning algorithms in twitter: A systematic review. *International Journal of Market Research*, 62(5):578–598.

KP Soman, R Loganathan, and V Ajay. 2009. *Machine learning with SVM and other kernel methods*. PHI Learning Pvt. Ltd.