MSCI 598 Report

Nicholas Calen, Darion Stewart

05/10/2022

1 Abstract

In the internet age, we have been given the luxury of endless streams of quickly accessible information. Although this data rate has allowed us to advance at an unprecedented rate the burden of sorting through what is true and what is not has grown for the consumer of information. This false information or "Fake News" has become increasingly prevalent. The task of sorting through this information is of a monumental scale, which is why many government institutions and organizations have applied AI to do such sorting in an effort to relieve some of the consumer burden.

The organizers of the Fake News Challenge, an AI competition created to tackle this sorting problem, believe a helpful first step towards identifying fake news is to understand what other news organizations are saying about a topic. This process, called Stance Detection, could serve as a useful building block in an AI-assisted fact-checking pipeline. They provide a dataset with the goal of labeling whether a particular headline and article body agree, disagree, discuss each other, or are unrelated. A machine learning model should be able to differentiate between the labels.

In this project we use the latest state of the art natural language algorithms to tackle this challenge. By incorporating these models as well as applying some clever thinking, we were able to both beat the baseline and winning scores of the competition. This paper describes the process of arriving to such a methodology. We discuss our research, data exploration, and experimentation that led us to the final methodology.

2 Background

2.1 Literature Review

The FNC dataset has been available for some time, therefore there was a significant amount of previous work to explore and build off of for this project. Since the release of the FNC data in 2017 there has also been significant improvements in language model architecture for text classification tasks such as in this project. We used knowledge from this previous work to both experiment and refine our methodology and additionally incorporated some modern techniques and models into our work.

We began by exploring the methodologies of those who had achieved the best competition scores and reading their respective papers that desdcribe their solutions. Papers with code is an excellent resource to explore the top papers and corresponding code for competition datasets [1]. The site provided us with the state of the art (SOTA) results for the FNC dataset. In Figure 1 we see that the best weighted accuracy result on this data was ~90%, with the rest of the pack achieving weighted accuracy results in the low 80's.

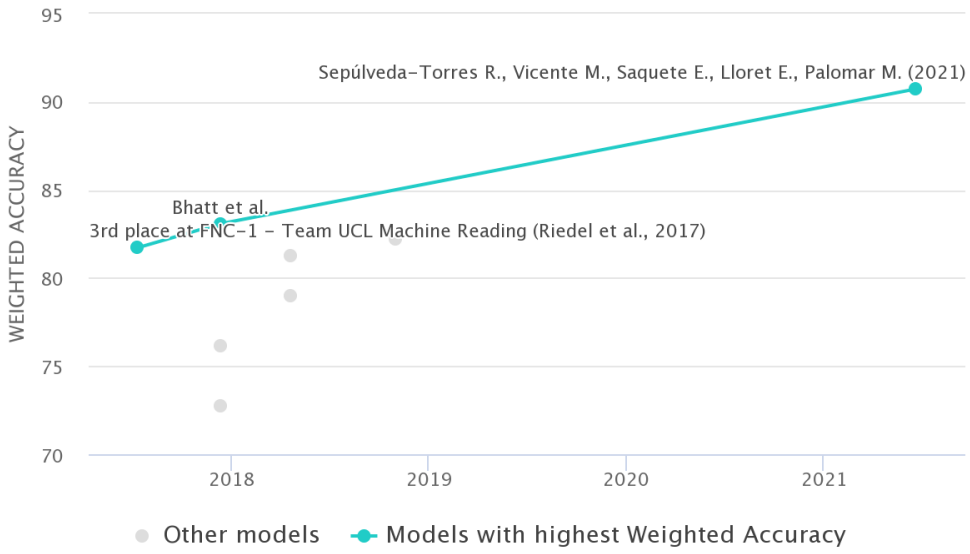

Figure 1: FNC All Time Best Results - Weighted Accuracy [1]

So what techniques did the competitors utilize to get them to the top of the leaderboard? Looking at the best result through the paper "Exploring Summarization to Enhance Headline Stance Detection" by Sepúlveda-Torres et. al. (2021), the authors use neural networks to first summarize the article bodies and used those along with the Headline to train a model to determine stance. It appeared that summarizing the articles reduced the text to key points, essentially removing the useless information that would confuse a later trained ML model [2]. The authors use neural networks throughout the pipeline for their models and take a two stage approach to generating predictions. The first stage used a model to determine if a headline is related or unrelated to a corresponding article body. The related posts then go to a second stage model to be further classified into agree, disagree, discuss. In the experiments section of this report, we'll describe our experience with this technique and whether or not it improved our results as well. Additionally we read the paper, "A simple but tough-to-beat baseline for the Fake News Challenge stance detection task" by Riedel et al. (2017), who happened to be the 3rd place team in the original FNC competition. Instead of typical tree-based methods that incorporate vectorizers and TFIDF to form the feature space of the examples, this paper utilizes word embeddings and neural networks to build out the training examples [3]. The relationships between words are richer with respect to the information they provide. A consequence of the weights being determined through previous training on very large corpuses of text data.

In further exploring the literature we observed that the best approach with respect to choosing a model is to go with a modern neural network such as a transformer (BERT and it's related models). We'll experiment with summarization of the posts as well and observe any improvements to our modeling results.

2.2 Exploratory Data Analysis

To get a better sense of the data we were working with, we started this project with an exploratory data analysis of our text examples and their stance. We wanted to look for relationships and patterns that we could then spin into strategies for producing a model to predict stance.

Given that this is a multi-class classification problem, we first observe the distribution of the unrelated, agree, disagree, discuss labels. In Figure 2, on the x-axis we have the specific class and on the y-axis we have the count of each class represented by the length of each bar. Above each bar is the percentage of the examples having the corresponding class label in the training dataset.

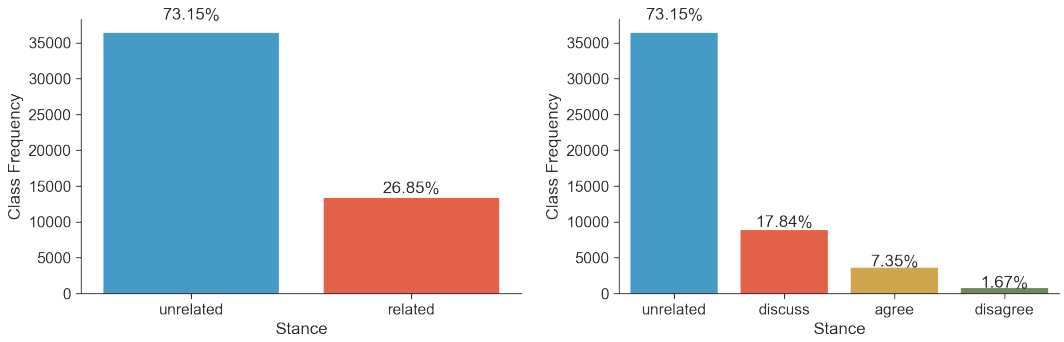

Figure 2: FNC Training Data Class Label Distribution

On the left bar diagram we see the breakdown between the related and unrelated examples. We observe that the training dataset is imbalanced in favor of the unrelated class. On the right we see the further breakdown of the related examples into agree, disagree, and discuss classes. Again the diagram on the right proves that the imbalanced classification problem persists when we break things down further. Given the distribution of the labels it appears that the best strategy would to break the stance classification problem into two stages to help with the imbalance. A first stage model will classify examples into unrelated and related buckets, and another model will classify the related posts into agree, disagree, discuss buckets. Here on, we'll refer to the two parts of the classification methodology as stage 1 and stage 2 respectively.

First focusing on stage 1, we wanted to determine what aspects of language relate a headline to an article body. What makes them related? What makes them unrelated? We studied this and came to an interesting conclusion. Using the two examples in Figure 3 we can explain our discovery.

Body ID	Headline	articleBody	target_stage1
2453	Michelle Obama's face blurred by Saudi state television	A short video clip has been circulating online Tuesday that purports to show Saudi Arabia's state television station blurring out the image of First Lady Michelle Obama as she and President Barack Obama met the new Saudi King during a visit to that country this week. But according to reporters who were on the ground as well as the information director at the Saudi Embassy in Washington, that was not the case.\n\n"Saudi TV has been showing the total arrival ceremony at the airport and at the Palace and nowhere is anything blurred," Nail al-Jubeir told Bloomberg View's Josh Rogin in an emailed statement.\n\nAdditionally, CNN's Hala Gorani said her colleague Nic Robertson saw the first lady's non-blurred image on Saudi TV himself:\n\nOur Nic Robertson in Riyadh telling me footage he saw on Saudi TV did not show a blurred Michelle Obama. On that note, Good night Twitter! — Hala Gorani (@HalaGorani) January 27, 2015\n\nThe Wall Street Journal's Ahmed Al Omran confirmed the same:\n\nMichelle Obama's blurry video is not real. The arrival was broadcast in full on state television without any blurring. @Max_Fisher — Ahmed Al Omran (@ahmed) January 27, 2015\n\nAs you can see from the un-blurred photo below, Michelle Obama declined to wear the traditional Muslim head scarf that is mandatory for all Saudi women but not required for visiting foreigners. However, she was not permitted to shake the new King Salman's hand:\n\nMichelle Obama forgoes a headscarf and sparks a backlash in Saudi Arabia http://t.co/uycgVygf7V pic.twitter.com/6ICNigFaPT — Washington Post (@washingtonpost) January 27, 2015\n\nWatch video below, via YouTube:\n\n[Photo via screengrab]\n\n– –\n\n>> Follow Matt Wilstein (@TheMattWilstein) on Twitter	related
1506	Boko Haram denies cease-fire, leader says he married off kidnapped girls	BAGHDAD, Iraq – The Islamic State (ISIS) incinerated the corpses of five militants who were suspected of contracting Ebola, an Iraqi health official indicated.\n\nFaisal Ghazi, member of the Health Committee in Iraq's council of ministries, said the five were incinerated in Mosul, the ISIS stronghold in Iraq.\n\n"The Islamic State organization incinerated five militants infected with Ebola to prevent further spread of the disease in Mosul," he said.\n\n"ISIS had proof that these militants were infected with Ebola," he added, without giving details of whether the fighters died of the disease or were killed and incinerated by the group.\n\nThe UN's World Health Organization (WHO) had been investigating cases of Ebola in Mosul, following reports that some militants with Ebola symptoms had reported to a hospital in the city.\n\nISIS volunteers have poured into Iraq and Syria from around the world, including countries in Africa.	unrelated

Figure 3: Unrelated - Related Examples

The first row is a related example. So what is the link between the two? Interestingly, what links the two is the nouns that are common between the headline and the article body! The nouns Michelle Obama, Saudi, link these two pieces of text. Likewise, the second row contains an unrelated example. There are no common nouns between the headline and the body, one is discussing a Boko Haram event, the other an Iraq event. Obviously the two are unrelated, but the nouns mentioned are the key words determining this link! Given the success of this first analysis we moved on to Stage 2 with the goal of determining what makes a headline and article body agree, discuss, or disagree with each other. We visualize some examples in Figure 4.

Body ID	Headline	articleBody	Stance
2345	CNN plays chilling audio recording allegedly from Michael Brown shooting: “At least 11 shots”	Video messaging firm confirmed today the tape where shots appear to ring out in the background was filmed at 12:02:14 PM (CDT) on August 9\n\nThe individual who made the recording has remained anonymous but handed tape over to the FBI\n\nMultiple shots can be heard - in two separate volleys of gunfire with a pause in the middle\n\nA video texting service confirmed today that an audio recording - which appears to contain the sound of shots fired by Officer Darren Wilson when he killed unarmed Michael Brown - is authentic.\n\nChaim Haas, head of communications at Glide, verified the tape recorded by an unnamed individual in Ferguson, Missouri on August 9.\n\nAn estimated 11 shots are heard to ring out in the background of the recording with a brief but significant pause between the first seven shots and the last volley of four.\n\nGlide released a lengthy statement on their website today which said the user had been live video-messaging with a friend when the gunshots that killed Michael Brown were caught in the background audio.\n\nScroll down for video\n\nGlide, an app which provides a real-time video texting service, said on Thursday that the audiotape was genuine and that it was now in the hands of the FBI\n\nThe statement continued: 'Because Glide is the only messaging application using streaming video technology, each message is simultaneously recorded and transmitted, so the exact time can be verified to the second. In this case, the video in question was created at 12:02:14 PM (CDT) on Saturday, August 9th.'\n\nOfficer Darren Wilson shot dead Michael Brown on August 9. On a new audio at least ten shots can be heard\n\nGlide confirmed to MailOnline today that the FBI had been in touch with them regarding the audio recording.\n\nThe start-up business also said in the statement that it was proud of their app user for turning over the video message to the FBI investigation.\n\nExperts who have listened to the audio have said it could prove damning for Officer ...	discuss
1412	Nun Complains Of Stomach Pains, Later Gives Birth To Baby Boy	A cloistered nun has stunned her mother superior and sisters after giving birth to a baby boy after complaining of severe stomach pains.\n\nThe sister, who belonged to an order in Macerata, in the eastern Italian region of Le Marche, claimed to have no idea she was pregnant when she was rushed to hospital in agony, after which she gave birth.\n\nThe South American nun, who arrived at the convent in June, when it is supposed she was already pregnant, was taken to the emergency department of 'Bartolomeo Eustachio' di San Severino Marche by her fellow sisters.\n\nThe childbearing nun, originally from South America, claimed to have 'no idea' she was pregnant\n\nDoctors quickly unravelled the cause of the mysterious ailment, Il Corriere Adriatico reported.\n\nThe baby was born healthy but remains in hospital to undergo more checks, while the nun's convent has expressed an interest in taking care of him, according to L'Unione Sarda.\n\nThe case bears a striking similarity to that of a 33-year-old Salvadorean nun in Italy, who gave birth to a baby boy last year, whom she named after Pope Francis.\n\nShe told her social worker she did not feel guilty and would raise the child saying: 'I am so happy. 'He is a gift from God. I feel more of a mother than a nun.'\n\nIn 2011 a Congolese nun in an Italian order gave birth to a baby girl after being raped by a priest.\n\nShe gave up the baby for adoption but after being refused re-entry to the convent changed her mind and recovered the child.	agree
1841	Christian Bale to play Steve Jobs in upcoming biopic	After early discussions to play the Apple founder, Christian Bale has decided to part ways with the Steve Jobs biopic at Sony.\n\nWhile negotiations were never fully under way, sources tell Variety that Bale had talks with director Danny Boyle about taking on the role, but a deal never came to fruition. The news comes almost two weeks after the biopic's screenwriter, Aaron Sorkin, told Bloomberg that Bale was a lock for the part of Jobs. ““We needed the best actor on the board in a certain age range and that's Chris Bale,” he said. “It's an extremely difficult part and he's gonna crush it.”\n\nSony will now look to find its replacement after Bale and Leonardo DiCaprio both passed on the part. The studio is in discussions with Seth Rogen to play Apple co-founder Steve Wozniak.\n\nScott Rudin, Mark Gordon and Guymon Casady are producing.\n\nBale is in negotiations to play Travis McGee in Fox's “The Deep Blue Goodbye” and can be seen next in Fox's “Exodus: Gods and Kings.” He is repped by WME.	disagree

Figure 4: Agree, Disagree, Discuss Examples

Unlike with the first stage targets the pattern is not clear. There is likely a more complicated underlying function that determines the classes in this case. We'll likely have to use a more advanced model for this stage.

3 Methodology

3.1 General Approach

As previously stated we took a two stage approach to this classification problem. We treat the first stage as a binary classification problem, classifying article and headline pairs as either related or unrelated. The second stage is treated as a multi-class classification problem taking as input the article/body pairs that were labelled related in stage 1 and then classifying them as either in agreement, disagreement or discussed. Figure 5 illustrates this flow along with the intermediate steps.

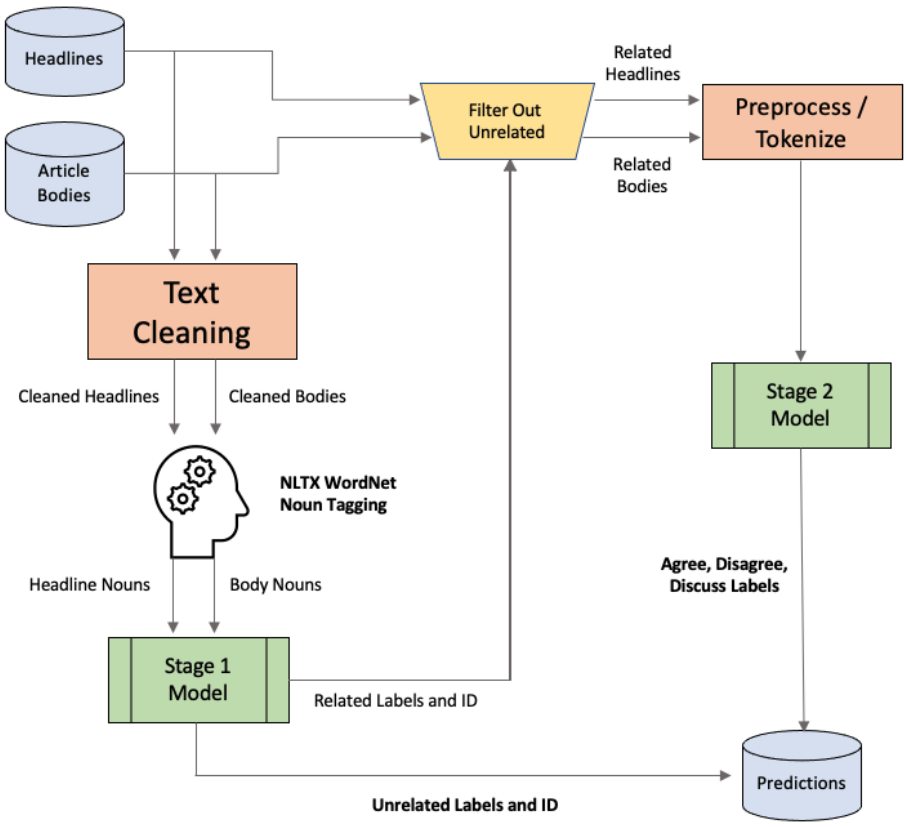

Figure 5: Methodology Flow

3.2 Stage 1

In stage 1 we opted for a less than traditional approach to solve this binary classification problem. Typically we would throw a transformer model at the problem and fine-tune it on our dataset and get relatively good results without much effort (when we tried this we achieved 92% accuracy out of the gate without much work). Using a "model-free" approach (methodology) we can actually do a lot better and become competitive with the SOTA models from the literature review. We decided that this approach would be more interesting/enjoyable to try, something to differentiate our group from the current methodologies. As shown in section 5 of this report, we were quite succesful using it.

In our EDA we learnt that nouns (ex. hat, boy, mother), proper nouns (ex. Karen, Canada), and sometimes numbers give a strong signal on whether or not a headline/body pair are related. In our approach we only keep these nouns and throw away everything else. We use NLTK's WordMet and entity recognition to do this tagging, removal of non nouns, stemming and lemmatization. We noticed that special characters ($,#,%) can get picked up in the tagging process so we remove them in a text cleaning step before preforming the tagging. Figures 6 and 7 show examples of the end of this process. We can observe that nouns and numbers are what remains in the variables headline_ents and body_ents.

```
Headline         Soldier shot, Parliament locked down after gunfire erupts at war memorial
headline_ents                                  [soldier, parliament, gunfir, war, memori]
Name: 0, dtype: object
```

Figure 6: Noun Tagging - Headline

```
articleBody    Afghanistan veteran Sam Arnold uploaded this spine-chilling video of a US Marine getting a direct head
shot from a Taliban sniper—only to be saved by his kevlar helmet. It's incredible to watch, especially the face of re
lief and disbelief of the impact victim. That was a really close call.\n\nAccording to Arnold, "the Marines were cond
ucting a joint helicopter raid in the Now Zad district, Helmand Province in 2013. The shot occurs right at the :45 ma
rk in the video."
body_ents
[afghanistan, veteran, sam, arnold, spine, video, marin, headshot, taliban, sniper, kevlar, helmet, face, relief, dis
belief, impact, victim, arnold, marin, helicopt, raid, zad, district, helmand, provinc, 2013, shot, 45, mark, video]
Name: 115, dtype: object
```

Figure 7: Noun Tagging - Body

Now that we have retrieved the nouns we need some way to compare the two sets of nouns for similarity. We had a few options to compute the scores, however, the simplest option the Jaccard Similarity Index [3] proved to yield the strongest signal regarding the relation between a headline and body. The Jaccard index, also known as the Jaccard similarity coefficient, is a statistic used for gauging the similarity and diversity of sample sets, yielding a value between 0 (no common set elements) and 1 (the sets are exact copies). For our task we are essentially counting the number of shared nouns normalized by the size of the total set of nouns. The Jaccard index between a headline (A) and body (B) is calculated using the following equation:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

When first experimenting with this methodology we were hoping to observe two distinct groups of Jaccard scores. A group of higher Jaccard scores corresponding to the examples labeled related and another group of lower Jaccard scores corresponding to the unrelated examples. Observing Figure 8 we got exactly what we were looking for! We observe two distinct groups with little overlap.

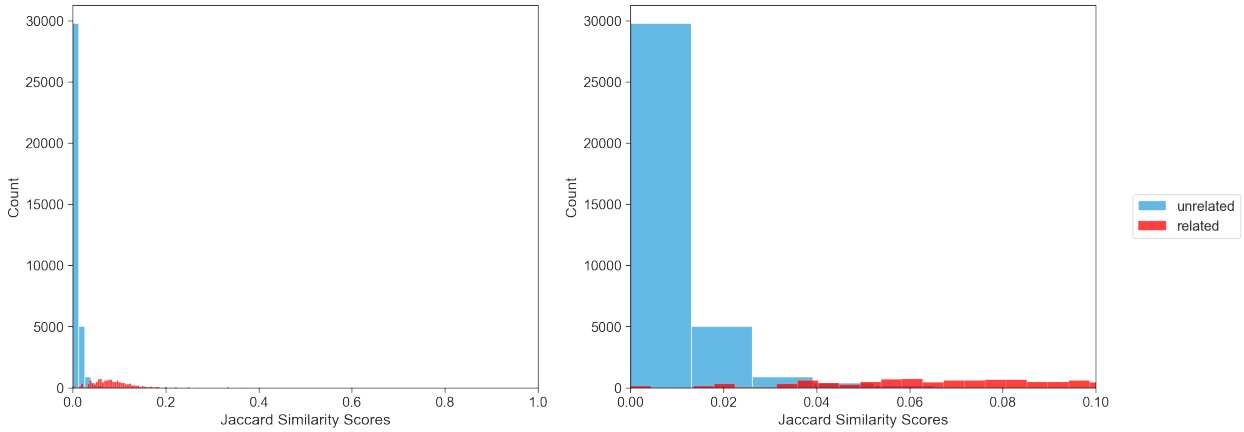

Figure 8: Distribution of Jaccard Scores Training Set

This provides evidence that we can perform the labeling task in stage 1 using a model free threshold line. All we have to do is optimize the location of this line by maximizing accuracy on the training data! Observing the right subplot of Figure 8, we might want to place this line around 0.04 for example. Anything greater than the threshold line getting the label related, else unrelated. We can further optimize the Jaccard scoring by optimizing the number of nouns that we include in the scoring. We place this cap on the article body because typically we find more nouns. We found that optimizing this number improved predictive accuracy of the model free approach and therefore include it as a parameter in the flow diagram illustrated in Figure 9. Details on accuracy of this stage 1 approach are included in section 5 of this report.

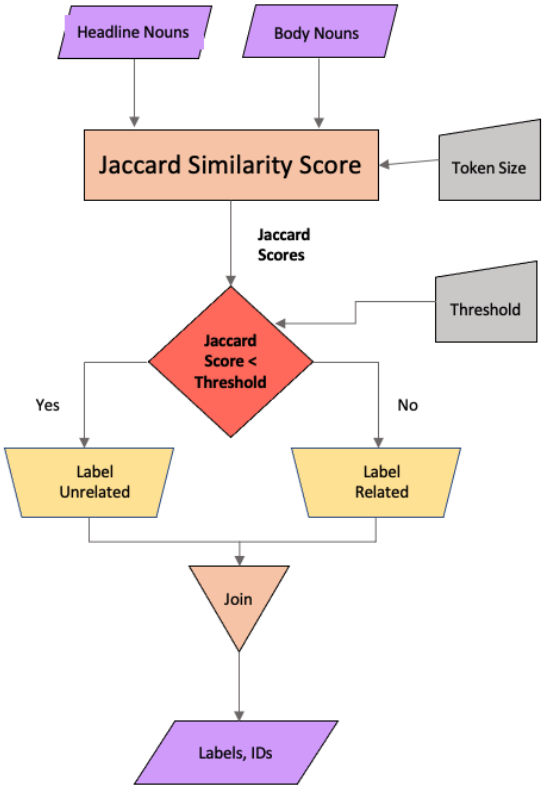

Figure 9: Stage 1 Flow

3.3 Stage 2

As stated in the EDA section of this report, the complexity of the multi-class stage 2 labeling problem had us move away from the interesting model free approach of stage 1 to a model based approach in stage 2. To harness the power of transfer learning and enable quick experimentation our group utilized the HuggingFace Transformers library to develop our model. Through experimentation we found the RoBERTa model to provide the best results on the given test set. RoBERTa [4] is generally the same as the BERT discussed in class with better pretraining tricks [5]:

- dynamic masking: tokens are masked differently at each epoch, whereas BERT does it once and for all
- no NSP (next sentence prediction) loss and instead of putting just two sentences together, put a chunk of contiguous texts together to reach 512 tokens (so the sentences are in an order than may span several documents)
- train with larger batches
- use BPE with bytes as a subunit and not characters (because of unicode characters)

RoBERTa is quite large so we won't display it's architecture here. Details can be found on the HuggingFace model repository. We fine-tuned a pretrained RoBERTa model for sequence classification on our training dataset with the following parameters:

```
from transformers import TrainingArguments

training_args = TrainingArguments("test-trainer",
                                  evaluation_strategy="epoch",
                                  per_device_train_batch_size=16,
                                  num_train_epochs=3,              # total number of training epochs
                                  weight_decay=0.1,                # strength of weight decay
                                  per_device_eval_batch_size=16)
```

Figure 10: RoBERTa Training Arguements

The HuggingFace library provides a useful API to tokenize multi-text inputs! Pre-tokenization we cleanup the special characters in the text examples. The tokenization strategy given that we have two text inputs produces the output show in Figure 11. Sentence 1 representing the headline, sentence 2 the article body.

```
inputs = tokenizer("This is the first sentence.", "This is the second one.")
inputs
```

```
{
  'input_ids': [101, 2023, 2003, 1996, 2034, 6251, 1012, 102, 2023, 2003, 1996, 2117, 2028,
  'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],
  'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
}
```

Figure 11: Tokenization [6]

We see how the token_type_ids indicate which piece of text the tokens belong to. The flowchart in Figure 12 summarizes the stage 2 process.

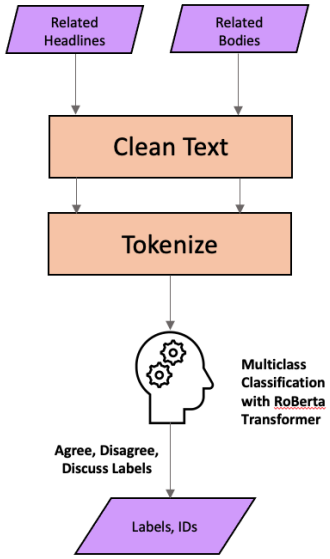

Figure 12: Stage 2 Flow

4 Experiments

4.1 Stage 1 Optimizations

Noun tagging proved to be expensive computationally because of the size of the FNC dataset. Optimization were performed on a sample of 1000 from the training dataset. Results were computed on the provided training set and decisions were made with respect of the methodologies performance on it. Due to this methodology being model free, statistically there's no need to look at the performance on a separate test set. Train is essentially test in this case because there is no fitting process. NLTK's entity recognition feature allowed us to specifically pick up different noun categories [7]. The list included:

'PERSON','NORP','FAC','ORG','GPE','LOC','PRODUCT','EVENT','WORK_OF_ART', 'LAW','LANGUAGE','DATE','MONEY', 'NOUN','PROPN','NUM'

Details on what each specific noun tag refers to can be found in the data dictionary at Reference 7 in this report. We first optimized which set of categories chosen resulted in the best test set performance. The search space would be too large to try all possible combinations so we picked six combinations that we thought would be comprehensive. The results are presented in Table 1.

Table 1: Stage 1 Tag Optimization

Trial #	Noun Types Included	Recall-Related (%)	Recall-Unrelated (%)	Accuracy (%)
1	NOUN	76.8	90.1	86.5
2	NOUN, PROPN	94.2	95.6	95.2
3	NOUN, PROPN, NUM	91.7	95.9	94.7
4	NOUN, PROPN, NUM, ORG, DATE	93.8	95.9	95.3
5	PERSON, NORP, FAC, ORG, GPE, LOC, PRODUCT, EVENT, WORK_OF_ART, LAW, LANGUAGE, DATE, MONEY	72.1	95.1	86.8
6	PERSON, NORP, FAC, ORG, GPE, LOC, PRODUCT, EVENT, WORK_OF_ART, LAW, LANGUAGE, DATE, MONEY, NOUN, PROPN, NUM	95.2	94.9	95.0

It's clear by Table 1 that having nouns and proper nouns was essential. Including dates, organizations and numbers boosted the accuracy slightly. Lemmatization and stemming were used on the nouns in the first case. We wanted to determine if performance changed without incorporating these techniques (lem and stem). The results of these next trials are shown in Table 2.

Table 2: Stage 1 Word Augmentation Optimization

Trial #	Augmentation Technique (%)	Recall-Related (%)	Recall-Unrelated (%)	Accuracy (%)
1	None	90.2	93.8	92.8
2	Stemming	93.8	96.0	95.4
3	Lemming	92.7	96.1	95.2
4	Lem and Stem	93.8	95.9	95.3

We can see that only using stemming leads to the optimal result. Additionally, we wanted optimize the placement of the threshold line and not just "eyeball it" to maximize classification accuracy in this stage. We use the SciPy library's optimization feature to do so and call this threshold placement objective_a. Additionally, we want to maximize accuracy according to the number of nouns we allow into the Jaccard score from the article bodies. We call this objective_b. Figure 13 displays these functions as implemented in python code.

```
def objective_a(threshold):
    df['pred'] ='unrelated'
    df.loc[df.jaccard_similarity_nouns > threshold, 'pred'] = 'related'
    return -accuracy_score(df.target_stage1, df.pred)

def objective_b(token):
    df['pred'] ='unrelated'
    df['jaccard_similarity_nouns'] = df.apply(lambda x: jaccard_similarity_token_opt(x.headline_ents,
                                                                                     x.body_ents,
                                                                                     token), axis=1)
    df.loc[df.jaccard_similarity_nouns > threshold, 'pred'] = 'related'
    return -accuracy_score(df.target_stage1, df.pred)
```

Figure 13: Stage 1 Objective Functions

In our experiments, we saw on average a 1% bump in accuracy on the training set attributed to each optimization (~2% gain total).

4.2 Stage 2 Optimizations

The results in this section were calculated on the test set provided by the FNC competition, NOT on a sample of the training set as in section 4.1. Over the course of these experiments we keep the training parameters consistent with those shown in Figure 10. We note that keeping weight decay higher leads to increased accuracy but we did not fully investigate this parameter in this project.

In an effort to match the performance by Sepúlveda-Torres et. al. (2021) we used the RoBERTa transformer model to first summarize the article bodies before training on them then predicting on the test set provided by the competition. We did not see improved performance, rather we saw a 4% drop in accuracy from our groups stage 2 baseline. Given the drop in performance we will not be providing those results in this section.

Given the label distribution shown in Figure 2, we knew immediately that class imbalance would prove to be an issue. To overcome this typical classification problem we applied oversampling and undersampling techniques to the training data. We first trained a baseline model using BERT on the full training dataset. Subsequently, we tried undersampling by reducing the discuss examples to 4000 so that the class distribution would be more even but still resemble the training data slightly. Following this we attempted to oversample the minority class (disagree label) by doubling the number of training examples corresponding to it (increasing from ~1000 to ~2000). The oversampling strategy we used involved using the NLPAug python library to

change words at random within an example, replacing the chosen word(s) with it's synonym. We used a Bernoulli trial to determine whether the headline should be augmented or the article body. Finally we tried a combination of oversampling the minority class (disagree, doubling) and undersampling the majority class (discuss, reducing to 4000 rows, synonym augmentation). Table 4 presents the results of these trials.

Table 3: Stage 2 Data Optimization

Trial #	Sampling Strategy	Recall-Agree (%)	Recall-Disagree (%)	Recall-Discuss (%)	Accuracy
1	None	76	14	88	78
2	Undersampling	76	45	82	77
3	Oversampling	64	43	88	77
4	Oversampling and Undersampling	76	30	84	76

From this first experiment we see at least a 1% drop in accuracy when applying any of the sampling strategies outlined. Although there is a small drop in accuracy, oversampling seems to balance the recall a bit better across all classes. We suggest then using the oversampling strategy in favor of getting more disagree values correct at the small expense of the other classes.

Without oversampling, we experimented with several model types as many are available on the HuggingFace repository. Each has various strengths as described in the literature which is why we decided to try them out. We fine-tuned 3 pretrianed model architectures, specifically distil-bert, know for being a light weight BERT model thus making training quicker, BERT, the original transformer, and RoBERTa a popular extension of the original BERT model. Table 4 presents the results of these trials.

Table 4: Stage 2 Model Optimization

Trial #	Model Type	Recall-Agree (%)	Recall-Disagree (%)	Recall-Discuss (%)	Accuracy
1	distil-bert	68	7	85	72.2
2	BERT	76	14	88	78.0
3	RoBERTa	77	31	89	80.0

5 Conclusion

5.1 Results

5.1.1 Test Set Results

In this section we present the results of our methodology on the competition test set. Beginning with the first stage, in Table 5 we present the optimized stage 1 parameters for the threshold line (used to classify given Jaccard score input) and the max number of article nouns used (token size) in calculating the Jaccard score.

Table 5: Stage 1 Parameters

# Nouns (Token Size)	Jaccard Threshold Line
68	0.0238609

We can visualize the threshold line along with the Jaccard scores in Figure 14. We see the training distribution of Jaccard scores in Figure 8 closely resembles those of 14. Hence we have a methodology that works out of sample!

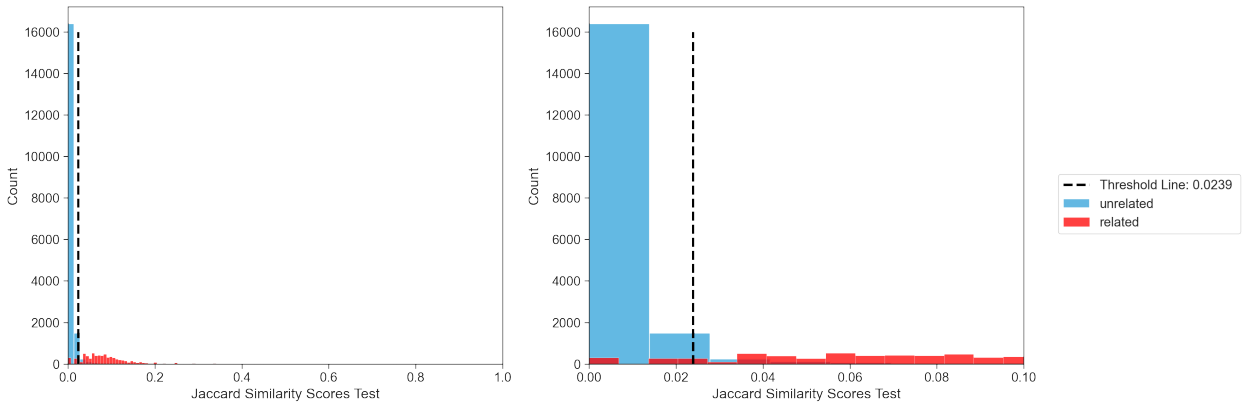

Figure 14: Jaccard Scores Test and Threshold Classification Line

The stage 2 model parameters were presented in section 3 and have not changed. In Figure 15 we visualize the confusion matrices of stage 1 on the left, followed by stage 2 on the right. We computed the stage 2 confusion matrix using only the posts that made it into stage 2 from stage 1 (those labeled related). There was some error in the first stage thus not all true related test examples made it through to stage 2. The confusion matrices displayed contain two metrics in each cell, the first number being the count of examples falling into the cell, the second number being a percent proportion of this count with respect to the dataset. If you were to add this second number along the main diagonal you would arrive at the total accuracy on the corresponding test set.

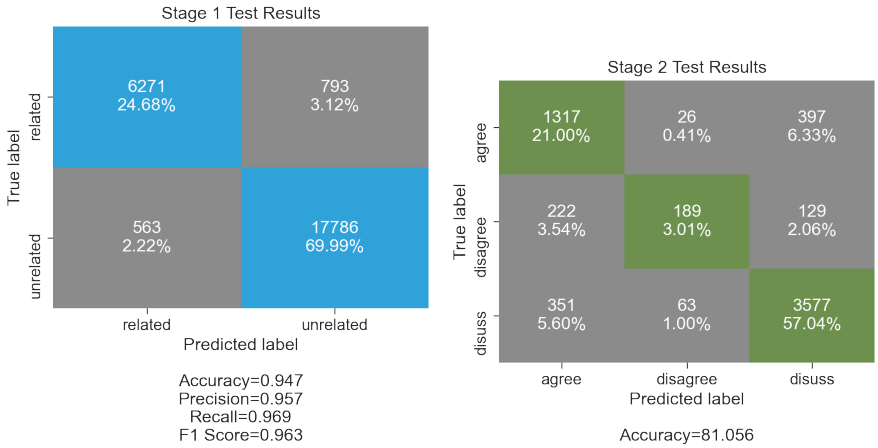

Figure 15: Confusion Matrices, Stages 1 and 2

We aggregate these results to form the overall confusion matrix shown in Figure 16 and tabulate the class by class accuracy in Table 6. We also use the FNC competition's scorer function to calculate our methodologies' weighted accuracy on the test set and add this to Table 6.

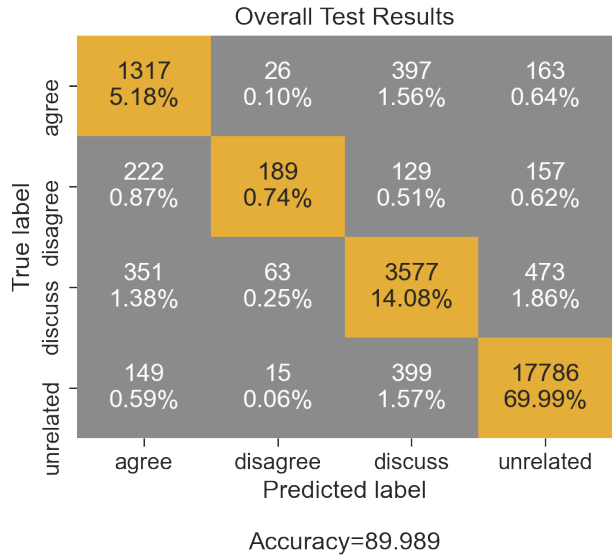

Figure 16: Confusion Matrix, Overall Results

Table 6: Overall Test Results, Metrics

Recall-Agree (%)	Recall-Disagree (%)	Recall-Discuss (%)	Recall-Unrelated (%)	Accuracy (%)	FNC Weighted Accuracy	FNC Score
69.2	27.1	80.1	96.9	89.9	84.34	9826.5

5.1.2 Comparison to SOTA

We can conclude that our overall approach is quite competitive within even the SOTA approaches. Tabulating the top 3 competitors from papers with code, along with our model shown in comparison produces Table 7. Our results are presented in bold along with the new ranking (by FNC weighted accuracy) with the addition of our approach [1].

Table 7: Overall Test Results, Metrics

Rank	Team/Model	Recall-Agree (%)	Recall-Disagree (%)	Recall-Discuss (%)	Recall-Unrelated (%)	FNC Weighted Accuracy
1	Sepúlveda-Torres R., Vicente M., Saquete E., Lloret E., Palomar M.	75.03	63.41	85.97	99.3600	90.73
2	**Our Approach**	**69.20**	**27.10**	**80.10**	**96.9317**	**84.34**
3	Bhatt et al.	43.82	6.31	85.68	98.0400	83.08
4	Bi-LSTM	51.54	10.33	81.52	96.7400	82.23

Our performance in the stage compared to those of the second and third place papers vaults our approach to second place in the rankings. We do quite well in the unrelated column as well, however this is scored less in the competition so increasing recall of the 'related' label in the first stage helped us in the second stage. More opportunity to get labels right here. We are quite proud of our approach, using no model for the first stage and limited time spent tuning RoBERTa in the second stage. We took advantage of statistics (oversampling) in the second stage only, and used our knowledge of the English language in the first stage to develop a simple methodology.

5.1.3 Error Analysis

So what kind of examples did our model miss? Why did we get things wrong? We noticed in the first stage that our method struggled with examples that had short headlines (< 4 word). There just wasn't enough nouns to calculate a proper Jaccard score. There were several of these with a 0 Jaccard score that should have been related but were labeled unrelated. That being said, we believe that it would be difficult for a human to distinguish between some of these as well without proper context such as an article link. Observing the confusion matrix of stage 2, agree and discuss examples are confused a lot. This might be do to similar language and semantics, the article or headline being vague. Disagree examples are most confused with agree labels which is interesting. We would think that disagree and agree examples should be both most confused with discuss examples but this is not the case. The model could be focused on the nouns mentioned instead of the context. There are ways to add context to transformer models, however this would require future work.

5.2 Recommendations for Future Work

To conclude this report, our group clearly achieved some excellent results given our hybrid modeling approach. We presented a novel method for the stage 1 portion of the classification not seen in the current literature for the FNC dataset. This stage was competitive with some of the top models consisting of advanced architectures. It was interesting to observe that simple calculations with the knowledge of a pattern could beat these other methods. In the future it would be interesting for someone to extend the Jaccard calculation into a weighted score. Weighting certain type of nouns more than others to compute the score. For example, if the same person let's say Michelle Obama was tagged in both the article body and headline, these two texts are most likely related. Adding some weight to names would be interesting to boost Jaccard in some instances and really separate those distributions more from Figures 8 and 14. Theoretically this would make the thresholding task easier. Additionally we could likely ensemble this stage with a transformer based model. Typically ensembeling increases classification accuracy thus this could be an interesting next step experiment.

With respect to stage 2, we did not spend much time optimizing the model hyperparameters nor the hyperparameters of the oversampling method in training. The others of the first ranked paper did optimize their models heavily which gave them quite a great score. If our group had more GPU resources at our disposal we could likely do the same and get closer to the top rank.

6 References

[1] https://paperswithcode.com/sota/fake-news-detection-on-fnc-1
[2] https://link.springer.com/chapter/10.1007/978-3-030-80599-9_22
[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html#:~:text=Jaccard%20similarity%20
[4] https://arxiv.org/abs/1907.11692
[5] https://huggingface.co/docs/transformers/model_summary
[6] https://huggingface.co/course/chapter2/4?fw=pt
[7] https://pahulpreet86.github.io/name-entity-recognition-pre-trained-models-review/