Combating Fake News with ULMFiT

Team B
Prajwal Vijendra, Zisheng Jason Chang,
Ana Belen Barcenas Jimenez, Jingwen Wang

April 2019

Contents

Abstract

Identifying fake news is a complicated and challenging task. An important first step of checking a piece of news is fake or not is to determine the stance different news contents take towards the assertion.The Fake News Challenge Stage 1 (FNC-1) held in 2017 addressed to this task as stance detection. In this paper, we present our stance detection system which utilize the cutting-edge model in Natural Language Processing, ULMFit, with both qualitative and quantitative results showing that our system outperforms the winner of the FNC-1.

1 Introduction

Fake news has a negative impact on online media. People are baited by misleading headlines to click on links to irrelevant content. The headlines and contents of the article can even disagree with each other. Regardless of its type, fake news can result in misleading users. With social media today, these stories can be amplified to a much broader audience. The first step in combating fake news is to classify fake news articles. However, it takes a huge amount of human resources to classify all articles manually. With the advancement of Artificial Intelligence, researchers have been trying to take advantage of the technology to help tackle with fake news detection. The FNC-1 [**fake_news_challenge**] was hosted by more than 100 volunteers from academia and industry, aiming to encourage participants to come up with ideas to design machine learning models that could be used to combat fake news. The goal of this challenge was to focus on stance detection. The self described goal by FNC-1 is to "[...] address the problem of fake news by organizing a competition to foster development of tools to help human fact checkers identify hoaxes and deliberate misinformation in news stories." The first iteration of the challenge was from December 1s 2016 to June 2nd 2017, a crucial step to helping detect fake news.

2 Background

Fake news detection is a complicated task involving political and technical issues to be taken into consideration. In the context of the FNC-1, the goal of this project is to tackle fake news with a more robust solution rather(stance detection) than expansive(classifying fake news). Stance detection is a crucial building block for a variety of tasks, such as understanding the structure of essays[**DBLP:journals/corr/StabG16**], analyzing contents of online debates[**jointModelsOfDisagreement**] and determining the veracity of rumors on twitter[**simpleopenstance**]. We utilize stance detection to determine if the title of a article is related to the content, and if related, whether the content agrees with, disagrees with, or discusses the headline. The four outputs of the model are: "agree," "disagree," "discuss," and "unrelated." The top-performing model in FNC-1 is called SOLAT in the SWEN by Talos Intelligence

[**largent_1970**]. They implemented some of the cutting-edge models to compare their performances and won the competition by using an ensemble model which takes 50/50 weighted average between gradient-boosting decision tree and deep convolutional neural network. Team Athene [**athene**] won second place with a feature-rich stacked LSTM model.

Universal Language Model Fine-tuning (ULMFiT) is an effective transfer learning Document–headline pairs of 200 reserved for training, the remaining document–headline pairs of 100 topics for testing. Topics, headlines, and documents are therefore not shared between the two data splits. To prevent teams from using any unfair means by deriving the labels for the test set from the publicly available Emergent data, the organizers additionally created 266 instances. The figure below describes the statistics of the dataset.method that can be applied to any task in NLP. Here, we take advantage of the state-of-the-art ability in text classification ULMFiT to solve the stance detection problem.

3 Data

The dataset contains 300 topics. The topics are represented by claims with 5–20 news article documents each. The dataset is derived from the Emergent project (Silverman, 2017) which addressed rumor debunking. In the project, each news article document was summarized into a headline that reflects the stance of the whole document. The FNC-1 organizers match each document (d) with every summarized headline (h) and then label the (d,h) pair with one of the four stance labels S (Unrelated, Agree, Disagree and Discuss). To generate the unrelated class UNR, headlines and documents belonging to different topics are randomly matched.

3.1 Training Dataset

The training dataset has 49972 records and the percentage distribution of the target classes are shown in the below table.

Rows	Unrelated	Discuss	Agree	Disagree
49972	0.73131	0.17828	0.0736012	0.0168094

3.2 Testing Dataset

The testing dataset has 25413 records and the percentage distribution of the target classes are shown below in the table

Rows	Unrelated	Discuss	Agree	Disagree
25413	0.72203	0.17565	0.07488	0.02742

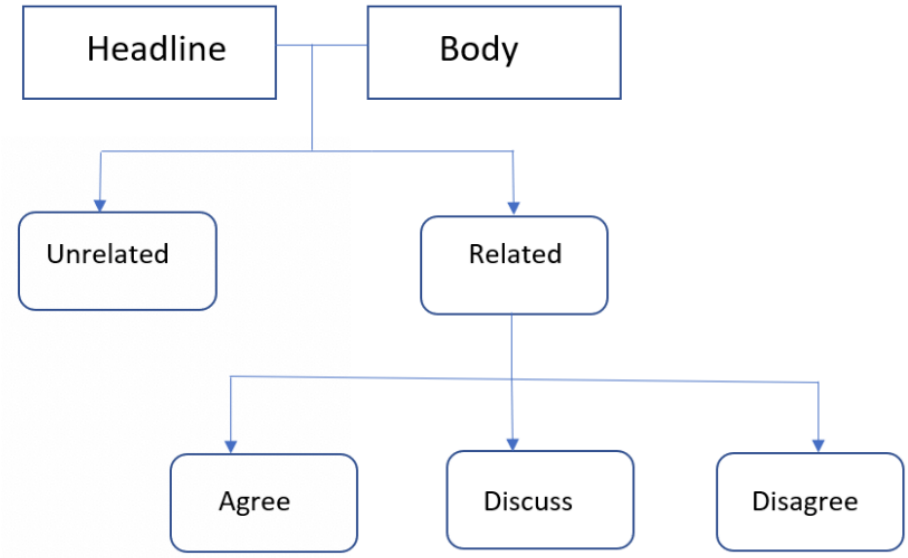


Figure 1: Overview of headline and body relationship.

Headline	**"Robert Plant Ripped up $800M Led Zeppelin Reunion Contract"**
Agree	"... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup."
Disagree	"...No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together."
Discusses	"...Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal."
Unrelated	"...Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today."

Table 1: Example of a headline and text bodies.

4 Methods

We developed a two-staged classification model to classify the Headline and Body text into Agree, Disagree, Discusses or Unrelated classes. In the first stage, we compute the cosine similarity score to classify the records into "Related" and "Unrelated" based on a threshold. In the second stage, we developed a text classification model based on Universal Language Model Fine-Tuning to classify the "Related" records into Agree, Disagree or Discuss classes. We trained the model using the training dataset which is detailed above, and tested the model performance on the testing dataset.

4.1 Compute Similarity

We compute the similarity between the headline and the body text using Cosine Similarity. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between

them. To apply cosine similarity, we first need to process the text to include only meaningful words and remove stopwords. We do this by employing Part-of-Speech tagging from SpaCy on the Headline and the Body text. We include only the words tagged as 'Pronoun', 'Noun', 'Symbol', and 'Number' and remove the rest of the words. We then vectorize the headline and body words and compute the cosine similarity between the two vectors. If the Headline and the Body vectors are dissimilar (oriented at 90 degrees relative to each other) then the cosine similarity score is 0. In contrast, if the Headline and the Body vectors have same orientation, then the cosine similarity score is 1.

4.2 Classify Related VS Unrelated

To classify the records as "Related" or "Unrelated", we use the computed cosine similarity score. If the cosine similarity score between the Headline and Text is less than threshold 0.10 then we classify the record as "Unrelated". If not, we classify the record as "Related" and send it for additional classification.

4.3 Classify Agree, Disagree, Discuss

To further classify the record into Agree, Disagree or Discusses, we use a text classification model called Universal Language Model Fine-tuning [**ULMFiT**]. There are three parts to the model:

4.3.1 General Language Model Pretraining

We used the weights of the Language Model pre trained on Wikitext-103 [**merity**] which consists of 28,595 preprocessed Wikipedia articles and 103 million words. Pre Training enables generalization even to small dataset.

4.3.2 Target Language Model fine-tuning

We fine-tune the Language model on our dataset (News Headline and Body). Since the Language Model is already pretrained, the model converges faster as it has to only learn the idiosyncrasies of our dataset. Here we employ slanted triangular learning rate (STLR) to finetune the model. In STLR, the learning rate first linearly increases and then linearly decays which is found to be key for good performance.

4.3.3 Target Task Classifier fine-tuning

We train a text classifier using the Language Model encoder weights. Following standard practise, each block has batch normalization and dropout, with ReLU activation for intermediate layers and softmax activation that outputs the probability distribution over target classes at the last layer. We used gradual unfreezing to fine-tune the model. Instead of fine-tuning all the layer at once, we unfreeze the model starting from the last layer. We first unfreeze the last layer and fine-tune all unfrozen layers for one epoch. We then unfreeze the

next lower frozen layer and repeat, until we finetune all layers until convergence at the last iteration.

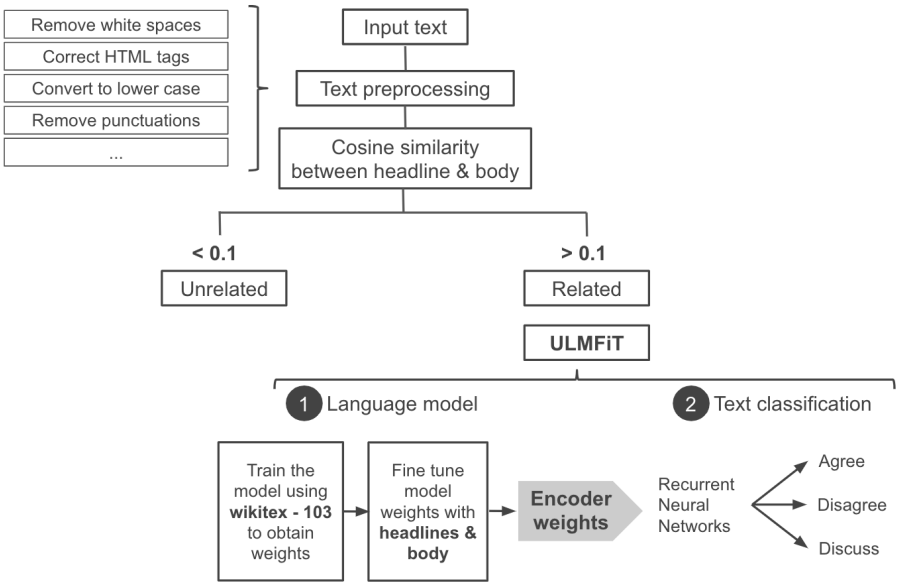


Figure 2: A visualization of our methods for developing this model.

5 Results

As we mentioned in Section 3, the dataset has already be splitted into training set and test set by organizers of the competition with similar ratio of entries in each class.

We use a Relative Weighted Accuracy score as our metric to evaluate the model performance.

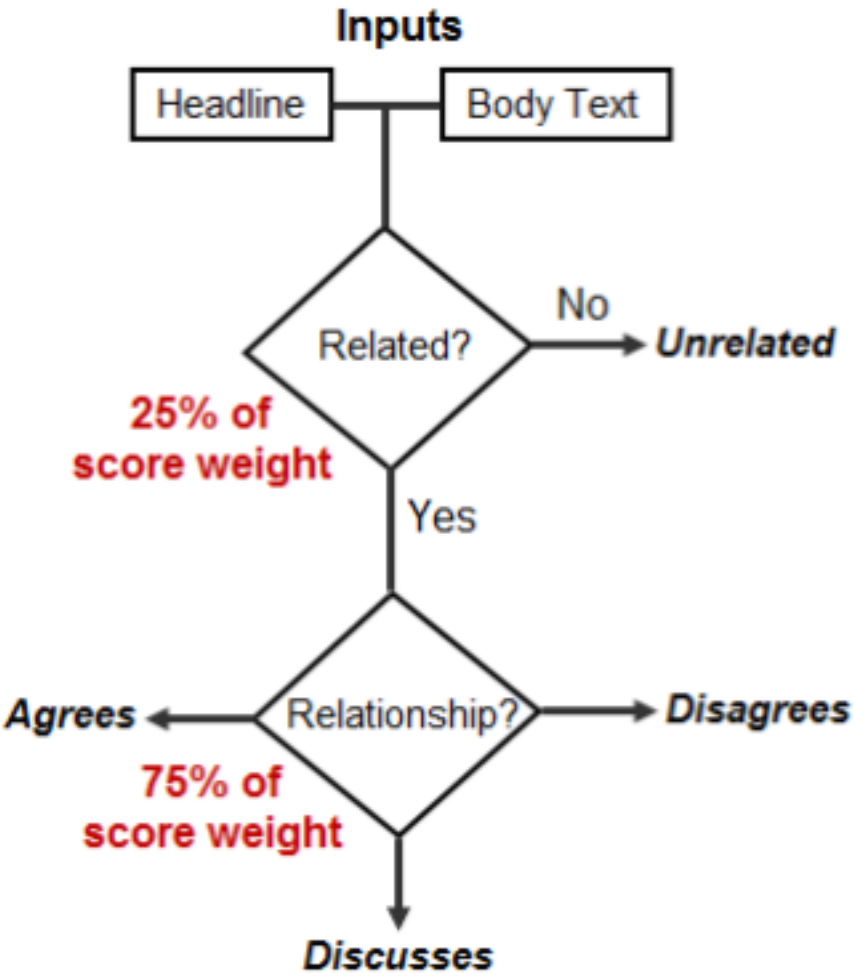


Figure 3: Visualization of Scoring Metrics.

if a [HEADLINE, BODY TEXT] pair in the test set has the target label unrelated, a team's evaluation score will be incremented by 0.25 if it labels the

Pred/True	Agree	Disagree	Discuss	Unrelated
Agree	1045	288	735	322
Disagree	23	19	7	56
Discuss	622	222	3083	356
Unrelated	398	38	298	17615

pair as unrelated.

If the [HEADLINE, BODY TEXT] test pair is related, a team's score will be incremented by 0.25 if it labels the pair as any of the three classes: agrees, disagrees, or discusses.

The team's evaluation score will so be incremented by an additional 0.75 for each related pair if gets the relationship right by labeling the pair with the single correct class: agrees, disagrees, or discusses.

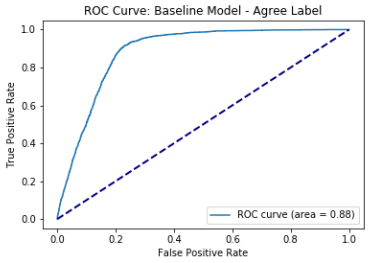
Relative Weighted Accuracy = Sum of weighted score of the predicted values/ Sum of weighted score of the entire set

5.1 Confusion Matrix of ULM Model vs Baseline Model

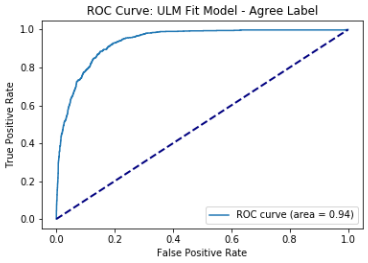
TO DOOO

5.2 ROC Curves for ULM Model vs Baseline Model

As we discussed, there could be four possible classes that a news can be categorized by our model: Agree, Discuss, Disagree, Unrelated. Figure 3 shows the ROC curves of baseline model (left) and our ULM model (right) for each of the class:

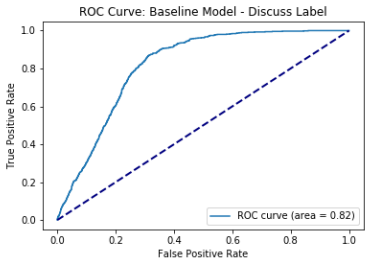


(a) ROC Curve for Baseline Model for Agree

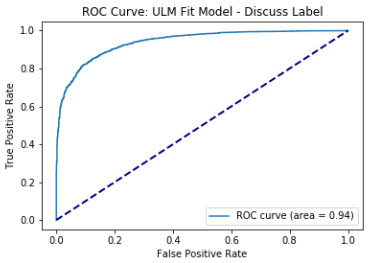


(b) ROC Curve for ULM Model for Agree

Figure 4: Agree Label ROC Curve

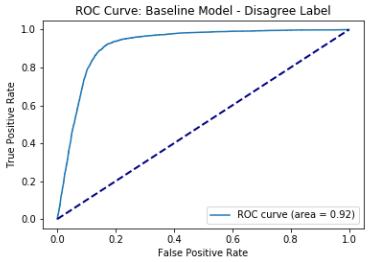


(a) ROC Curve for Baseline Model for Discuss

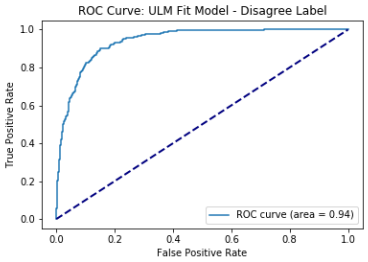


(b) ROC Curve for ULM Model for Discuss

Figure 5: Discuss Label ROC Curve

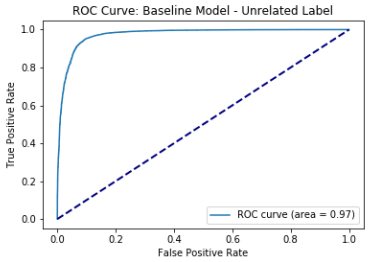


(a) ROC Curve for Baseline Model for Disagree

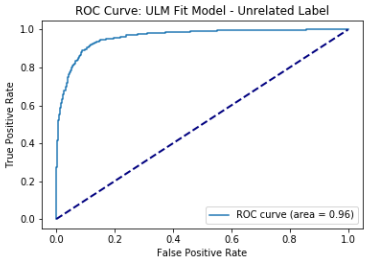


(b) ROC Curve for ULM Model for Disagree

Figure 6: Disagree Label ROC Curve



(a) ROC Curve for Baseline Model for Unrelated



(b) ROC Curve for ULM Model for Unrelated

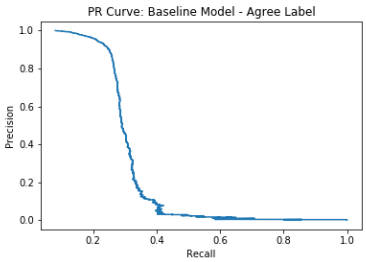
Figure 7: Unrelated Label ROC Curve

As can be seen, our ULM model outperforms the baseline model on those

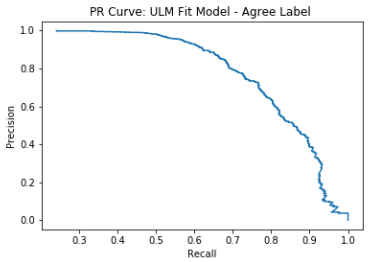
minority classes ('Agree', 'Discuss' and 'Disagree'; see Figures 4 6 5). The performance is close to the baseline model on the majority class ('Unrelated'; see Figure 6). We would argue that our ULM model's is able to distinguish the semantic information in the text since in all three minority classes, since each pair of the headline and body are related. Although the baseline model is able to detect the 'unrelated' pairs better than ULM model, it fails to capture the detailed attitude in those text.

5.3 PR Curves for ULM Model vs Baseline Model

PR curve is a crucial metric for evaluating classifier performance. This is especially true for unbalanced dataset, because a classifier could perform very well on the majority class while having a poor performance on the minority class. This could not be distinguished by only inspecting the ROC curve. Figure 4 shows the PR curves of baseline model (left) and our ULM model (right) for each of the class:

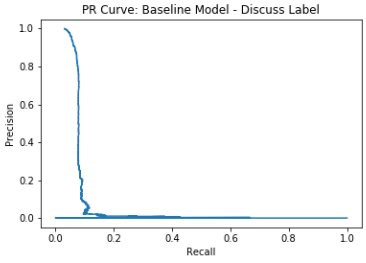


(a) PR Curve for Baseline Model for Agree

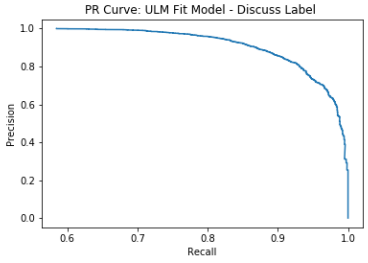


(b) PR Curve for ULM Model for Agree

Figure 8: Agree Label PR Curve

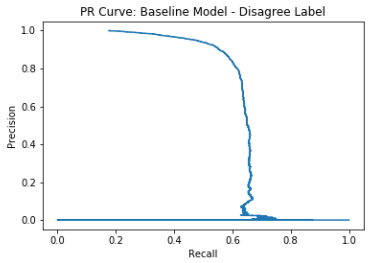


(a) PR Curve for Baseline Model for Discuss

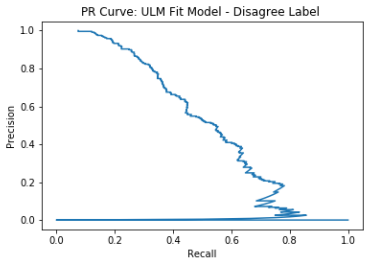


(b) PR Curve for ULM Model for Discuss

Figure 9: Discuss Label PR Curve

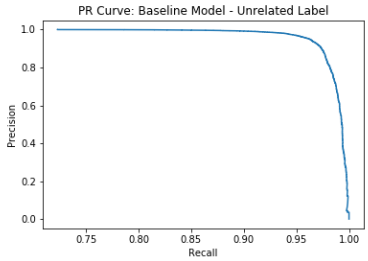


(a) PR Curve for Baseline Model for Disagree

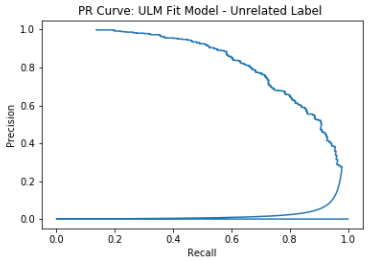


(b) PR Curve for ULM Model for Disagree

Figure 10: Disagree Label PR Curve



(a) PR Curve for Baseline Model for Unrelated



(b) PR Curve for ULM Model for Unrelated

Figure 11: Unrelated Label PR Curve

As we can see on the graphs above, our model outperforms (Figures 8 10 9) classifying Agree, Disagree, and Discuss cases. However, the performance for the Unrelated cases is not optimal (Figure 11).

6 Conclusion

In this paper, we implemented a ULMFiT-based model to tackle with stance detection task, the first step to combat fake news. Although this methodology is well-known for shared tasks, there is yet no analysis or reproduction study of the fake news problem we tackle. We took advantage of the cutting-edge NLP model in the hope that it could perform better than previous attempts. In the end, our model outperforms the winner's model score of (Talos Intelligence with relative score of 82.02) in FNC-1 by receiving a relative score of 83.6. Our work provides a new way of performing stance detection, and further confirm how powerful ULMFiT is in dealing with text classification problems. For future work, more sophisticated machine learning techniques can be incorporated to

model the semantic understanding, and to determine the stance on the basis of propositional content instead of relying on lexical features.