Incorporating Count-Based Features into Pre-Trained Models for Improved Stance Detection

Anushka Prakash
School of Computer Science
University of Birmingham
United Kingdom
anushkaprakash6@gmail.com

Harish Tayyar Madabushi
School of Computer Science
University of Birmingham
United Kingdom
harish@harishtayyarmadabushi.com

Abstract

The explosive growth and popularity of Social Media has revolutionised the way we communicate and collaborate. Unfortunately, this same ease of accessing and sharing information has led to an explosion of misinformation and propaganda. Given that stance detection can significantly aid in veracity prediction, this work focuses on boosting automated stance detection, a task on which pre-trained models have been extremely successful on, as on several other tasks. This work shows that the task of stance detection can benefit from feature based information, especially on certain under performing classes, however, integrating such features into pre-trained models using ensembling is challenging. We propose a novel architecture for integrating features with pre-trained models that address these challenges and test our method on the RumourEval 2019 dataset. This method achieves state-of-the-art results with an F1-score of 63.94 on the test set.

1 Introduction and Motivation

In today's world, it is hard to imagine life without the internet and social media. The exponential growth of social media platforms over the past decade has provided people with an extremely convenient medium to access, consume and disseminate information. Social media has also become the primary source of news due to its ability to allow users to follow real-time updates (Huang et al., 2015). Unfortunately, this has also led to the spread of misinformation and propaganda. A recent example is that of Covid-19 which saw a plethora of misinformation online (Donovon, 2020).

Social media users often have trouble identifying credible resources on various platforms and are frequently guided towards misinformation, causing users to trust misinformation. With about 200 billion tweets posted every day (Krawczyk et al., 2017), manually processing this information to detect rumour can be extremely tedious, time consuming, prone to human error and bias and most importantly impractical. Thus, it is essential to develop efficient automated methods that help in detecting and tracking misinformation online to aid in the fight against misinformation.

There are four primary components to rumour determination: rumour detection, rumour tracking, stance classification, and veracity classification (Zubiaga et al., 2018). Stance classification aims at studying the stance of a reply post (referred to as an 'Opinion') on the source post (referred to as the 'Target' - to what the opinion is subjected). Studies have shown that posts containing false information have a notably higher number of 'denying' replies than the ones that do not (Derczynski et al., 2015). Hence, stance classification can play an important role in assessing the veracity of a rumour.

Given that stance classification can strongly aid in determining the veracity of a post, it is an important if challenging problem to solve. It remains challenging because posts are often sarcastic or witty, while also questioning the validity of the original poster's assumptions or evidence (Hasan and Ng, 2013).

1.1 Count-based Features and Pre-Trained models

Recent trends in Natural Language Processing (NLP) have seen the rise of pre-trained language models like GPT (Radford et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and so on.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom, pages 22–32
Barcelona, Spain (Online), December 12, 2020

These models are pre-trained on corpora containing millions of words and can be fine-tuned on several tasks to achieve state-of-the-art results. However, it is conceivable that these deep models can benefit from handcrafted features based on human intuition such as an extensive error analysis of these systems' results. For example, early work by Somasundaran and Wiebe (2010) on ideological stance detection observed that a simple unigram model outperformed more complex models using sentiment. This highlights that the stance of a statement is often related to the specific words used to express that opinion (for example, words such as 'Fake', 'Not true', 'disagree' are often used to deny a target claim), and stance classification systems could benefit from explicit features depicting such words. This is especially true when models are used to analyse social media data where elements such as hash tags could have special meaning that can be incorporated using handcrafted features. Furthermore, feature engineering can often be used to boost performances on specific classes that deeper models fail to perform well on. Hence, while it is evident that pre-trained language models have outperformed classical neural networks, simple models trained on hand crafted features can perform well on specific classes or instances that deeper models fail on.

Not only can models designed to analyse social media benefit from the use of handcrafted and count-based features but they are particularly important in handling disinformation and fake news as knowledge of the world at large and global events is required for improved tracking of disinformation. This non-linguistic information is likely to be available through count-based and handcrafted features but not in the pre-trained vector. This is also supported by prior work by Aker et al. (2017) and Bahuleyan and Vechtomova (2017), discussed in Section 2.

Given that handcrafted and count-based features can be used to improve the performance of deep learning models on specific classes or kinds of input, it is natural to ensemble these simple models based on handcrafted features with deeper models such as BERT. However, this is rather challenging for two important reasons:

1. Pre-trained models, such as BERT, are often trained for between 2 and 5 epochs during fine-tuning whereas simpler feature based models need to be trained for much longer. Our experiments show that a simple ensemble of these models results in over-fitting (Section 3.2) .

2. There are likely to be too many features to directly ensemble the raw features with pre-trained models (resulting in too much noise), a loss of important - task specific - information when using dimensionality reduction methods (Section 3.2), and too few output classes to use only the outputs of a feature based model in an ensemble (lack of information).

Therefore, this paper focuses on establishing methods of effectively integrating count-based features into pre-trained models such as BERT and RoBERTa, with specific regard to the identification of disinformation and propaganda on social media, where handcrafted and count-based features can be particularly helpful.

In this work, we formulate stance classification as a multi-class classification problem and experiment with various state-of-the-art NLP techniques to establish the best method of achieving this objective. We use the dataset provided for RumourEval 2019 (Task 7a of SemEval 2019) or the task of stance classification (Gorrell et al., 2019). While tweets have been extensively studied in the past due to their vast reach and brief nature (Krawczyk et al., 2017), this data set also includes posts from Reddit, so focusing on a broader range of social media data. Each conversation thread consists of a source post which states or discusses a rumour. This source post is followed by a set of replies (and replies to these replies) which either 'Support', 'Deny', 'Query' or 'Comment' on the source text (See Figure 1). The source post is also labelled depending on the stance it takes at its (hidden) target (Gorrell et al., 2019).

So as to ensure reproducibility and enable other researchers to build upon this work, we make our program code, hyperparameter details and models available [1].

[1] https://github.com/Anushka-Prakash/RumourEval-2019-Stance-Detection/

2 Related Work

The effectiveness of stance detection in identifying and tracking misinformation online has led to significant research in this area. This section provides an overview of work related to this study.

Early work on studying perspectives such as that by Lin et al. (2006) use Naïve Bayes and SVM based classifiers to show that word usage alone can help in determining the point of view of a document or sentence. In a similar study, Kim and Hovy (2007) go beyond word usage and exploit the lexical patterns used by people while expressing an opinion. Somasundaran and Wiebe (2010) showed that a unigram based model outperformed a sentiment-based model for determining stance thus reinforcing the importance of words in detecting stance. This shows that the stance of an opinion could be related to the kind of words used to express it. Although working on a different problem set (focused on binary classification) and using different datasets, the above-mentioned studies are relevant to this work because they emphasise on the importance of words used to express an opinion to determine their stance.

Bahuleyan and Vechtomova (2017) hypothesise the presence of certain words in the reply text or the opinion as an indicative measure of its stance. They use these and some tweet specific features like the word count of the tweet to train on a gradient boosting classifier and show that stance classifiers could benefit from these topic independent features. Aker et al. (2017) explore a novel set of features and claim that simpler classifiers with profuse feature knowledge can outperform several complex machine-learning techniques in stance detection. Riedel et al. (2017) use a Multi Layer Perceptron (MLP) having a single hidden layer with term frequency and TF-IDF vectors as inputs to perform at par with several complex models. These works lend weight to our hypothesis that handcrafted and count-based features can significantly aid in the task of stance detection.

Recent work in stance classification shows extensive use of deep-learning models to achieve state-of-the-art results. Fajcik et al. (2019) and Yang et al. (2019) who worked on the same task and dataset as this work, use the BERT and GPT architectures respectively for stance classification. Fajcik et al. (2019) use an ensemble model wherein they combine outputs from several BERT models to increase the F1-score. Their system achieves a significantly high F1-score without the use of any hand-crafted features. Yang et al. (2019), who achieved state of the art results on this dataset prior to this work, use an inference chain based system that was fine-tuned on GPT. They use handcrafted features such as the presence of question marks, URLs, positive and negative words, etc. and leverage the structure of a conversation thread to achieve state-of-the-art results in stance classification.

2.1 Class-imbalance

Since most data sets based on real-world data (including the stance detection data set used in these experiments) suffer from class-imbalance, we explore recent methods of addressing this issue. One method of addressing class imbalance is the use of a two-step classifier, such as work by Wang et al. (2017). They first classify the tweets as 'comment' and 'non-comment' and use a second classifier to distinguish non-comments as 'Support', 'Deny', or 'Query'. Krawczyk et al. (2017), on the other hand, propose a one-vs-one decomposition of the multi-class problem before then combining the outputs of these binary classifiers using a weighted approach to rebuild the multi-class problem. Yang et al. (2019), on the other hand, use examples from similar data sets to increase the training data for minority classes.

The other method of handling class-imbalance is by use of cost-weighting. Tayyar Madabushi et al. (2019) apply cost-sensitive learning for the task of Fine-Grained Analysis of Propaganda in News Article (Da San Martino et al., 2019) wherein they first test the similarity between training and validation sets using Wilcoxon signed-rank test. They also show that cost-weighting (increasing the cost of a minority class) can be more beneficial when applied to dissimilar datasets. Fajcik et al. (2019) also use cost-weighting to tackle class-imbalance.

3 Experimental and Model Design

This section describes initial exploratory experiments and the process behind building a model that address the two obstacles to integrating features into pre-trained models mentioned in Section 1.1, namely: **a)** the significant difference in the number of epochs required by pre-trained models and feature based

models, and **b)** the fact that using raw features might contain too much noise in an ensemble, the outputs alone might contain too little information, and dimensionality reduction methods result in the loss of task specific information.

3.1 Pre-Processing and Experimental Setup

The pre-processing steps used for all our experiments were consistent. URLs and mentions present in all the source and the reply posts were replaced with the special tokens 'URL' and '$MENTION$' respectively. This was adapted from Fajcik et al. (2019) and found to be helpful in our experiments.

We use two sequences as input to pre-trained model, which represent an opinion (the 1st sequence) and its target (the 2nd sequence). It is the 1st sequence, the opinion, that we are interested in classifying. This first sequence consists of the reply being classified and, in instances where such a post is itself a reply to another reply, is concatenated with its 'parent'. Figure 1 illustrates the tree structure of a conversation thread from the dataset.

Inputs to the pre-trained models were generated as follows: **(1)** While classifying the source post of a conversation thread (TE-1 in Figure 1), the text of this source post is the first input sequence (the one being classified). Since this post is not a reply and the target information is not explicitly available, the second input sequence is empty. **(2)** When classifying a reply (for example, TE-2 in Figure 1), the text of this reply is the first input sequence and the source post of the conversation thread becomes the second sequence. **(3)** While classifying a nested reply (reply to a reply, such as TE-3 in Figure 1), the text of this nested reply is concatenated with the text of its parent and used as the first input sequence, and the source post of the entire conversation thread becomes the second sequence. This method of structuring the input was adapted from work by Fajcik et al. (2019) who hypothesise that the stance of a given post depends on itself, its previous post and the source post of the conversation thread.

We use the *encode_plus* function provided by HuggingFace in their re-implementation of RoBERTa to encode the input sequences. The input sequences are first appended with special tokens - RoBERTa uses special tokens to indicate the start of the input ($< s >$), the end ($< /s >$) and two end tags to differentiate between different input sequences ($< /s >< /s >$). Figure 1 illustrates how special tokens were embedded in each example. These sequences are then encoded using Byte-Pair encoding and fed to RoBERTa.

3.2 Exploratory Experiments

Initial experiments using BERT yielded an F1-score of 0 for the class 'Deny', one of the harder to predict labels (See table 1). RoBERTa, however, produced an F1-score greater than 0 for all 4 classes, which led us to pick RoBERTa over BERT. We pre-process each example and feed two sequences as input to RoBERTa (as explained in section 3.1).

Next, we attempt to find a set of features that help in stance classification. As discussed in Section 1.1, work by Somasundaran and Wiebe (2010) on ideological stance classification showed that a unigram model can outperform more complex models. While their study was on a different problem set and focused on binary classification ('for' and 'against') of ideological stance, they showed that the stance of an opinion could be related to the kind of words used to express it. This is particularly true of a the 'Deny' class in the data set we work with. As discussed in section 1.1, words such as 'Not true' and 'Disagree' are frequently used to deny a claim.

We train a Multi Layer Perceptron (MLP) using TF-IDF features as input. We use a single hidden layer consisting of 128 hidden units, a tanh activation function and a final linear layer with a softmax function to make predictions. We use a learning rate of 0.02 and train this model for 55 epochs (Additional details such as the complete list of hyperparameter are available as part of the program code released). This model achieves an F1-score on 'Deny' class greater than that achieved using the RoBERTa-base (See table 1).

These results indicate that the use of TF-IDF features was particularly beneficial in classifying 'Deny'. In their recent work, Lim and Tayyar Madabushi (2020) incorporate TF-IDF features with BERT to improved performance on Offensive Language Identification in Social Media (Zampieri et al., 2020).

Source Post (TE-1): Darren Wilson is a six year veteran of the #Ferguson Police and had no disciplinary actions against him. **[Support]**

Reply 1 (TE-2): Can we see video proof **[Query]**

Reply 2 (TE-3): HE ISN'T THE SHOOTER RT [MENTION] **[Comment]**

Reply 2.1 (TE-4): [MENTION] well who is #Ferguson **[Comment]**

TE-1: $< s >$ Darren Wilson is a six year veteran of the #Ferguson Police and had no disciplinary actions against him. $< /s >< /s >< /s >$

TE-2: $< s >$ Can we see video proof $< /s >< /s >$ Darren Wilson is a six year veteran of the #Ferguson Police and had no disciplinary actions against him. $< /s >$

TE-3: $< s >$ HE ISN'T THE SHOOTER RT [MENTION] $< /s >< /s >$ Darren Wilson is a six year veteran of the #Ferguson Police and had no disciplinary actions against him. $< /s >$

TE-4: $< s >$ [MENTION] well who is #Ferguson HE ISN'T THE SHOOTER RT [MENTION] $< /s >< /s >$ Darren Wilson is a six year veteran of the #Ferguson Police and had no disciplinary actions against him. $< /s >$

Figure 1: An example of a conversation thread and associated labels (in bold at the end of each post) from the RumourEval data set showing the tree like structure. The second part shows the training examples (TE) constructed for use in pre-trained models from each of the original posts. Further details in text.

This shows that tasks like stance classification and abuse detection greatly benefit from information pertaining to the words used in expressing that emotion.

Model	**Macro F1-Test**	**F1 Support**	**F1 Deny**	**F1 Query**	**F1 Comment**
MLP Model	0.38	0.18	0.24	0.22	0.87
BERT	0.46	0.44	0.0	0.52	0.90
RoBERTa-base	0.51	0.46	0.14	0.52	0.91
RoBERTa-base + All TF-IDF Features	0.49	0.45	0.06	0.53	0.91
RoBERTa-base + PCA Transformed TF-IDF	0.56	0.34	0.46	0.54	0.89
RoBERTa-base + Output of MLP	0.51	0.46	0.19	0.50	0.91

Table 1: Achieved results on the test set for Exploratory Models

Given the improved performance of the MLP on 'Deny', we aim to create an ensemble of the MLP and a pre-trained model so as to boost the overall performance. We ensemble the pooled output of RoBERTa with the output of a Multi-Layered Perceptron (MLP) (consisting of 4 units, one for each class). This combination is then connected to a linear layer followed by a softmax function to make predictions. While this model performed slightly better on the 'Deny' class, it performed worse on the class 'Query', thus achieving an overall Macro F1 score exactly equal to RoBERTa Base. We also create an ensemble model combining all the TF-IDF features with RoBERTa which we train for 4 epochs. This model performs worse than RoBERTa-base.

In order to verify that this decrease in performance is related to the TF-IDF features being high di-

mensional and noisy, we conduct experiments wherein we apply Principal Component Analysis (PCA), to reduce the dimensions of the TF-IDF vector to a length of 128 before using this shortened vector in the ensemble. We train this ensemble model for 40 epochs and note an increase in performance over the first 4 epochs and a subsequent increase in loss (with no change in accuracy) indicating that the model overfits. We similarly train the ensemble of RoBERTa and the MLP for 20 epochs and find that the model dramatically overfits with a drop in F1 from 0.51 to 0.15.

This section described experiments on six different models (See Table 1 for comparative results, we use the macro-averaged F1-score to compare models). Our finds are that **a)** the use of an MLP can be beneficial in boosting performance, **b)** the use of all tf-idf features in an ensemble is too noisy, **c)** the output of the MLP alone contains too little information to be useful in an ensemble, and **d)** ensembles of pre-trained models and an MLP will suffer from over-fitting before the MLP can be trained sufficiently. Additionally, training an ensemble of an MLP and RoBERTa takes about 10 min per epoch thus dramatically increasing the training time when we train for 20 or 40 epochs.

We address these shortcomings by introduction of a novel architecture that makes use of an *already trained MLP* which prevents overfitting, reduces train time while also ensuring the effective integration of pre-trained models and handcrafted features. The next Section (3.3) describes this model.

3.3 Model Architecture

This section describes the novel ensemble architecture that was used to incorporate count-based features, particularly useful for the classification of posts belonging to 'Deny', while simultaneously addressing the difficulties in ensembling feature based models and pre-trained models.

Figure 2 provides an illustration of the proposed architecture. This ensemble architecture uses two novel elements to address the two difficulties in creating an ensemble of the feature based and pre-trained models. Specifically **a)** the MLP used in the ensemble is one that *is already trained and optimised* (hyperparameters) for this task, and **b)** the output of the *hidden layer* of this already trained MLP is used in ensembling with pre-trained models instead of either the input or the final output.

The use of an already trained MLP in the ensemble ensures that the MLP does not underfit when trained in an ensemble with pre-trained models for a small number of epochs. This method, while similar to pre-training is not exactly the same as it is trained on the same task unlike in pre-training. The use of the hidden layer in the ensemble simultaneously deals with the problem of having too much noise in the input and too little information in the output by providing an abstract layer of condensed information from the MLP as input to the final ensemble model.

More concretely, this ensemble model is constructed as follows: The pooled output from RoBERTa (vector of length 768) is concatenated with the output of the hidden layer (vector of length 128) of an already trained MLP. This combined vector, having a length of 896, is further connect to a linear layer and a softmax function to make predictions (See Figure 2). We train this ensemble model for 6 epochs with a learning rate of $2e^{-6}$ and batch size 4. Further details on this model are available in the program code released as part of this work. We also test on ensembling the output of an already trained MLP instead of the hidden layer, and while it does do better than RoBERTa alone, it does not do as well as an ensemble of the hidden layer.

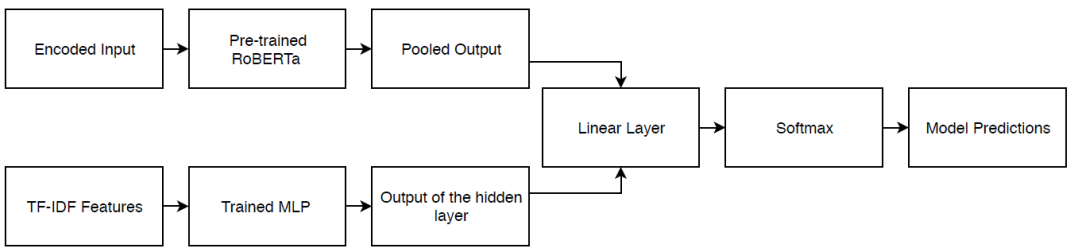

Figure 2: Architecture of the Proposed Model which combines output from RoBERTa and the hidden layer of an already trained MLP model.

This ensemble model outperforms RoBERTa along with all previously used ensemble models. We find this result, presented in Table 2, to be consistent across both RoBERTa-base and RoBERTa-large.

To address the possibility that our model was caught in local optima, we train each of our models 5 times using different seeds and evaluate them using the development set. We then use the model that performs the best on the development set to classify the test set and report results.

3.4 Tackling Class-imbalance

Class-imbalance implies that the number of training examples of one or more classes are significantly lower than the other classes. The RumourEval data set suffers from this same problem (Support 13.9%, Deny 6.6%, Query 4.8% and Comment 72.4%), and the 'Comment' class is not only dominant in the data set but is also the least helpful in detecting rumour (Gorrell et al., 2019). While there are several ways of handling class-imbalance (as discussed in section 2.1), we make use of cost-weighting due to its flexibility and proven performance.

Cost-weighting consists of assigning a higher weight to a minority class which holds more relevance. The cost function is thus modified to accommodate the weights assigned to each class - the more the weight, the more the model is penalised for inaccurate predictions on that class. The cost weights for our models were empirically determined via experiments and significantly boosted performance. We incorporate cost-weighting in all our experimented models.

4 Results and Analysis

This section presents an overview of our experiments and findings. We compare our results with prior state-of-the-art work on stance classification which makes use of this data set. We also perform an error analysis of our proposed model and report findings.

4.1 Results

Table 2 presents the macro-averaged F1-score we obtained by our proposed model (an ensemble model combining RoBERTa's pooled output with the output of the hidden layer of an already trained MLP). We also present results from both RoBERTa-base and RoBERTa-large to draw attention to how our proposed method significantly improves performance on both these versions of RoBERTa. For each experiment, we also report the macro-averaged F1-score and the F1-scores for all classes. These results show how how our proposed architecture significantly improves accuracy on the previously under-performing class 'Deny', which our count-based features were designed to improve.

Model	**Macro F1-Test**	**F1 Support**	**F1 Deny**	**F1 Query**	**F1 Comment**
RoBERTa-base	0.51	0.46	0.14	0.52	0.91
RoBERTa-base + Hidden layer output of Trained MLP	**0.58**	**0.43**	**0.39**	**0.58**	**0.92**
RoBERTa-large	0.57	0.43	0.42	0.54	0.92
RoBERTa-large + Hidden layer output of Trained MLP **(Proposed Model)**	**0.64**	**0.48**	**0.55**	**0.60**	**0.93**

Table 2: Results of our Proposed architecture on the test set using RoBERTa-base and RoBERTa-large showing a consistent improvement in performance across both versions of RoBERTa.

Table 3 presents our results alongside top performing systems on this dataset. We compare our results with the winning teams of SemEval 2019 for the same task. A branch-LSTM based system was used as a baseline for the stance classification task of SemEval 2019. We note that the model proposed in this work achieves state-of-the-art results on this dataset.

System	Details	Macro Averaged F1-Score (%)
This work		**63.94**
Yang et al. (2019)	1st Rank - SemEval 2019	61.87
Fajcik et al. (2019)	2nd Rank - SemEval 2019	61.67
Kochkina et al. (2017)	Baseline - SemEval 2019 (Winning method, SemEval 2017)	49.3

Table 3: A comparison of this work with prior state-of-the-art methods on the same task and data set.

4.2 Discussion

Previous sections provided details of the various models we experimented with, along with their results on the test set. In this section we present an overview of our intuitions behind and observations from these experiments.

Our initial experiments showed how a feature based model can outperform pre-trained models *on specific classes*. This is a rather intuitive result which could potentially be true of various tasks that benefit from handcrafted and count-based features. We reiterate that tasks related to disinformation, propaganda and rumour are particularly capable of benefiting from handcrafted and count-based features such as user information, hashtags, URLs, word frequencies, and so on, as these features provide additional information that can be useful in classification and possibly unavailable to pre-trained models pre-trained on corpora that are different from social media and fine-tuned on the task over very few epochs.

We then presented two significant challenges (Section 3.2) that are faced in creating an ensemble of features and pre-trained models such as BERT or RoBERTa. We addressed the first challenge of too many or too few input features to be combined with pre-trained models by using the output of the MLP's hidden layer. We then address the second challenge - the difference in the number of training epochs required by feature-based and pre-trained models - by use of an already trained MLP.

The experiments with the ensemble model using RoBERTa's pooled output and the output of the trained MLP model showed a significant increase in model performance on stance classification. We show how we address each challenge through our experiments and propose a novel architecture that utilises abstract information from a trained MLP to increase the performance of pre-trained models.

In summary, this work shows that stance classification systems benefit from features that depict the kind of words used to express an opinion by ensembling TF-IDF features with RoBERTa. We also propose a novel architecture to effectively integrate a feature engineered model with pre-trained deep learning models so as to significantly boost performance on tasks that benefit from handcrafted features (such as stance classification).

4.3 Error Analysis

We perform an extensive error analysis of our proposed model. First, we study the impact of availability of trained features to RoBERTa for stance classification. As depicted from the confusion matrix (Table 4), the ensemble model using the abstract features from the trained MLP model specifically improves the performance of 'Support', 'Deny' and 'Query' classes. Our model performs exceedingly well on the 'Deny' class, which previous models have typically struggled with. However, this comes at the cost of misclassifying some other classes ('Support' and 'Comment') as 'Deny', it increases the overall F1-score. This observation is detailed further using examples from the data set.

An analysis of the predictions made by the model on the test set shows that it learns to associate words such as 'Fake', 'not' and 'not true' with the 'Deny' class. Again, this shows that effective integration of features by use of trained MLP models can significantly help in the classification of output classes that benefit from such features. While this was helpful in the classification of most examples from the 'Deny' class, some posts that used the aforementioned words were 'comments' and misclassified as 'Deny' (Refer Note 1 from Table 5). Similarly, some 'comments' that contained interrogative words

True label	Predicted label			
	Support	Deny	Query	Comment
Support	**53 (46)**	4 (1)	1 (1)	99 (109)
Deny	1 (1)	**52 (31)**	3 (2)	45 (67)
Query	14 (11)	6 (4)	**52 (47)**	21 (31)
Comment	1 (1)	26 (12)	22 (31)	**1427 (1432)**

Table 4: Confusion matrix for RoBERTa-large and RoBERTa-large + Trained MLP models. Values associated with RoBERTa-large are in brackets.

and question marks were classified as 'Queries' without taking the content of the post into consideration (Refer Note 2 from table 5). This is interesting as it reintroduces the problems encountered by earlier models - problems that were, to some extent, addressed by contextual pre-trained models, showing the need for a careful balance between these two approaches.

Note	Reply-post	Target	Predicted label	True label
1	*$MENTION$ If this is fake, the poster should be charged with spreading mass panic*	*This is crazy! #CapeTown #capestorm #weather-forecast URL*	Deny	Comment
2	*Why would I need a kettle?*	*Is it true most Americans don't own a kettle? If so, why not?*	Query	Comment

Table 5: Error Analysis of predictions by the RoBERTa + trained MLP ensemble model on the test data. While the model learns to associate words like 'Fake' and 'why' with the 'Deny' and 'Query' classes respectively, it gives too much weight to these features, resulting in errors.

The model also finds it challenging to distinguish between a 'support' and a 'comment' with respect to a post - a problem that could be attributed these two classes being somewhat similar. The model is also not able to classify posts that contain a URL, possible due to the missing information from the URL. Finally, the model also often inaccurately predicts the labels of the source post. We believe that this is due to the explicit unavailability of target information.

5 Conclusions and Future Work

This work introduced a novel architecture that makes use of abstract information from an already trained MLP to boost RoBERTa on the task of stance detection. Our technique achieves state-of-the-art results with an accuracy of 86.69% and a macro-averaged F1-score of 63.94 on a standard data set. We conclude that an effective integration of features with models such as RoBERTa can significantly increase model performance on tasks that benefit from such features and that our architecture is an effective solution.

In future work we aim to integrate entire conversational threads to study its impact of such information as, in this work, we limit the information pertaining to one example to reply, previous and the source post of the conversation thread. We also aim to use these methods on other related propaganda and abuse datasets along with exploring the impact of other handcrafted features.

Acknowledgements

We would like to thank the NVIDIA Deep Learning Institute for the provision of AWS credits which we used to access GPU resources in this work.

References

A. Aker, Leon Derczynski, and Kalina Bontcheva. 2017. Simple open stance classification for rumour analysis. In *RANLP*.

Hareesh Bahuleyan and Olga Vechtomova. 2017. UWaterloo at SemEval-2017 task 8: Detecting stance towards rumours with topic independent features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 461–464, Vancouver, Canada, August. Association for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China, November. Association for Computational Linguistics.

Leon Derczynski, Kalina Bontcheva, Michal Lukasik, Thierry Declerck, Arno Scharl, Georgi Georgiev, Petya Osenova, Toms Pariente Lobo, Anna Kolliakou, Robert Stewart, et al. 2015. Pheme: Computing veracity—the fourth challenge of big social data. In *Proceedings of the Extended Semantic Web Conference EU Project Networking session (ESCW-PN)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Joan Donovon. 2020. Covid hoaxes are using a loophole to stay alive—even after content is deleted. *MIT Technology Review*.

Martin Fajcik, Pavel Smrz, and Lukas Burget. 2019. BUT-FIT at SemEval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1097–1104, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Y. Linlin Huang, Kate Starbird, Mania Orand, Stephanie A. Stanek, and Heather T. Pedersen. 2015. Connected through crisis: Emotional proximity and the spread of misinformation online. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, page 969–980, New York, NY, USA. Association for Computing Machinery.

Soo-Min Kim and Eduard Hovy. 2007. Crystal: Analyzing predictive opinions on the web. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1056–1064, Prague, Czech Republic, June. Association for Computational Linguistics.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 task 8: Sequential approach to rumour stance classification with branch-LSTM. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 475–480, Vancouver, Canada, August. Association for Computational Linguistics.

Bartosz Krawczyk, Bridget T McInnes, and Alberto Cano. 2017. Sentiment classification from multi-class imbalanced twitter data using binarization. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 26–37. Springer.

Wah Meng Lim and Harish Tayyar Madabushi. 2020. Uob at semeval-2020 task 12: Boosting bert with corpus level information. *ArXiv*, abs/2008.08547.

Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on? identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 109–116, New York City, June. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, page 116–124, USA. Association for Computational Linguistics.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China, November. Association for Computational Linguistics.

Feixiang Wang, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 task 8: Rumour evaluation using effective features and supervised ensemble models. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 491–496, Vancouver, Canada, August. Association for Computational Linguistics.

Ruoyao Yang, Wanying Xie, Chunhua Liu, and Dong Yu. 2019. BLCU_NLP at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1090–1096, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020).

A. Zubiaga, A. Aker, Kalina Bontcheva, Maria Liakata, and R. Procter. 2018. Detection and resolution of rumours in social media. *ACM Computing Surveys (CSUR)*, 51:1 – 36.