Plumeria at SemEval-2022 Task 6: Sarcasm Detection for English and Arabic Using Transformers and Data Augmentation

Mosab Shaheen[*]
Indian Institute of Technology Kanpur
mosab@iitk.ac.in

Shubham Kumar Nigam[*]
Indian Institute of Technology Kanpur
sknigam@iitk.ac.in

Abstract

The paper describes our submission to SemEval-2022 Task 6 on sarcasm detection and its five subtasks for English and Arabic. Sarcasm conveys a meaning which contradicts the literal meaning, and it is mainly found on social networks. It has a significant role in understanding the intention of the user. For detecting sarcasm, we used deep learning techniques based on transformers due to its success in the field of Natural Language Processing (NLP) without the need for feature engineering. The datasets were taken from tweets. We created new datasets by augmenting with external data or by using word embeddings and repetition of instances. Experiments were done on the datasets with different types of preprocessing because it is crucial in this task. The rank of our team was consistent across four subtasks (fourth rank in three subtasks and sixth rank in one subtask); whereas other teams might be in the top ranks for some subtasks but rank drastically less in other subtasks. This implies the robustness and stability of the models and the techniques we used.

1 Introduction

Sarcasm is a figurative language where speakers or writers usually mean the contrary of what they say. Recognizing whether a speaker or writer is sarcastic is essential to downstream applications to understand the sentiments, opinions, and beliefs correctly (Ghosh et al., 2020). Sarcasm is ubiquitous on the social media text and, due to its nature, can be highly divisive of computational systems that perform tasks on that kind of data such as sentiment analysis, opinion mining, and harassment detection (Van Hee et al., 2018; Bing, 2012; Rosenthal et al., 2014; Maynard and Greenwood, 2014).

Our team Plumeria participated in SemEval 2022 task 6 (Abu Farha et al., 2022) in all its subtasks on English and Arabic. Previous shared tasks on sarcasm detection (Hee et al., 2018; Ghanem et al., 2019; Ghosh et al., 2020; Abu Farha et al., 2021) have only two subtasks; one is sarcasm detection and another is predicting the type of sarcasm. However, in SemEval 2022 task 6 (Abu Farha et al., 2022) organizers formulate three subtasks for both languages following the methods described in (Oprea and Magdy, 2020).

Using the two datasets for English and Arabic, organizers formulate three subtasks as follows:

- **Subtask A (English and Arabic):** It is a binary classification subtask where submitted systems have to predict whether a tweet is sarcastic or not.
- **Subtask B (for English only):** It is a multi-label classification subtask where submitted systems have to predict one or more labels out of six ironic-speech labels: sarcasm, irony, satire, understatement, overstatement, and rhetorical question.
- **Subtask C (English and Arabic):** It is a binary classification subtask. Given two texts, a sarcastic tweet and its non-sarcastic rephrase which conveys the same meaning, submitted systems have to predict which text is the sarcastic one.

In this shared task, our submitted systems primarily focused on the transformer based approaches because of their success in the field of NLP. The multi-head attention mechanism in transformers captures the relations between the words in a sentence which helps in identifying sarcasm. Moreover, to capture long-term dependencies between words, especially the contradicting ones, we used a hierarchical network by stacking a BiLSTM layer on top of a transformer. In order to emphasize the important tokens afterwards, we tried adding a dot-product attention layer to give different weights to the tokens. For prediction, the final information

[*] These authors contributed equally to this work

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 923 - 937
July 14-15, 2022 ©2022 Association for Computational Linguistics

are passed to a fully connected layer followed by a linear layer with a softmax activation function for classification (i.e. subtask A and C) or a sigmoid activation function for multi-label classification (i.e. subtask B).

The major constraint of a neural network is that they need lots of data for training to give satisfactory results. In addition, if the dataset is imbalanced with few instances of a class, this can result in poor results for detecting instances that belong to the class. This motivates us to create biased datasets, towards the concerned class/label, from existing datasets by increasing the number of instances of such a class/label. A detailed explanation of dataset creation is in section 3. We also illustrate the composition of created datasets in each subtask and the performance of the models on them.

The rank of our team was consistent across most subtasks (fourth rank in three subtasks, sixth rank in one subtask, and tenth rank in one subtask); unlike many teams which scored high in one or two subtasks but scored considerably less in other subtasks. This shows that the robustness and the consistency of our methods. The biased datasets were crucial for subtask A and subtask B, while augmenting a dataset plays a key role in subtask C. We released the codes and datasets for all subtasks via GitHub[1].

The paper is organized as follows. Section 2 lists some abbreviations used across the paper. Section 3 shows the datasets we used for fine-tuning the models. In Section 4 we list the preprocessing types applied on the datasets. The experiments and results are presented in section 5, and the analysis of the results is presented in section 6. This is followed by a conclusion in section 7.

2 Abbreviations

The following abbreviations were used frequently, and are shown in Table 1.

Full Form	Abbreviation
Language	Lang
Sarcastic	S
Non-Sarcastic	NS
English	En
Arabic	Ar
External	Ext
True Positive	TP
False Positive	FP
True Negative	TN
False Negative	FN

Table 1: Abbreviations used in the paper.

3 Datasets

The organizers provided datasets for English and Arabic. Regardless of the dataset, these are the fields in each row of a dataset:

- **Tweet**: a text specifying a tweet. This field is for all subtasks.
- **Sarcastic**: a binary field specifying whether a tweet is sarcastic or not. This field is for subtask A.
- **Rephrase**: a text specifying a non-sarcastic rephrase of a sarcastic tweet. This field is for subtask C.
- **Sarcasm, Irony, Satire, Understatement, Overstatement, and Rhetorical question** (English only): These binary fields are the labels of a sarcastic tweet. These fields are for subtask B.
- **Dialect** (Arabic only): a text specifying the dialect of a tweet from one of five dialects: Modern Standard Arabic (MSA), Egyptian, Levantine, Maghrebi, and Gulf. This field is for subtask A and C.

In the following sections we will describe the datasets given by the organizers, other available datasets, and augmented datasets.

3.1 Datasets Given by Organisers (Original)

The organizers released two training datasets for Arabic and English to train the systems on them for all subtasks. Later on, they released a test dataset for each subtask. We called these datasets "original" datasets as they are the official datasets for the subtasks. Information about the distribution of sarcastic and non-sarcastic tweets is presented in Appendix A.7 in Table 27 and Figure 6. Furthermore, information about sarcastic labels for subtask B is presented in Appendix A.7 in Table 28 and Figure 7.

3.2 Other Available Datasets (External)

The datasets in this section are not the official datasets and thus we called them "external" datasets. However, we used these datasets for subtask A and subtask B as they are created for similar subtasks.

[1] https://github.com/mosab-shaheen/iSarcasm-SemEval-2022-Task-6

Datasets Downloaded Using Twitter API: Initially the organizers provided the participants with train and test datasets which covered subtask A and subtask B for English only (later on the original datasets explained in subsection 3.1 were released instead). However, they provided the tweet ID instead of the tweet text and they asked the participants to download the tweet text using the Twitter API[2] and the tweet ID. Therefore, we downloaded the tweet texts we found for these two datasets. We were able to download 2841 tweets for training and 713 tweets for testing as shown in Table 27. The distribution of the tweets over the sarcastic labels is presented in Table 28.

Datasets of SemEval-2018 Task 3: These datasets are on same subtasks of subtask A and subtask B but for SemEval 2018 (Hee et al., 2018). We used the dataset for subtask A which has emojis and sarcasm hashtags. The datasets are available for download in this link[3]. More information about the distribution of sarcastic and non-sarcastic tweets is presented in Table 27.

ArSarcasm-v2 Dataset: It contains train and test datasets for sarcasm detection in Arabic (Abu Farha et al., 2021). Each row contains a tweet, sarcastic class, sentiment, and dialect. The datasets are available for download in this link[4]. More information about the distribution of sarcastic and non-sarcastic tweets is presented in Table 27.

3.3 Augmented Datasets

In addition to the original and external datasets, we created more datasets with more number of instances using the following methods:

1. **Augmenting an original dataset with external datasets:** We added instances to an original dataset from the matching external datasets either to balance it or just to augment it, and we filtered out the NAN entries .

2. **Augmenting a dataset using word embeddings:** For word embeddings we used Gensim library[5] together with GloVe word vectors[6] trained on two billion tweets with 100 dimension word vectors (glove-twitter-100). To create new instances in a dataset, we took a copy of one instance in the dataset and replaced up to four keywords in a tweet (or its rephrase) by replacing each keyword with one of the top three similar words according to the similarity between the corresponding word vectors, then we added the copied modified instance to the dataset and we repeated the process for other instances till we reached the required number of instances.

3. **Augmenting a dataset by repeating instances:** We repeated instances from a dataset mostly to balance the classes/labels in a dataset.

The final datasets used for each subtask is explained in the dedicated section for it.

4 Preprocessing

- **Type I:** no preprocessing

- **Type II:** Emotion icons were converted to their string text using the "emoji" Python library. Then, URLs were converted to "HTTPURL" token, also every mention in a tweet was converted to "@USER" token using regular expressions. These conversions was done because the BERTweet model (we will talk about it later) was pre-trained on tweets after these conversions.

- **Type III:** same as in Type II besides converting the smiley face codes e.g. ":-)" and ":)" to one of three values (smiley, sad, and playful). More than two successive occurrences of any punctuation like in "why?!!!!" were removed, then we removed more than two successive occurrences of same character like in "Superrrr" which can be found frequently in tweets. Moreover, a contraction (e.g. "isn't and "'cause") was replaced with its full form (e.g. "is not" and "because").

- **Type IV:** same as in Type III besides stemming and stop-word removal. For English we used WordNet lemmatizer and for Arabic we used ISRI stemmer. The NLTK Python library[7] was used for this purpose.

5 Approaches and Results for Subtasks

5.1 Conventions

In the following tables, if a table cell is highlighted with a light brown color, it means the score is

[2] developer.twitter.com/en/docs/api-reference-index
[3] SemEval2018-Task3 Dataset
[4] ArSarcasm-v2
[5] Gensim Library
[6] GloVe Word Vectors
[7] NLTK Python Library

among the best results, in the corresponding section of the table, on the validation dataset; whereas the one with brown color is the highest score. Furthermore, if a cell is highlighted with a green color, it means that the score is our final submission score (released by the organizers) in the subtask on the test dataset; whereas the one with blue color is the score of a submitted model but not the final one (a team can have multiple submissions).

5.2 Subtask A (English)

5.2.1 Datasets

- **Original**: The train split of the original dataset for English in Table 27 is splitted into train and validation datasets as shown in Table 2.
- **External**: As the measure for this task is F1-score for the sarcastic class, thus we created datasets which are biased towards the sarcastic class as shown in Table 29 in Appendix A.7.

Dataset	Total	S%	NS%
Original Train	2080	25	75
Original Val	1388	25	75

Table 2: Original datasets for subtask A (English) with the total number of tweets and the percentage of sarcastic (S%) and non-sarcastic (NS%) tweets.

5.2.2 Approaches

We primarily focused on the transformer based models. Since the task is a binary classification on tweets, the excellent choice to start with is BERTweet-base[8] and BERTweet-large[9] (Nguyen et al., 2020), a pre-trained language model on 845M English Tweets. Likewise, we tried the ELECTRA[10] (Clark et al., 2020) replaced token detection model (a pre-training task in which the model learns to distinguish real input tokens). In ELECTRA model, some tokens in the input are replaced with sample tokens instead of masking the tokens as in BERT. Moreover, we used a hierarchical network by passing the input tokens to the BERT model, then each token embedding is passed to a Bi-LSTM layer either with or without attention. The architecture of the BERT model, ELECTRA model, and hierarchical network is shown in Appendix A.3, A.4, and A.1 respectively. The final layer of each model was a linear layer with softmax activation function and we used the cross entropy loss function.

Note: We ran several experiments on all the approaches of this subtask with different datasets and preprocessing types. We also experimented with different learning rates, epochs, and loss functions to verify which one is performing best.

5.2.3 Results

Metric: The main metric is F1-score for the sarcastic class.

BERT: We used BERTweet-large in the following experiments, as it gave better performance than BERTweet-base, on the original train dataset shown in Table 2. We experimented with different learning rates and preprocessing types, and ran for 5 epochs. The results are shown in Table 3.

Learning Rate	Type	Val	Test
2 e - 6	I	0.0057	0.0293
	II	0.5017	0.457
	III	0	0
	IV	0	0
3 e - 6	I	0.3786	0.5068
	II	0.5552	0.4874
	III	0.5405	0.4972
	IV	0	0
4 e - 6	I	0.5585	0.4981
	II	0.4926	0.4717
	III	0.5407	0.4772
	IV	0	0
5 e - 6	I	0.5275	0.4724
	II	0.5655	0.4841
	III	0	0
	IV	0	0

Table 3: F1-score of the BERT model for subtask A (English) on original datasets.

From these experiments we found that Type II preprocessing is performing better than other types and same applies for the learning rate 4e-6. We conducted similar experiments on the external datasets in Table 29 in Appendix A.7 and we found similar results. We tried using the cross entropy loss function with and without weights on the external datasets using the same learning rate, preprocessing type, and number of epochs. We got our best result on the B4 dataset with weighted loss function which was our final submission score for this subtask. The results are shown in Table 4.

[8] HuggingFace Bertweet-Base

[9] HuggingFace Bertweet-Large

[10] HuggingFace Electra Large Discriminator

Biased	Loss1: Without Weights		Loss2: W1=1/#NS, W2=1/#S	
	Val	Test	Val	Test
B0	0.5784	0.4487	0.5944	0.4519
B1	0.5714	0.4548	0.5738	0.479
B2	0.5767	0.465	0.5951	0.4917
B3	0.601	0.4626	0.5931	0.4817
B4	0.5954	0.4727	0.6025	0.4769
B5	0.5874	0.5142	0.5803	0.5008
B6	0.5858	0.4791	0.5624	0.4884
B7	0.5957	0.5016	0.5637	0.5034
B8	0.584	0.492	0.5814	0.5
B9	0.5723	0.487	0.5554	0.49

Table 4: F1-score of the BERT model for subtask A (English) on external datasets.

We used 5 epochs and 4e-6 learning rate because they gave the best results as shown in Appendix A.5.

The official scores and leader-board ranks of the teams for subtask A (English) are shown in Table 5.

Rank	User	F-1 sarcastic
1	stce	0.6052
2	emma	0.5691
3	saroyehun	0.5295
4	**ShubhamKumarNigam**	**0.4769**

Table 5: Scores and leader-board ranks for subtask A (English).

ELECTRA: We used the ELECTRA model on the external datasets with Type II preprocessing, 6e-6 learning rate, and 5 epochs as they were performing the best as shown in Table 6.

Biased	Val	Test
B0	0.5525	0.4684
B1	0.4002	0.25
B2	0.5738	0.4762
B3	0.4002	0.25
B4	0.5756	0.4879
B5	0.4002	0.25
B6	0.5468	0.4642
B7	0.5702	0.4789
B8	0.479	0.5073
B9	0.4002	0.25

Table 6: F1-score of the ELECTRA model for subtask A (English) on external datasets.

BERT+BiLSTM with and without attention: The results we got using this architecture were not deterministic (i.e. they change when re-running the experiment and they may become better or worse than the results of BERT alone) and thus we did not use this model for the official submission. More details about the results of the model can be found in Appendix A.6.

5.3 Subtask A (Arabic)

5.3.1 Datasets

Original: The train split of the original dataset for Arabic in Table 27 is splitted into train and validation datasets as shown in Table 7.

Dataset	Total	S%	NS%
Original Train	1861	24	76
Original Val	1241	24	76

Table 7: Original datasets for subtask A (Arabic).

External: Same as in subtask A (English), we created datasets which are biased towards the sarcastic class as shown in Table 30 in Appendix A.7.

5.3.2 Approaches

The approaches used here are similar to subtask A (English) except for the used transformers. Since the data is in the Arabic language, we tried some models from The **C**omputational **A**pproaches to **M**od**e**ling **L**anguage (CAMeL) research lab [11]. They majorly focused on Arabic and Arabic dialect processing, machine translation, text analysis, and dialogue systems.

The models are available on the Hugging Face library. CAMeLBERT is a collection of BERT models pre-trained on Arabic texts with different sizes and variants (Inoue et al., 2021). They released pre-trained language models for Modern Standard Arabic (MSA), dialectal Arabic (DA), and classical Arabic (CA). We tried CAMeLBERT-DA and CAMeLBERT-Mix for sarcasm detection. Likewise, we tried the AraBERT v2 which is a pre-trained BERT based on Google's BERT architecture for Arabic Language Understanding[12] (Antoun et al.).

5.3.3 Results

Metric: The main metric is F1-score for the sarcastic class.

BERT: We used CAMeLBERT-Mix in the following experiments as it performed the best among other BERT models. We applied it on the external datasets with non-weighted cross entropy loss function, 5 epochs, and 2e-5 learning rate because they

[11] HuggingFace CAMeL-Lab

[12] HuggingFace Bert-Base-Arabert-v02

gave the best results, which included our final submission score for this subtask, as shown in Table 8.

Biased	Val	Test
B0	0.7168	0.3438
B1	0.7025	0.4163
B2	0.7131	0.4071
B3	0.6804	0.4335
B4	0.7015	0.4186
B5	0.6927	0.4332
B6	0.7012	0.4048
B7	0.7124	0.4365
B8	0.6731	0.4589
B9	0.7094	0.4589

Table 8: F1-score of the BERT model for subtask A (Arabic) on external datasets.

BERT+BiLSTM+Attention: We used attention with BiLSTM on top of BERT model. The results also were not deterministic. However, the best results for this architecture occurred when using 5 epochs and 9e-6 learning rate on B3 and B9 datasets as shown in Table 9.

Biased	Hidden State Size	Val	Test
B3	50	0.6849	0.4234
B9	1000	0.7123	0.4693

Table 9: F1-score of the BERT+BiLSTM+Attention model for subtask A (Arabic).

The official scores and leader-board ranks of the teams for subtask A (Arabic) are shown in Table 10.

Rank	User	F-1 sarcastic
1	Abdelkader	0.5632
2	Aya	0.5076
3	rematchka	0.4767
10	**ShubhamKumarNigam**	**0.4072**

Table 10: Scores and leader-board ranks for subtask A (Arabic).

5.4 Subtask B

5.4.1 Datasets

Original: The train split of the original dataset for English in Table 28 in Appendix A.7 is splitted into train and validation datasets as shown in Table 11.

External: The original and external datasets presented in Table 28 in Appendix A.7 (without the validation dataset) were added together to form a new dataset (Ext-NB). Then the resulting dataset was balanced either by using word embeddings (Ext-UW) or by repeating instances (Ext-UR). We created a dataset (Ext-EB) to give more importance to the labels of low number of instances by repeating the instances of these labels up to the limits specified by these heuristic formulas:

#irony=#sarcasm*(1+1/sqrt(#irony)) (1)

#satire=#sarcasm*(1+2/sqrt(#satire)) (2)

#understatement=#sarcasm*(1+3/sqrt(understatement)) (3)

#overstatement=#sarcasm*(1+1.5/sqrt(overstatement)) (4)

#rhetorical=#sarcasm*(1+1.2/sqrt(#rhetorical)) (5)

Dataset	Total	Sarcasm Under- statement	Irony Over- statement	Satire Rhetorical question
Original Train	606	67.60% 1%	15.10% 3.50%	2.60% 10.20%
Original Val	261	70% 1%	14.20% 4.50%	1.90% 8.40%

Table 11: Original datasets for subtask B (English).

The datasets are shown in Table 31 in Appendix A.7.

5.4.2 Approaches

This subtask primarily focused on BERTweet-large. As it is a multi labeling subtask, we used sigmoid as the activation function in the last layer and binary cross entropy as the loss function.

5.4.3 Results

Metric: The main metric is Macro-F1 score.

BERT: We used BERTweet-large in the following experiments on the external datasets using 5 epochs, 6e-6 learning rate, and Type II preprocessing. The results are shown in Table 12 which included our final submission score for this subtask.

Dataset	Val	Test
Ext-NB	0.1513	0.038
Ext-UW	0.318	0.0716
Ext-UR	0.3412	0.076
Ext-EB	0.4152	0.0778

Table 12: Macro-F1 score of the BERT model for subtask B (English) on external datasets.

The official scores and leader-board ranks of the teams for subtask B (English) are shown in Table 13.

Rank	User	macro F1-score
1	Duxy	0.163
2	Abdelkader	0.0875
3	robvanderg	0.0851
6	**ShubhamKumarNigam**	**0.0778**

Table 13: Scores and leader-board ranks for subtask B (English).

5.5 Subtask C (English)

5.5.1 Datasets

Original: The train split of the original dataset for English in Table 27 in Appendix A.7 is splitted into train and validation datasets. As we do not have external datasets here, so we augmented the train split once with word embeddings (Original-Embedding) and once with repetition (Original-Repetition). We swapped between the tweet and its rephrase for half of the instances together with flipping the value of the sarcastic field, so that the model will be able to learn. Otherwise, it may always predict the first text as the sarcastic tweet and the second one as its non-sarcastic rephrase. The datasets are shown in Table 14.

Dataset	Total
Original-Train	606
Original-Validation	261
Original-Embedding	1606
Original-Repetition	1606

Table 14: Original datasets for subtask C (English).

5.5.2 Approaches

Organizers provide a sarcastic text, and its non-sarcastic rephrase, i.e., two texts convey the same meaning. Since the input format changed in this subtask, we input both texts together to the BERT model as one text separating them by the separating token. We focused on transformers which are trained on question-answering tasks. We tried BERT models trained on the **S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD) dataset (Rajpurkar et al., 2018). SQuAD is a reading comprehension dataset consisting of questions posed by crowdworkers on a set of Wikipedia articles.

We took models from the Hugging Face library; one is BERT large model (cased)[13], trained on whole word masking, and fine-tuned on the SQuAD dataset. Another is BERT base model (uncased)[14], trained on Masked language modeling (MLM), and fine-tuned on the SQuAD dataset.

5.5.3 Results

Metric: The main metric is the accuracy.

BERT: We used BERT large model (cased) on the original datasets, because it gave better results compared to the other models, using 15 epochs, 8e-6 learning rate, and the cross entropy as the loss function. The results are shown in Table 15. We did our submission using Type I preprocessing as it gave the best result on the validation dataset.

Dataset	Type	Val	Test
Original-Training	I	0.951	0.79
	II	0.9395	0.8
	III	0.9193	0.83
	IV	0.9135	0.79
Original-Embedding	I	0.9454	0.83
	II	0.9385	0.8
	III	0.9366	0.82
	IV	0.8588	0.72
Original-Repetition	I	0.9395	0.8
	II	0.9222	0.815
	III	0.9078	0.8
	IV	0.9078	0.68

Table 15: Accuracy of the BERT model for subtask C (English) on original datasets.

The official scores and leader-board ranks of the teams for subtask C (English) are shown in Table 16.

Rank	User	Accuracy
1	emma	0.87
2	lizefeng	0.855
3	leon14138	0.805
4	**ShubhamKumarNigam**	**0.79**

Table 16: Scores and leader-board ranks for subtask C (English).

5.6 Subtask C (Arabic)

5.6.1 Datasets

Original: The datasets were generated in the same way as in subtask C (Engligh) and are shown in Table 17

5.6.2 Approaches

The approached used here are similar to subtask C (English) except for the used transformers. Since the data is in the Arabic language, we took a couple of models from the Hugging Face library like the

[13] HuggingFace Bert-Large-Cased-Whole-Word-Masking-Finetuned-Squad

[14] HuggingFace Bert-Base-Uncased-Squad2

Dataset	Total
Original-Train	521
Original-Validation	224
Original-Embedding	1521
Original-Repetition	1521

Table 17: Original datasets for subtask C (Arabic).

multilingual model mBERT base (cased), trained on the Question Answering (QA) dataset in seven languages and fine-tuned on the combination of XQuAD (Artetxe et al., 2020) and MLQA (Lewis et al., 2020) datasets. We compared their performance to the CAMeLBERT-Mix model.

5.6.3 Results

Metric: The main metric is the accuracy.

BERT: We used CAMeLBERT-Mix model, because it gave better results among the other models, on the original datasets using 5 epochs, 3e-5 learning rate, and the cross entropy as the loss function. The results are shown in Table 18. We used Type II preprocessing for submitting the results as it is the best performing.

Dataset	Type	Val	Test
Original-Train	I	0.6242	0.5
	II	0.6242	0.5
	III	0.6711	0.72
	IV	0.3758	0.5
Original-Embedding	I	0.8792	0.825
	II	0.8792	0.845
	III	0.8691	0.845
	IV	0.7517	0.71
Original-Repetition	I	0.8993	0.855
	II	0.9262	0.87
	III	0.8658	0.86
	IV	0.6409	0.705

Table 18: Accuracy of BERT model for subtask C (Arabic) on original datasets.

The official scores and leader-board ranks of the teams for subtask C (Arabic) is shown in Table 19.

Rank	User	Accuracy
1	lizefeng	0.93
2	AlamiHamza	0.885
3	maryam.najafi	0.875
4	**ShubhamKumarNigam**	**0.87**

Table 19: Scores and leader-board ranks for subtask C (Arabic).

6 Analysis

6.1 Subtask A

English: There are 1400 instances in the test set out of which our model correctly classified 1060 instances (TP=155 and TN=905). There are bigger number of misclassified negative (NS) instances (FP=295) than the number of misclassified positive (S) instances (FN=45) and these are reflected in the precision (34.44) and the recall (77.5). This can be due to the fact that our model was trained on B4 dataset (59% S and 41% NS) taking F1-score for the sarcastic class as the metric for evaluation. This made the model focus more on identifying the positive instances sacrificing the considerable number of misclassified negative instances.

We dived into the details to see when the model predicted well and when it could not predict properly. We found that the majority of tweets which have specific punctuation marks like the exclamation mark were classified correctly. Short tweets were not classified properly which can be due to the insufficient information present in the tweets for classification. Interestingly, the existence of emojis highly increased the recall but not the F1 score and this is because sarcastic tweets can be easily recognized from their emojis but this does not apply on non-sarcastic ones. Also, tweets which include opposite emotions tend to be more sarcastic and to give better F1-score. Moreover, we found that tweets which contain misspellings or that need human knowledge to interpret can cause misclassification. Examples of all previous cases are presented in Table 20.

Arabic: There are 1400 instances in the test set out of which our model correctly classified 903 instances (TP=170 and TN=733). There are bigger number of misclassified negative (NS) instances (FP=467) than the number of misclassified positive (S) instances (FN=30) and these are reflected in the precision (26.69) and the recall (85). The number of instances which contain exclamation marks are less compared to English, and short tweets also were not classified properly for the same reason. Tweets that contain emojis were classified poorly and this is because we used Type II preprocessing which converted the emojis to text similar to the preprocessing of the model we used for English "BERTweet-large" and unlike the preprocessing of the model we used for Arabic "CAMeLBERT-Mix".

Example	Prediction
Exclamation Mark	
So you think the vaccine is a bad idea then. Glad you have that PHD in immunology!	TP
So many error codes in R today!	TN
Short Tweets	
Probably Jude mate	FP
Having the worst time on holiday	FN
Opposite Emotions	
Don't you just love Monday mornings, they are even better when its freezing cold and you have an uncooperative child too!	TP
Emojis	
Wow, can't wait to go into ANOTHER lockdown 🙄	TP
it doesn't need a consultation. Just ban it 🙃	FP
Human Knowledge	
Max Verstappen is such a clean driver, he never makes dirty moves when racing.	FN
Misspelling	
Boris has to bring in these restrictions he is dammed if he does and dammed if he doesn't. I live Boris.❤️❤️❤️❤️	FP

Table 20: Examples of the cases where tweets were classified correctly and incorrectly in subtask A (English).

6.2 Subtask B

Since the train and test datasets contain a very small number of instances of the understatement and overstatement tweets; therefore, the model could not identify any of them and the scores of them were zeros as shown in Table 21. For other labels, the model has far higher recall scores than precision scores which means the model performed well at identifying instances that belong to particular labels at the cost of mislabeling many instances.

	Precision	Recall	F1-score	Support
sarcasm	0.1335	0.8333	0.2301	180
irony	0.0708	0.75	0.1293	20
satire	0.0345	0.0204	0.0256	49
under-statement	0	0	0	1
over-statement	0	0	0	10
rhetorical_question	0.061	0.4545	0.1075	11

Table 21: Performance analysis for subtask B.

6.3 Subtask C

The model performed well in this subtask for both English and Arabic as shown in Table 22. This can be attributed to the nature of the task where the sarcastic tweet and its non-sarcastic rephrase of same meaning are given, besides the ability of the model to extract the relevant information for classification.

	Precision	Recall	F1-score	Support
English				
non_sarcastic	0.7576	0.8065	0.7812	93
sarcastic	0.8218	0.7757	0.7981	107
Arabic				
non_sarcastic	0.8936	0.84	0.866	100
sarcastic	0.8491	0.9	0.8738	100

Table 22: Performance analysis for subtask C.

7 Conclusion

The paper describes our participation in SemEval-2022 Task 6. The models used for sarcasm detection were mainly stand-alone transformers. In addition to this, we ran other experiments by stacking a BiLSTM layer with or without attention mechanism on top of the transformers. We created new datasets for each subtask by augmenting with external datasets, word embedding, or repetition. Our results shows that the augmented datasets enhanced the results for most subtasks. Moreover, we found that the fine-tuned stand-alone transformers gave the best results especially with Type II preprocessing. We also showed the enhancement when using a weighted loss function and the effect of using different learning-rates, epochs, and preprocessing types. We gave analysis of the performance of the models for each subtask, and revealed the possible cases that might have enhanced or deteriorated the performance. Finally, the rank of our team is consistent across most of the subtasks (the fourth rank) which shows the robustness of the used techniques.

References

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for Arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

L Bing. 2012. Sentiment analysis and opinion mining (synthesis lectures on human language technologies). *University of Illinois: Chicago, IL, USA*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

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.

Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. 2020. A report on the 2020 sarcasm detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 1–11, Online. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4238–4243, Reykjavik, Iceland. European Language Resources Association (ELRA).

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Silviu Oprea and Walid Magdy. 2020. 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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

A.1 Hierarchical Architecture

Figure 1 shows a hierarchical network based on a transformer. The input tokens are passed to the transformer, then the output token embeddings are passed to a Bi-LSTM layer which can be with or without attention mechanism.

A.2 Sarcasm Types Description

1. **Sarcasm:** tweets that contradict the state of affairs and are critical towards an addressee.

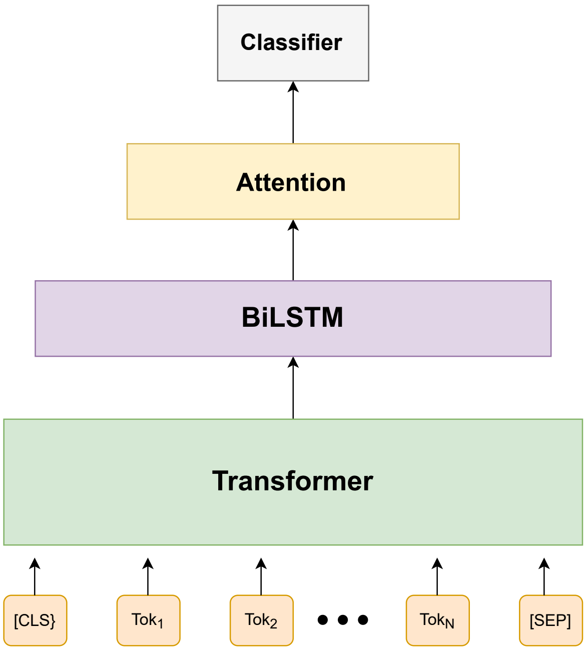

Figure 1: Hierarchical architecture.

2. **Irony:** tweets that contradict the state of affairs but are not obviously critical towards an addressee.
3. **Satire:** tweets that appear to support an addressee, but contain underlying disagreement and mocking.
4. **Understatement:** tweets that undermine the importance of the state of affairs they refer to.
5. **Overstatement:** tweets that describe the state of affairs in obviously exaggerated terms.
6. **Rhetorical question:** tweets that include a question whose invited inference (implicature) is obviously contradicting the state of affairs.

A.3 BERT Classification Architecture

Figure 2 shows the BERT-base classification architecture[15]. From the output of the final (12th) transformer, only the first embedding (corresponding to the [CLS] token) is used by a classifier.

A.4 ELECTRA:- Replaced Token Detection

ELECTRA (**E**fficiently **L**earning an **E**ncoder that **C**lassifies **T**oken **R**eplacement **A**ccurately) (Clark et al., 2020) replaces the MLM of BERT with Replaced Token Detection (RTD), which looks to be more efficient and produces better results. In BERT, the input is replaced by some tokens with [MASK] and then a model is trained to reconstruct the original tokens.

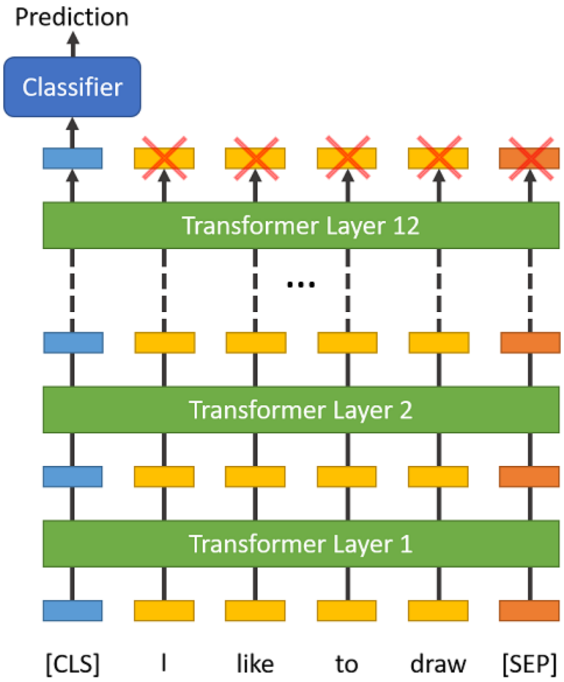

Figure 2: BERT classification architecture.

In ELECTRA, instead of masking the input, the adopted approach corrupts it by replacing some input tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original tokens, a discriminative model is trained that predicts whether each token in the corrupted input was replaced by a generator sample or not.

This approach trains two neural networks, a generator and a discriminator. Each one primarily consists of an encoder (e.g., a transformer network) that maps a sequence of input tokens into a sequence of contextualized vector representations. The discriminator then predicts whether it's fake by analyzing its data distribution.

A.5 Effect of Learning Rates and Epochs in Subtask A (English)

In Subtask A (English), we fine-tuned the BERT model on the B4 dataset with different learning rates and epochs as shown in Table 23 and Table 24 respectively. As shown in tables the best learning rate was 4e-6 and the best number of epochs was 5.

A.6 Performance of BERT+BiLSTM with and without Attention in Subtask A (English)

BERT+BiLSTM: The best results we got using this architecture were on the B4 dataset with 4e-

[15] BERT Fine-Tuning Tutorial with PyTorch by Chris McCormick and Nick Ryan

Val	Test	Learning Rate
0.4002	0.25	1 e - 5
0.5871	0.4558	9 e - 6
0.5913	0.4639	8 e - 6
0.4002	0.25	7 e - 6
0.4002	0.25	6 e - 6
0.5952	0.4883	5 e - 6
0.6025	0.4769	4 e - 6
0.5798	0.475	3 e - 6
0.5644	0.4724	2 e - 6
0.6012	0.5017	1 e - 6
0.4002	0.25	9 e - 7
0.4002	0.25	8 e - 7

Table 23: The effect of learning rates on the performance of the BERT model on the B4 dataset with weighted loss function using F1-score.

Val	Test	Epochs
0.5282	0.5304	1
0.5877	0.5092	3
0.6025	0.4769	5
0.6019	0.4647	7
0.5856	0.457	10
0.6017	0.5099	13
0.594	0.475	15
0.575	0.4943	17
0.5882	0.4675	20
0.5938	0.4633	23
0.575	0.4926	25

Table 24: The effect of epochs on performance of the BERT model on the B4 dataset with weighted loss function using F1-score.

6 learning rate, 10 epochs, and 50 LSTM hidden state size. Table 25 shows the results using same hyperparameters but with different hidden state sizes.

BERT+BiLSTM+Attention: The best results we got using this architecture were on the B3 dataset with 4e-6 learning rate, 5 epochs, and 600 LSTM hidden state size. Table 26 shows the results using same hyperparameters but with different hidden state sizes.

A.7 More Information about the Datasets:

Total number of tweets and percentage of sarcastic and non-sarcastic tweets in each dataset for subtask A is shown in Table 27, and the total number of tweets and percentage of sarcastic labels in each dataset for subtask B is shown in Table 28.

Hidden State Size	Val	Test
50	0.6027	0.4741
100	0.5882	0.4765
300	0.5813	0.4627
600	0.561	0.4702
900	0.5831	0.4516

Table 25: F1-score of the BERT+BiLSTM model for SubTask A (English) on the B4 dataset.

Hidden State Size	Val	Test
50	0.5726	0.4685
100	0.5978	0.468
300	0.5935	0.4627
600	0.6087	0.4625
900	0.5777	0.4618

Table 26: F1-score of the BERT+BiLSTM+Attention model for SubTask A (English) on B3 dataset.

Dataset	Split	Lang	Total	S%	NS%
Original	Train	En	3468	25	75
Original	Train	Ar	3102	24	76
Original	Test	En	1400	14.3	85.7
Original	Test	Ar	1400	14.3	85.7
Twitter API	Train	En	2841	16.8	83.2
Twitter API	Test	En	713	16.8	83.2
SemEval 2018	Train	En	3834	49.8	50.2
ArSarcasm-v2	Train	Ar	12548	17.3	82.7
ArSarcasm-v2	Test	Ar	3000	27.4	72.6

Table 27: Total number of tweets and percentage of sarcastic and non-sarcastic tweets in each dataset for subtask A.

External datasets for subtask A (English) is shown in Table 29, for subtask A (Arabic) is shown in Table 30, and for subtask B (English) is shown in Table 31.

Density of the number of words in tweets and their rephrases in the original datasets is shown in Figure 3 for English and in Figure 4 for Arabic.

In addition to this, information about dialects for Arabic subtasks is presented in Figure 5 and Table 32.

Information about the distribution of sarcastic and non-sarcastic tweets in the original datasets is presented in Figure 6.

Information about sarcastic labels, for subTask B, in the original datasets is shown in Figure 7.

Dataset	Split	Total	Sarcasm% Under- statement%	Irony% Over- statement%	Satire% Rhetorical question%
Original	Train	867	68.3	14.8	2.4
			1	3.8	9.7
	Test	1400	66.4	7.4	18.1
			0.4	3.7	4
Twitter API	Train	477	41.5	30.6	10.9
			1	8.2	7.8
	Test	120	41.7	33.3	11.7
			3.3	6.7	3.3

Table 28: Total number of tweets and percentage of sarcastic labels in each dataset for subtask B.

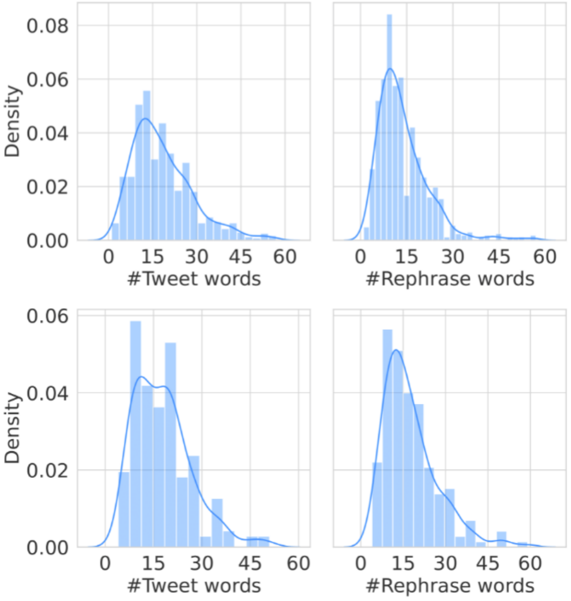

Figure 3: Density of the number of words in the original English train (top) and test (bottom) datasets.

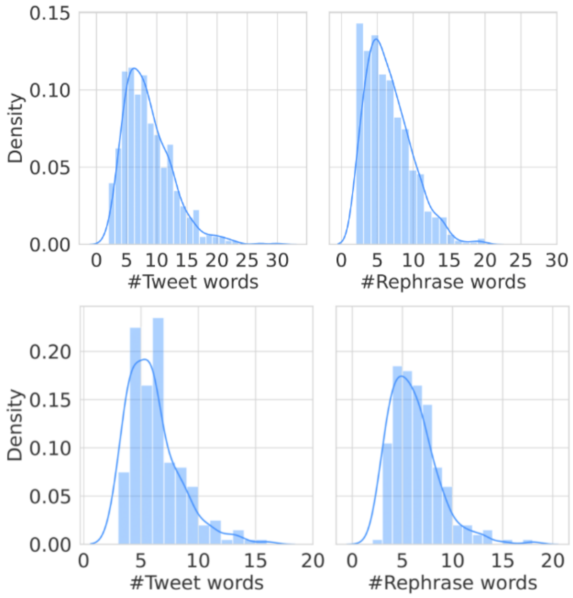

Figure 4: Density of the number of words in the original Arabic train (top) and test (bottom) datasets.

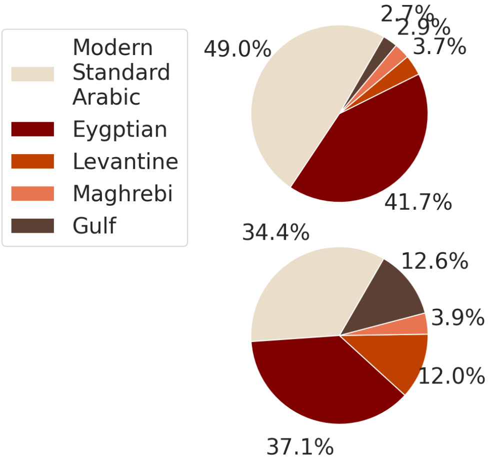

Figure 5: Percentage of dialects of tweets in the original Arabic train (top) and test (bottom) datasets.

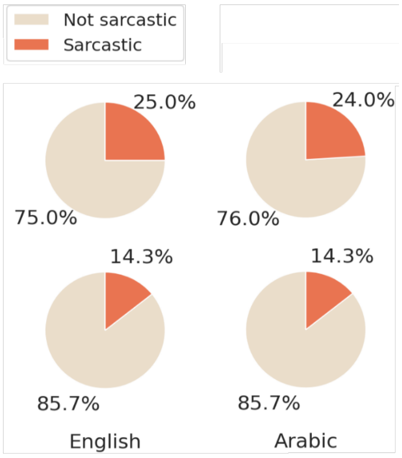

Figure 6: Percentage of sarcastic and non-sarcastic tweets in the original English and Arabic train (top) and test (bottom) datasets.

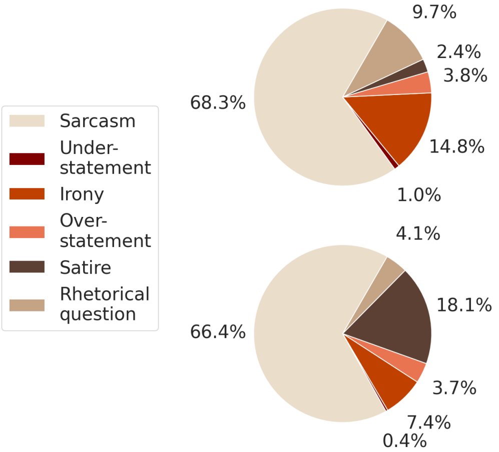

Figure 7: Percentage of tweets under each sarcastic label in the original English train (top) and test (bottom) datasets.

Dataset	Contributing Datasets	Additional Non Sarcastic Tweets from SemEval 2018-Train	Total	S%	NS%
B0	Original Train + Twitter API Train (only sarcastic) + Twitter API Test (only sarcastic) + SemEval 2018 Train (1911 sarcastic)	0	4578	66	34
B1		145	4723	64	36
B2		290	4868	62	38
B3		435	5013	60	40
B4		580	5158	59	41
B5		725	5303	57	43
B6		870	5448	55	45
B7		1015	5593	54	46
B8		1160	5738	53	47
B9		1305	5883	51	49

Table 29: External datasets for subtask A (English).

Dataset	Contributing Datasets	Additional Non Sarcastic Tweets from ArSarcasm-v2 Train	Total	S%	NS%
B0	Original Train + ArSarcasm-v2 Test (only sarcastic) + ArSarcasm-v2 Train (2168 sarcastic)	0	4850	71	29
B1		202	5052	68	32
B2		404	5254	65	35
B3		606	5456	63	37
B4		808	5658	61	39
B5		1010	5860	59	41
B6		1212	6062	57	43
B7		1414	6264	55	45
B8		1616	6466	53	47
B9		1818	6668	52	48

Table 30: External datasets for subtask A (Arabic).

Dataset	Balanced	Contributing Datasets	Total	Sarcasm Irony Satire	Under-statement Over-statement Rhetorical question
Ext-NB	Not Balanced	Original Train + Twitter API Train + Twitter API Test	1203	55.90%	1.20%
				22.30%	5.50%
				6.40%	8.70%
Ext-UW	Using Word Embedding		4336	16.50%	16.60%
				16.50%	16.80%
				16.60%	17%
Ext-UR	Using Repetition		4336	16.50%	16.60%
				16.50%	16.80%
				16.60%	16.90%
Ext-EB	Not Balanced		5314	15%	18.80%
				15.80%	17%
				16.70%	16.80%

Table 31: External datasets for subtask B (English).

Dataset	Split	Total	MSA	Eygptian	Levantine	Maghrebi	Gulf
Original	Train	3102	49	41.7	3.7	2.9	2.7
	Test	1400	34.4	37.1	12	3.9	12.6
ArSarcasm-v2	Train	12548	68.2	21.2	5	0.3	5.1
	Test	3000	77.4	10.2	1.6	0.1	10.7

Table 32: Distribution of tweets over dialects in Arabic Datasets.