reamtchka at SemEval-2022 Task 6: Investigating the effect of different loss functions for Sarcasm detection for unbalanced datasets

Reem Abdel-Salam
Computer Engineering Department
Cairo University
reem.abdelsalam13@gmail.com

Abstract

This paper describes the system used in SemEval-2022 Task 6: Intended Sarcasm Detection in English and Arabic. Achieving 20th, 3rd places with 34& 47 F1-Sarcastic score for task A, 16th place for task B with 0.0560 F1-macro score, and 10, 6th places for task C with 72% and 80% accuracy on the leaderboard. A voting classifier between either multiple different BERT-based models or machine learning models is proposed, as our final model. Multiple key points have been extensively examined to overcome the problem of the unbalance of the dataset as: type of models, suitable architecture, augmentation, loss function, etc. In addition to that, we present an analysis of our results in this work, highlighting its strengths and shortcomings.

1 Introduction

Sarcasm detection in any language text is hard for many reasons. First Sarcasm is much more than just written words. It is the tone, emphasis, experience, personal knowledge, and even facial expressions and body language that convey the meaning. When written into text such information is lost completely. Another reason is cultural references which make it hard for non-natives to get. In addition, all that is needed to show sarcasm is to be clear and articulate and not be as sketchy and without formatting, punctuation, proper use of language as texting might be for some people in their usual use of such media. Sarcasm is widespread on the social web and, by definition, may be extremely disruptive to machine learning/ deep learning models that use this data to perform tasks such as sentiment analysis, opinion mining, author profiling, and harassment identification. Thus, sarcasm detection might be the first crucial step in these systems. Several methods have been proposed to detect sarcasm in text as the recent shared-task for sarcasm detection in Arabic language (Abu Farha et al., 2021), where several teams used pretrained language models such as AraBERT and MarBert (Wadhawan, 2021; Song et al., 2021; Naski et al., 2021; Faraj et al., 2021; Abdel-Salam, 2021; Abuzayed and Al-Khalifa, 2021). In some of the work proposed in English language sarcasm detection involved rule-based and statistical approaches using: (a) Unigrams and pragmatic features (for example emoticons, etc.) (b) Sentiment and polarity estimation (c) Extraction of common patterns (Nagwanshi and Madhavan, 2014; Kumar et al., 2020; Bouazizi and Ohtsuki, 2016). A Recent shared-task for English language (Ghosh et al., 2020), where several methods has been proposed were based on pretrained model (i.e. BERT and RoBERTa) (Baruah et al., 2020; A. and D., 2020; Amir et al., 2016; Shangipour ataei et al., 2020).

In SemEval-2022 Task 6 (Abu Farha et al., 2022), the goal is to determine if sarcasm is presented in text or not, based on the text that is manually labeled by the task organizers. The shared task consists of three subtasks:

- Subtask A: this subtask is a binary classification task, where the goal is to determine whether is tweet is sarcastic or not, for the English language and Arabic languages. The official metric for this subtask is F1-Sarcastic.
- Subtask B: this subtask is a multi-label classification task where the goal is to determine which ironic speech category the tweet belongs to (English only). There are 6 ironic speech categories which are: irony, sarcasm, satire, understatement, overstatement, and rhetorical question. The official metric for this subtask F1- Macro score
- Subtask C: the goal of this task is to determine which text is more sarcastic given a pair of texts. The official metric for this subtask is accuracy.

This paper describes the system developed by the rematchka team for SemEval-2022 Task 6. Given

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 896 - 906

that a key challenge in this task is the limited size of annotated data and the unbalanced distribution, we follow best practices from recent work on enhancing model generalization and robustness and propose a voting classifier model that leverages pretrained representations(i.e. BERT and RoBERTa).

The main contributions of our work are as follows:

1. Identifying appropriate loss functions to help train Bert-Base models and Deep learning models in presence of extremely unbalanced datasets.
2. Investigating the importance of different layers in Bert-Base models. In addition, we present an analysis of our results in this work, highlighting its strengths and shortcomings.

Our code is made public and can be found here[1]. The rest of the papers goes as follow: section 2 discusses the proposed methods, section 4 shows experimental results, and section 5 concludes the paper.

2 System Overview

In this section, we discuss our prepossessing techniques, different hand-crafted features. We further explore different training techniques using Bert (Devlin et al., 2018), RoBERTa (Liu et al., 2019), BertTweet (Nguyen et al., 2020), Deberta (He et al., 2020), XLNet (Yang et al., 2019), and CANINE (Clark et al., 2022) models for English Language, and AraBert (Antoun et al., 2020), MarBert (Abdul-Mageed et al., 2021) and QARiB (Abdelali et al., 2021) models for Arabic Language. Moreover we investigate machine learning based models.

2.1 Dataset

In subtasks A, B, and C of the English language the dataset provided consisted of a total of 3468 manually annotated tweets. As shown in table 2, for subtask A English language, 867 of the total tweets are labeled sarcastic while 2601 are labeled not sarcastic. While in subtask A, C Arabic language the dataset provided consisted of 3102 manually annotated tweets. 745 of the total tweets are labeled sarcastic while 2357 are labeled not sarcastic. Table 1 shows distribution of labels for subtask B. For subtask B irony types: 713 out of the 867 sarcastic tweets are labeled sarcasm, 155 out of 867 are labeled irony, 25 out of 867 are labeled satire, 10 out of 867 are labeled understatement, 40 are labeled overstatement and 101 are labeled rhetorical-question.

2.2 Machine learning Based Approaches

The machine learning pipeline for subtask A and subtask B goes as follows: given a tweet a set of features are computed: lexical, syntactic, semantic, pragmatic, and polarity feature representations. These features are then fed to multiple models such as SVM, Logistic Regression (LR), random forest, boosting classifiers, and Xgboost for classification.

The Sarcastic tweets in the provided dataset are categorized as sarcasm, irony, satire, overstatement, rhetorical question, and understatement. It is crucial to extract features that cover those classes. For example, overstatement contains exaggerated terms, one way to extract it is to calculate the number of elongated punctuation or the number of characters in the word. Therefore a large set of hand-crafted features including lexical, syntactic, semantic, pragmatic, and polarity features are used.

Lexical features our lexical features contains word and character level n-grams. For word-level we use 1-gram, and for character level we use 4-gram, top 5000 n-grams are utilized based only on the term frequency-inverse document frequency (TF-IDF) values.

Syntactic features for syntactic features we use Spacy to calculate the number of adjectives, adverbs, nouns, pronouns, and verbs. In addition to that, we calculated the count of NER words in the tweet.

Sentiment & Polarity features sarcasm is used to express annoyance or outrage about a bad circumstance. As a result, people employ exaggerated and extremely positive terms to describe their negative condition (Yadollahi et al., 2017). Therefore it is important to extract them. For polarity & sentiment estimation, each tweet is divided into two parts. Then, for each part, its polarity and sentiment are calculated using NLTK Senti-WordNet (Baccianella et al., 2010). In addition to that, the overall polarity and sentiment for the whole tweet are calculated. Furthermore, the number of positive and negative sentiment words in a document, the number, and the count of the longest run of positives/negatives are computed as described in (Joshi

[1] https://github.com/rematchka/Intended-Sarcasm-Detection-In-English-and-Arabic-for-extremly-unbalanced-datasets

Task	Tweets Distribution					
	sarcastic	irony	satire	understatement	overstatement	rhetorical-question
B	713	155	25	10	40	101
Total	867					

Table 1: Dataset distribution for subtask B for English language

Language	Number of Tweets	Task	
		A	C
EN	Total Number	3468	1734
	Number of Sarcastic Tweets	867	867
	Number of Non-Sarcastic Tweets	2601	867
AR	Total Number	3102	1490
	Number of Sarcastic Tweets	745	745
	Number of Non-Sarcastic Tweets	3102	745

Table 2: Dataset distribution for subtask A and C for English and Arabic language

et al., 2015).

Pragmatic features it is hard to detect sarcasm in speech, as the word may have several meanings, and also the text contains behavioral aspects such as low tones, facial gestures, or exaggeration. These forms can be translated into elongation, repetition, and punctuation. To recognize such characteristics, we extract a set of features known as punctuation-related features. For each tweet, the following is computed:

- Presence of emoji's (González-Ibánez et al., 2011)
- Count of number of question marks
- Count of number of colons
- Count of number of a dollar sign
- Count of the number of quotes.
- Count of the number of exclamation marks.
- Count of number of special characters
- Count of number of hashtags
- Count of number of mentions
- Rate of capitalization
- Mixed cases count
- Rate of punctuation

Although punctuation is not relevant in itself and may not indicate whether the user is expressing sarcasm or any other emotion, when paired with other features, these attributes are anticipated to provide value to the classification.

Semantic& other features semantic features usually capture the conceptual relationship between words. For this we extracted multiple semantic features such as the presence of contradiction, interjections, the number of laughing expressions (Bouazizi and Ohtsuki, 2016) and the number of specific hashtags such as: "irony", "sarcasm", "hypocrisy" and "seriously". In addition to these features, other features are extracted as profanity count, topic modeling, and the presence of a numeric mismatch.

2.3 BERT-based Models

Figure 1 and 2 show overall architecture for BERT-based models used for most of the subtask A, subtask B and subtask C. The pipeline for training for most of the models in subtasks A and B as shown in figure 1 goes as follows, the input text is fed to BERT-base models, and the output of arbitrary 4 layers is taken and fed to KimCNN. The output is fed finally to the Fully connected layer (FC) layer. 4 losses can be used to assess the model performance F1-Cross-Entropy, Recall-Cross-Entropy, Balanced loss, and Asymmetric loss.

2.4 Subtask A& B English language

One of the biggest challenges in this task was the presence of an extremely unbalanced dataset. Using the provided dataset without any external dataset or modification of the widely used loss function as cross-entropy, hinge loss, etc leads to bad model performance, where the model focuses only on the majority class which was the Non-Sarcastic class. Even data augmentation couldn't boost performance that much. In one of our early experiments using RoBERTa model on the provided dataset, the model could achieve 77% accuracy and an F1-sarcastic score of zero. Therefore, increasing the length of the dataset seemed crucial at that point to allow better fine-tuning. NLPAug (Ma, 2019) was used for augmentation. Spelling augmentation and ContextualWordEmbsAug for insertion and substitution using Bert and RoBERTa

were used. Furthermore, SynonymAug and ContextualWordEmbsForSentenceAug were used to make the dataset balanced, resulting in a balanced dataset where the number of non-sarcastic examples was 1560 and the number of sarcastic examples was 1040. Using this setting with RoBERTa model the model performance improved having F1-sarcastic 27 and accuracy 73% on the dev set. However augmentation didn't work all the time, some augmentation may lead to performance degrading or no improvements as RandomWordAug and WordEmbsAug using google news. Another interesting key point is that loss function does matter. Early experiments using cross-entropy with any BERT-based models achieved high accuracy however, achieved a 0 F1-sarcastic score on the dev set. When using loss functions that incorporate class imbalance the model performance improved. In one of the experiments conducted using Bert-Base uncased with sigmoid focal cross-entropy without any augmentation, the model performance improved reaching a 32 F1-Sarcastic score on the dev set. Therefore, we believe that there are 5 key points needed to be adjusted in order to improve model performance and recognition for the Sarcastic class:

- BERT-based models: which BERT-based model to use and achieve high performance.
- Model architecture: it is crucial to adjust the model architecture whether to use only the last layers and feed it to the Fully connected layer (FC), or whether to use the last n-layers from the model and apply average pooling which is then fed to FC layer, or feed output to Convolution or LSTM layer then FC layer. Also if important to choose which layer/s output will be used.
- Augmentation: it is important to decide whether to use augmentation to balance the dataset (however it is debatable whether it's okay to use them or not since the original dataset is manually labeled. Therefore augmentation might produce new examples we are not sure if it's truly sarcastic or not), or just down-sample the provided dataset by removing some examples.
- Loss function: loss function plays a key role in improving model performance. Therefore choosing the appropriate one will hugely impact the model performance to focus on both classes. A good choices may be sigmoid focal cross-entropy, recall-cross entropy loss (Tian et al., 2020), Dice Loss (Li et al., 2019), Asymmetric loss for multi-label classification and multi-class classification (Ben-Baruch et al., 2020), Distribution Balanced Loss (Wu et al., 2020) and F1-cross-entropy loss (which we believe is effective in our problem and propose).
- Data-Sampler: whether to use data-sampler instead of using augmentation. However, in this case, there are no examples removed from the dataset. The sampler (Yang et al., 2021) just makes sure that each batch fed to the model is balanced.

4 out of the 5 key points were heavily investigated: 1) BERT-based models, 2) Loss functions 3) Different architectures 4) Data-Sampler. For BERT-based models. It turns out that BertTweet, Bert-Base Uncased, RoBERTa, were the best performing models. Furthermore using multiple layers from the BERT-based model improves model performance drastically by 5-15%. Moreover, sigmoid focal cross-entropy, F1-cross-entropy, and also data-sampler improve model performance by 4-10%. F1-cross-entropy loss is denoted by this equation 1

$$\sum_{c=1}^{c=C} -(1 - F1_c) * N_c * log(P_c) \quad (1)$$

, where c is the class number, $F1_c$ is the F1-Score corresponding to specific class, P_c output of sigmoid/softmax for specific class c. Similarly, the recall-cross-entropy loss is denoted by this equation 2.

$$\sum_{c=1}^{c=C} -(1 - Recall_c) * N_c * log(P_c) \quad (2)$$

2.5 Subtask A Arabic Language

For the Arabic language, multiple Bert-Base models were fine-tuned on the provided dataset as AraBert, MarBert, and QARiB. Although the Arabic dataset was also imbalanced the model fine-tuning was easier. Without any modification to the dataset or the loss function, these models could achieve high performance on the dev set, unlike the English language. This might indicate that it is easier to detect sarcasm or find the sarcastic pattern in the Arabic language than the English.

2.6 Subtask C English and Arabic Languages

Based on Analysis and Experiments conducted in subtasks A & B, the following were investigated:

A) several architectures were investigated including the usage of the output of the last layer from Bert-Base models only of the usage of multiple layers from Bert-Base models. In addition to that, all architecture was similar to the siamese network where, the input to the model is both sarcastic and non-sarcastic, and the loss function is calculated based on the model prediction for both texts.

2.7 Final Recipe

In this subsection, we discuss the final models that were used during the evaluation phase for all subtasks.

2.7.1 subtask A English

A voter classifier is used based on different approaches since the voter classifier is model-agnostic, and can be tested on a variety of models and tasks to ensure that our findings are generalizable. 7 different approaches were used for the final classification. The first Model was Berttweet, where we used the output of the last four layers and fed it into KimCNN (Chen, 2015) while using recall-cross-entropy loss and data-sampler. The second model was Berttweet, where the input to the model was the original dataset. The input is fed to the model then the output of the last four layers is extracted and fed to four LSTM layers. The output of the four LSTM layers is fed into the FC layer. F1-cross-entropy loss is used with a data sampler. The third model is Bert-Base Uncased, where the input to the model was the original dataset and all hand-crafted features described in the previous subsection. The input is fed to the model then the output of the last four layers is extracted and fed to four LSTM layers. The output of the four LSTM layers is fed into the FC layer, the hand-crafted features are fed into another FC layer then the output of both is concatenated and fed to the last FC layer with sigmoid activation. F1-cross-entropy loss is used with a data sampler. For the fourth model, the architecture is similar to the first model except that Deberta model is used, and for the fifth model RoBERTa model is used with the same architecture and data-sampler as the first model but F1-cross-entropy loss is used instead of recall-cross-entropy loss. For the 6th and 7th models, linear SVM and linear SVM bagging classifiers were used with all of the set of hand-crafted features.

2.7.2 Subtask A Arabic

For this subtask, MarBert was used only. The input was fed to the model then the output was fed into FC layers. Cross-entropy loss was used.

2.7.3 Subtask B

The final model was a voting classifier between 5 models. The first and the second models were based on BertTweet model with KimCNN. However, the first model was trained using Asymmetric loss and the other model was trained using Distribution Balanced Loss with data sampler. RoBERTa model was used as the third model where the output of the 6th, up t 9th layers, were fed to KimCNN, and the model was trained using Asymmetric loss. For the 4th and 5th models multi-output linear SVM and multi-output linear SVM bagging classifiers were used with all of the set of the hand-crafted features.

2.7.4 subtask C English and Arabic

In the English subtask, the final model was voting classifiers between three models. The first model was based on XLM-RoBERTa (Conneau et al., 2019), where the output of the 6th, up t 9th layers, were fed to KimCNN and the model was trained using Margin ranking loss. For the second and third models, RoBERTa model was used with Margin ranking loss. However, for the second model architecture, the output of the last layer of RoBERTa was fed into FC layer, while for the third model the output of the 6th, up to 9th layer was fed to KimCNN. Similarly for the Arabic language subtask, a voting classifier between three models where used. The first and the second models were based on Arabert and MarBert where the output of the last layer was fed into FC layer, and the models were trained using Margin ranking loss. For the third layer, MarBert with KimCNN was utilized with Margin ranking loss for training. The whole training pipeline is demonstrated in figure 2. Since, the target of the subtask is to determine, which tweet is more sarcastic. During inference, both tweets are fed into the final model and whoever has a larger value is determined as a sarcastic tweet.

3 Experimental setup

Experiments were conducted via Python and PyTorch framework, running on Google Colab resources, which are Nvidia Tesla P100-PCIE-16GB GPU, Intel ® Xeon ® CPU @ 2.20 GHz, and 12GB RAM. We used 70%-10%-20% strategy for

train-validation-test splits respectively for the training phase and 5-fold cross-validation during training. All of the presented models (submitted to the leaderboard) were trained on the provided dataset for the shared task only with no augmentation and the same splitting criteria. If augmentation is used in the dev-phase/test-phase, the following types of augmentation were used: 1) Spelling augmentation and contextual Word embedding augmentation for insertion and substitution using BERT and RoBERTa. 2) Synonym augmentation and contextual word embedding for sentence augmentation. Precision, recall, f-score (f1-sarcastic for subtask A, and macro average f1-score for subtask B), and accuracy (for subtask C) were used as evaluation metrics. All BERT-based models were trained using an AdamW optimizer with an initial learning rate of 0.001, weight decay rate of 0.0000001, and cosine annealing learning rate scheduler with minimum learning rate values of 0.0000001 and maximum temperature of 500. All the models were fine-tuned for 3-5 epochs. Pre-processing wasn't conducted on the dataset. During inferences, an ensemble of the 5 models is used (due to 5-fold validation)and is referred to as 1 model during our discussion.

4 Results

In this section, we discuss our main results during the development and evaluation phases. In addition to that failed cases analysis is discussed.

4.1 Results on Trial and Simulated Data

The evaluation was based on the train-test split criteria on the provided dataset. Table 3 shows different models' performance on the dev-set, for subtask A in the English language based on F1-sarcastic and accuracy. Early experiments were based on augmentation and Dice/cross-entropy loss. The performance of the models started to increase when using suitable architecture, multiple outputs from BERT-based models, and a suitable loss function that is sensitive for low classes. The difference between RoBERTa with dice loss and augmentation and RoBERTa with KimCNN and F1-cross-entropy is around 20 in F1-score. Surprisingly machine learning models could achieve a good F1-Sarcastic score. Table 7 shows different models performance on the dev-set, for subtask B based on F1-macro. The model performance is not high due to the small size of the dataset. It can be seen that Distribution Balanced Loss and Asymmetric loss is a good choice for unbalanced multi-label classification. For subtask, A Arabic language MarBert could achieve 89 F1-sarcastic while AraBert and QARiB achieve 54 and 79 F1-sarcastic on dev-set. For subtask C English language XLM-RoBERTa KimCNN could achieve 89% accuracy, while RoBERTa KimCNn and RoBERTa could achieve 90% and 85% on dev-set. For subtask C Arabic language, AraBert could achieve 82%, while MarBert and MarBert KimCNN could achieve 83% and 76% on dev-set.

4.2 Test Results

Tables 5, 4, 6, and 8 show the results of the proposed models and the official model (Voting classifier) that was used in the leaderboard. The voting classifier model is our official submitted model. Table 4 shows the results of our submitted model and the performance of each individual model that is used in the voting classifier. Table 8 shows the results of our submitted model (the voting classifier) in both languages. In addition, the performance of each model contributed to the voting classifier. For subtask C, In the English language, the voting classifier is composed of XLM-RoBERTa and RoBERTa with KimCNN and RoBERTa. For the Arabic language, the voting classifier is composed of MarBert, MarBert, and MarBert with KimCNN. Table 5 shows the results of our submitted model (the voting classifier). In addition to other models that were described in the dev-phase. The final voting classifier for subtask A English is composed of Bagging SVM with feature engineering, SVM with feature engineering, RoBERTa with KimCNN and F1-Cross-Entropy, Deberta with KimCNN and Recall-Cross-Entropy, BertTweet with KimCNN and Recall-Cross-Entropy, BertTweet with LSTM and F1-Cross-Entropy, Bert-Base-Uncased with LSTM and feature engineering and Recall-Cross-Entropy.

Our voting based classifier achieved 20th, while MarBert achieved 3rd place for subtask A, voting based classifier achieved 16th place for subtask B and 10, 6th places for subtask C. For subtask C the performance of the voting classifier and the other models are similar to the dev-set. For subtask B the voting classifier model performance was bad compared to a single model. All the models failed to detect over-statement in the test set. For subtask A English languages the voting classifier model performance was bad compared to a single model,

this happened as the performance of machine learning models was bad on the test set. For subtask A, Arabic language the model performance falls below the performance observed on the dev-set. Based on tables 3 and 5 it can be seen that suitable architecture and loss aware function are the key to better performance. In addition to that, it seems that Augmentation has promising performance, and might boost results significantly if integrated with suitable architecture and loss function.

4.3 Error Analysis

In subtask A, the test-set contains 1400 examples, 200 of which are sarcastic, and 1200 of which are non-sarcastic. For subtask1 English language, the submitted model (Voting classifier) miss-classified 224 tweets as sarcastic and 113 as not-sarcastic. In the misclassifications, there were numerous prominent trends. Among the false negatives and false positives in the samples (Most of the time), the sarcasm and non-sarcasm communicated in many cases could only be extrapolated via world knowledge (for example:“Max Verstappen is such a clean driver, he never makes dirty moves when racing." and “This does not surprise me! Kat is a PR queen "). Some of the false positives (detected as sarcasm which is not) depend on the personality and the situation, which is in reality hard to detect is it a touch of sarcasm or not (for example: “Brrrr it’s cold outside...I love it!" and “What people?"). For the Arabic language subtask A, Marbert miss-classified 324 tweet as sarcastic and 36 as not-sarcastic. Similarly most the wrongly tweets miss-classified as no-sarcastic was due to world knowledge (for example: trasnalted as “fifi abdo the ideal mother"). الام المثالية فيفي عبده

Some of the miss-classified tweets as sarcastic was related to personal life and personality (for example: trasnslated as “The class who shouldn’t be called/named"). الدفعة التي لا يجب ذكر اسمها
In subtask B, the test-set contains 1400 examples, 180 are labeled sarcastic, 20 labeled irony, 49 labeled satire, 1 labeled under-statement, 10 labelled over-statement, and 11 labelled rhetorical-question. One the biggest problems that the dataset contained alot of non-sarcastic tweets, which affected the model prediction performance. A modification that should have done was to enter the tweet to subtask A models and then for the predicted sarcastic tweet subtask B model should be used to determine the type. In subtask C, the test-set contains 200 examples, 107 of them text 0 is more sarcastic than text 1. The voting model miss-classified 52 as text 0 being sarcastic and 4 examples as text 1 being more sarcastic. Some of the errors were a result of unclear sarcasm as an example these two text“Brr! It’s really cold outside today" and “I’m loving how warm it is outside today". The more sarcastic text is the second one, however, it’s hard to determine whether the weather is hot or cold, to come to this conclusion. Another common type of error is world knowledge as an example: “Benioff and Weiss will definitely go down in history for how terrible the script they wrote for GOT S8 was." and “Benioff and Weiss will definitely go down in history for their amazing work on the script of GOT S8.". If the model doesn’t have a knowledge about the TV series and rating it’s hard to tell that the second on is the sarcastic. In conclusion common error can be classified to:

- World knowledge
- Personality
- Personal experiences.

Model	F1-Saracstic
MarBert	0.4767

Table 6: Results for subtask A on the official test-set for Arabic language.

5 Conclusion

In this paper, we have discussed our system submitted to the SemEval-2022 Task 6. our model ranked 20th out of 43 participating teams in subtask A English language, 3rd out of 32 participating teams in subtask A Arabic language, 16th out of 22 participating teams in subtask B, 10th out of 16 teams in subtask C English language, and 6th out of 12 participating teams in subtask C Arabic language. We proposed a voting classifier that leverages fine-tuned, per-trained models. We proposed the usage of the different loss functions in each task to accommodate dataset imbalance and improve model training. Overall we showed the power of the loss function to improve model performance without the need to access any external dataset or to use any kind of augmentation. We also investigated the main common error, which disturbs model performance. In future efforts, we plan to further improve our model to better handle data-imbalance constraints and world knowledge needed to improve model performance.

Model	Loss Function	Augmentation	Accuracy	F1-Sarcastic
Bert-Base-Uncased with KimCNN	Recall-Cross-Entropy	No	74%	38
Bert-Base-Uncased with LSTM and Feature Engineering	Recall-CrossEntropy	No	75%	41
Bert-Base-multilingual-uncased	Dice Loss	Yes	73%	32
Bert-Base-Uncased	Dice Loss	Yes	76%	35
Bert-Base-Uncased	Cross-Entropy	Yes	73%	38
RoBERTa	CrossEntropy	Yes	73%	27
RoBERTa with KimCNN	F1-Cross-Entropy	No	75%	43
BertTweet with KimCNN	Recall-Cross-Entropy	No	79%	57
BertTweet with LSTM	F1-Cross-Entropy	No	78%	54
CANNIE	F1-Cross-Entropy	No	77%	33
Deberta with KimCNN	Recall-Cross-Entropy	No	77%	43
SVM + Feature Engineering	_	Yes	58%	38
SVM + Feature Engineering	_	No	63%	40
Logistic regression + Feature Engineering	_	Yes	57%	38
Logistic regression + Feature Engineering	_	No	55%	37
Bagging SVM + Feature Engineering	_	Yes	54%	34
Bagging SVM + Feature Engineering	_	No	66%	40

Table 3: Results for subtask A on the dev-set for various models and techniques for English language.

Model	Loss Function	F1-Macro	F1-Sarcasm	F1-irony	F1-satire	F1-understatement	F1-overstatement	F1-rhetorical question
RoBERTa + KimCNN	Distribution Balanced Loss	0.09	0.23	0.04	0.18	0.00	0.02	0.09
RoBERTa + KimCNN	Asymmetric loss	0.09	0.23	0.04	0.17	0.00	0.02	0.09
BertTweet + KimCNN	Asymmetric loss	0.10	0.23	0.05	0.16	0.00	0.03	0.11
BertTweet + KimCNN	Recall-Cross-Entropy	0.05	0.23	0.03	0.00	0.00	0.02	0.00
BertTweet + KimCNN	Distribution Balanced Loss	0.12	0.23	0.11	0.17	0.00	0.06	0.11
SVM+ FE	_	0.07	0.20	0.03	0.09	0.01	0.02	0.09
Bagging SVM + FE	_	0.08	0.24	0.03	0.14	0.00	0.02	0.09
Voting	_	0.0560	0.2251	0.0285	0.0664	0.0000	0.0161	0.0000

Table 4: Results for subtask B on the official test-set for various models and techniques for English language. The voting Classifier model is our official submitted model which is composed of the above models.

Model	Loss Function	Augmentation	Accuracy	F1-Sarcastic
Bert-Base-Uncased with KimCNN	Recall-Cross-Entropy	No	74%	28
Bert-Base-Uncased with LSTM and Feature Engineering	Recall-Cross-Entropy	No	76%	37
Bert-Base-Uncased	Dice Loss	Yes	79%	32
Bert-Base-Uncased	Cross-Entropy	Yes	74%	31
RoBERTa with KimCNN	F1-Cross-Entropy	No	73%	31
BertTweet with KimCNN	Recall-Cross-Entropy	No	76%	40
BertTweet with LSTM	F1-Cross-Entropy	No	75%	42
Deberta with KimCNN	Recall-Cross-Entropy	No	78%	32
SVM + Feature Engineering	_	No	55%	21
Bagging SVM + Feature Engineering	_	No	60%	18
Voting Classifier	_	_	_	34.05

Table 5: Results for subtask A on the official test-set for various models and techniques on English language. The voting Classifier model is our official submitted model.

Model	Loss Function	F1-Macro
RoBERTa + KimCNN	Distribution Balanced Loss	38
RoBERTa + KimCNN	Asymmetric loss	43
BertTweet + KimCNN	Asymmetric loss	44
BertTweet + KimCNN	Recall-CrossEntropy	21
BertTweet + KimCNN	Distribution Balanced Loss	44
SVM+ FE	_	32
Bagging SVM + FE	_	37

Table 7: Results for subtask B on the dev-set for various models and techniques for English language.

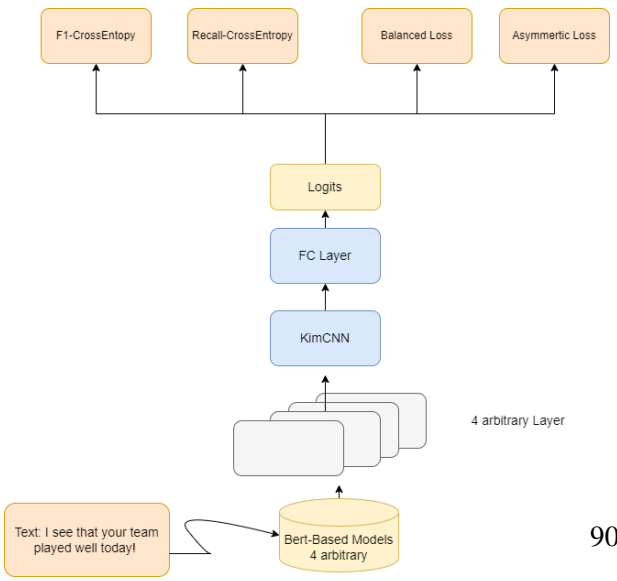

Figure 1: The overall architecture of our proposed BERT-based system used for subtask A and B.

Model	Language	Accuracy
AraBert	Ar	73
MarBert	Ar	81
MarBert + KimCNN	Ar	70
Voting	Ar	80
XLM-RoBERTa+ KimCNN	En	73
RoBERTa+ KimCNN	En	72
RoBERTa	En	69
Voting	En	72

Table 8: Results for subtask C on the official test-set for various models and techniques For English and Arabic language. The voting Classifier model is our official submitted model.

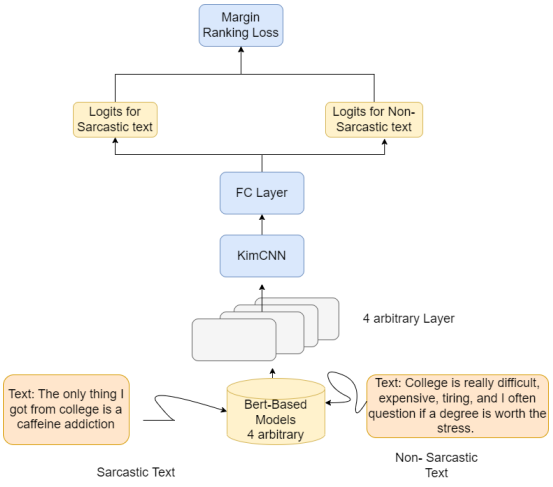

Figure 2: The overall architecture of our proposed BERT-based system used for subtask C.

References

Kalaivani A. and Thenmozhi D. 2020. Sarcasm identification and detection in conversion context using BERT. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 72–76, Online. Association for Computational Linguistics.

Reem Abdel-Salam. 2021. WANLP 2021 shared-task: Towards irony and sentiment detection in Arabic tweets using multi-headed-LSTM-CNN-GRU and MaRBERT. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 306–311, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

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.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 296–305, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Abeer Abuzayed and Hend Al-Khalifa. 2021. Sarcasm and sentiment detection in arabic tweets using bert-based models and data augmentation. In *Proceedings of the sixth Arabic natural language processing workshop*, pages 312–317.

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. 2020. Context-aware sarcasm detection using bert. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 83–87.

Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2020. Asymmetric loss for multi-label classification. *arXiv preprint arXiv:2009.14119*.

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

Yahui Chen. 2015. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dalya Faraj, Dalya Faraj, and Malak Abdullah. 2021. SarcasmDet at sarcasm detection task 2021 in Arabic using AraBERT pretrained model. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 345–350, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. 2020. A report on the 2020 sarcasm detection shared task. *arXiv preprint arXiv:2005.05814*.

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762.

Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti. 2020. Sarcasm detection using multi-head attention based bidirectional lstm. *Ieee Access*, 8:6388–6397.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Prateek Nagwanshi and CE Veni Madhavan. 2014. Sarcasm detection using sentiment and semantic features. In *KDIR*, pages 418–424.

Malek Naski, Abir Messaoudi, Hatem Haddad, Moez BenHajhmida, Chayma Fourati, and Aymen Ben Elhaj Mabrouk. 2021. iCompass at shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 381–385, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Taha Shangipour ataei, Soroush Javdan, and Behrouz Minaei-Bidgoli. 2020. Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 67–71, Online. Association for Computational Linguistics.

Bingyan Song, Chunguang Pan, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at WANLP-EACL2021 task 2: A deep ensemble-based method for sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 390–394, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Junjiao Tian, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-pang Chiu, and Zsolt Kira. 2020. Recall loss for imbalanced image classification and semantic segmentation.

Anshul Wadhawan. 2021. Arabert and farasa segmentation based approach for sarcasm and sentiment detection in arabic tweets. *arXiv preprint arXiv:2103.01679*.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision*, pages 162–178. Springer.

Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2):1–33.

Ming Yang, Jihwan Bang, Jirka Borovec, Tobias, and Davi Innovation. 2021. Imbalanced dataset sampler using pytorch.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.