BaIT: Barometer for Information Trustworthiness

G36 (s2258771, s2169244, s2258979)

Abstract

This paper presents a new approach to the FNC-1 fake news classification task which involves employing pre-trained encoder models from similar NLP tasks, namely sentence similarity and natural language inference. Proposes two neural network architectures using this approach. Explores the use of data augmentation in tackling class imbalance in the dataset, using both common pre-existing methods and proposing a method for sample generation for the under-represented class using a novel sentence negation algorithm. Achieves comparable overall performance with existing baselines, while significantly increasing accuracy on an under-represented but nonetheless important class for FNC-1.

1. Introduction

In the modern age, where most people rely on online platforms for their information, the rapid spreading of fake news poses a significant problem (Vosoughi et al., 2018). Consequently, the challenge of automatic fake news detection has attracted the attention of machine learning and linguistics researchers, and several approaches have been proposed (see e.g. Busioc et al., 2020 for an overview of NLP-based approaches). In reality, however, the 'truth labelling' of claims is incredibly complex, even for humans, and such fully automated detection systems can thus hardly be implemented in practice. A more feasible alternative would be reliable AI systems assisting humans in recognizing harmful spreading of fake news. Specifically, a useful first step is to classify the relationship of the headline and body of the news article – a task known as stance detection. This would enable a human reader to quickly extract a number of articles that agree, discuss, or disagree with a certain text and use this to make their own judgement.

From this perspective, the Fake News Challenge (FNC-1) was organized in 2017 to encourage the development of artificial intelligence tools to combat fake news and assist human fact checkers[1]. To this end, they provide a data set containing headlines and bodies of news articles, each pair of which is labelled as *agree*, *disagree*, *discuss*, or *unrelated*. The aim is to classify the stance of a given headline w.r.t. a given body.

In this paper, we develop a novel approach to the FNC-1 stance classification task that utilizes pre-trained transformer models, which are extended with a custom attention layer and a neural classifier. Large transformer-based language models are a recent development in NLP which have proven to be massively successful in a wide variety of tasks (Devlin et al., 2018). In our approach, we combine transformer models that have been trained on two domains: Natural Language Inference (NLI) and Sentence Similarity (SIM). These models are applied to encode sentences into vector representations that contain information useful for solving the corresponding task. We perform the classification in a hierarchical manner, first predicting whether a head is *unrelated* to the body, and, if not, classifying its stance as one of the remaining three (related) categories: *agree*, *disagree*, or *discuss*. The second task is more challenging than the first, as it requires identifying logical relations between head-body pairs, which are often subtle in the text. Pre-trained models were chosen based on the apparent similarity in the nature of the pre-train objective and the tasks at hand: SIM embeddings are used to discriminate between related and unrelated pairs, and NLI embeddings to classify the logical relationship. In the most extensive model, an attention layer is applied that weighs NLI embeddings of the body sentences based on their similarity w.r.t. to head, computed from SIM embeddings. It is noteworthy that while the *disagree* class is particularly important in this task (it is the case in which the news is 'fake'), it is severely under-represented in the dataset, accounting for only 2% of the samples. As a result of this imbalance, many existing works have scored impressive overall performance on the task with very poor accuracy on the *disagree* class. In light of this observation, this work places particular emphasis on the *disagree* accuracy, and tackling imbalance in the dataset.

Finally, the main contributions of this paper can be summarized as follows:

(i) The proposal of a novel approach for stance detection in the news domain leveraging the power of large transformer models trained on NLI and SIM tasks.

(ii) The development of a neural-network architecture that intuitively combines these two representation types, using an attention layer, to effectively transfer knowledge from the NLI and SIM domains to the problem of stance detection.

(iii) An evaluation of methods in data augmentation as a means of resolving issues of class imbalance in the FNC-1 dataset. Methods discussed include common pre-existing methods as well as a novel method for creating synthetic data samples.

[1] http://www.fakenewschallenge.org/

The remainder of this paper is structured as follows. Section 2 provides a concise review of the related literature. Section 3 discusses in more detail the relevant task and the data set used. Section 4 extensively presents our proposed approach to the problem. Section 5 describes the experiments designed to analyze the effectiveness of our method and discusses the results. Finally, Section 6 summarizes the most important findings and concludes with some recommendations for further research.

2. Literature Review

As described above, this paper studies a variant of the stance detection problem. Earlier work on this task frequently focused on target-specific stance classification, which is concerned with determining the attitude of a task towards a specified target, e.g. a person. Such work considered various domains, including debates (Hasan & Ng, 2013), student essays (Faulkner, 2014), or tweets (Augenstein et al., 2016). The latter proposes an method based on text encoding using an LSTM architecture (Hochreiter & Schmidhuber, 1997), which is a popular approach in stance detection problems. In 2016, the Emergent dataset was introduced (Ferreira & Vlachos, 2016), which comprises news articles relating to 300 rumoured claim. The task is to determine whether the headline of an article was for, against or observing the associated claim, which is closer to our problem. The authors propose a simple logistic regression classifier based on hand-crafted features extracted from both the headline and the claim. For a more comprehensive overview on the available literature on stance detection, the reader is referred to the recent survey by Küçük & Can, 2020.

The FNC-1 data set is a modification of Emergent, as further explained in Section 3. The winner of the associated competition used an ensemble of a convolutional neural network on word embeddings of both the head and body texts, and gradient boosted decision trees with external input features (Baird et al., 2017). The second team proposed an ensemble of five separately trained, randomly initialized MLP's, with solely external input features (Hanselowski et al., 2017). A team from UCL finishes third with a model they describe as a 'simple but tough-to-beat baseline', using a feature vector consisting only of two 5,000-dimensional term frequency vectors, of the head and body respectively, and the TF-IDF cosine similarity of the two vectors, passed through an MLP with one layer (Riedel et al., 2017). Although these teams achieve impressive overall accuracy (>88%), it should be noted that their performance is highly unbalanced. They all have accuracy <60% in the *agree* class and <10% for *disagree*.

After the competition finished, numerous researchers have continued using this data set for stance detection, and managed to obtain improved results. Bhatt et al., 2017 perform classification combining neural, statistical and external features. Their approach is very similar to that of the UCL team, but they add more external features and include skip-thought sentence embeddings (Kiros et al., 2015). Borges et al., 2019 encode head and body using bi-directional RNN's and combine these with numerous similarity features. The current state-of-the-art is set by Sepúlveda-Torres et al., 2021 who apply a large pre-trained language model, specifically RoBERTa (Liu et al., 2019). They also use a hierarchical classification approach and in both stages combine RoBERTa embeddings of the head and body with hand-crafted features.

We note that virtually all work on this data set develops classifiers that use as input both hand-crafted (external) features from the head and body texts, and embeddings of these texts, or exclusively the former. Our approach, on the other hand, is based solely on sentence embeddings and does not rely on any hand-crafted representations. More importantly, although others have introduced methods utilizing large pre-trained language models (Sepúlveda-Torres et al., 2021; Slovikovskaya, 2019), they have only applied models that have been trained for generic NLP tasks. To the best of our knowledge, no research is available that attempts to perform stance detection using (and combining) large transformer language models trained on specific tasks relevant to the problem, i.e. NLI and SIM. It has been repeatedly shown that such models are tremendously successful in these tasks (see e.g. Liu et al., 2019; Reimers & Gurevych, 2019; Jiang & de Marneffe, 2019).

3. Data set and task

As described above, we use the FNC-1 data set for classifying the stance of head-body pairs of news articles on 300 topics as either *agree*, *disagree*, *discuss* or *unrelated* (i.e. AGR, DSG, DSC, UNR). The associated stance detection task consists of learning a function that maps a head-body pair to a stance, i.e. $f : (h, b) \mapsto s$, with $s \in S = \{AGR, DSG, DSC, UNR\}$. This set-up for stance detection is an extension on the work of Ferreira & Vlachos, 2016. Within each topic, headlines and bodies are cross-paired, and for each pairing the appropriate stance is determined by human annotation. Moreover, unrelated head-body samples have been generated by randomly pairing heads with bodies from different topics. Approximately two thirds of the data is contained in the training set and the rest is saved for the test set. More details on how the data set has been constructed can be found on the website of FNC-1. An overview of the set size and distribution of stances is provided in Table 1.

Articles	Samples	AGR	DSG	DSC	UNR
2,587	75,385	7.4%	2.0%	17.7%	72.8%

Table 1. Size and stance distribution of FNC-1 complete data set (training and test). Each article consists of one head and body.

Crucially, it should be noted that the data is heavily unbalanced. The *unrelated* samples constitute over 70% of the data, while only 2% of samples are labelled as *disagree*. This could cause significant problems in training of the models. In Section 4.4 we discuss several methods to alleviate such issues.

To perform the stance classification task, we firstly pre-process the data by tokenizing the body texts into sentences. This is done using the `sent_tokenize` function from the `nltk.tokenize` package of NLTK, a well-known Python library for NLP (Bird & Loper, 2004).

4. Methodology

We propose a method based on applying large transformer language models specifically pre-trained on NLP tasks that are relevant to FNC-1. We use models from *Sentence-Transformers* (aka sBERT), a Python framework providing sentence-level BERT-based models trained on various tasks (Reimers & Gurevych, 2019). Specifically, the NLI embeddings are generated using a *RoBERTa* model (Liu et al., 2019) pre-trained on a number of NLI tasks[2], and the SIM embeddings are generated using a *MiniLM* (Wang et al., 2020) model trained with a self-supervised contrastive learning objective to identify similar sentences on a large number of sentence-level datasets[3]. Further specifications of the models and training procedures can be found in the sBERT paper (Reimers & Gurevych, 2019) and website[4]. In our training procedure, we have elected to completely 'freeze' both pre-trained models. This allows us to compute the NLI and SIM sentence embeddings for all heads and bodies as a pre-processing step, which massively reduces training time, and ultimately the size of our models. We denote these four sets of embeddings by SIM-head, NLI-head, SIM-body, and NLI-body, where the former two consist of one embedding each and the latter two consist of a number of embeddings equal to the number of sentences in the body. These four embedding sets constitute the only inputs used in our models. To enable efficient batch processing during training time, we set a fixed body length of 50. Shorter bodies are padded with all-zero embeddings and longer bodies are truncated to include only the first 50 embeddings. The number 50 was chosen based on the distribution of body lengths, a histogram for which is included in the Appendix (Figure 7).

As described in the introduction, we adopt a hierarchical classification approach. That is, given a head-body pair, we first predict whether they are *unrelated* or not, the model for which will be described in Section 4.1. For the second stage of classification, i.e. distinguishing between related stances (*agree, disagree, discuss*), we develop two similar models, as described in Section 4.2. We then concisely describe how these two stages are combined in Section 4.3 and finally propose some data augmentation strategies in Section 4.4.

[2]https://huggingface.co/sentence-transformers/nli-distilroberta-base-v2

[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[4]https://www.sbert.net/index.html

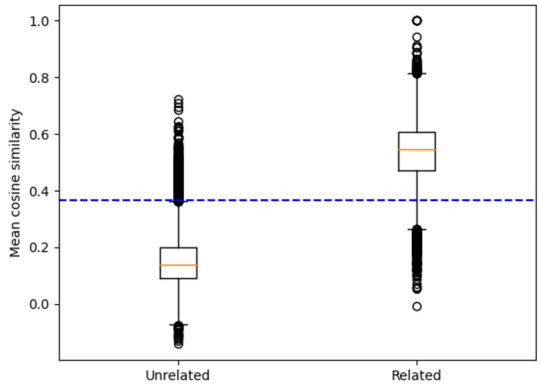

Figure 1. Boxplot of mean cosine similarity between headline and 5 most similar body sentences

4.1. Classifying Related-Unrelated: RelatedNet

For the first classification stage, i.e. determining whether a given head is at all related to a given article body, we propose a simple model based solely on the SIM embeddings of the two texts. Specifically, we compute the cosine similarity between SIM-head and each of the SIM-body embeddings. Then, we select the k most similar embeddings, concatenate these with SIM-head, and pass them through an MLP with four hidden layers with dropout (Srivastava et al., 2014) and ReLU activation functions, and a final Softmax layer for binary classification.

Even without running any experiments, we have strong reason to believe this method will be successful. Figure 1 shows a boxplot of the mean cosine similarity between SIM-head and the $k = 5$ nearest SIM-body embeddings, for unrelated and related pairs. The blue line shows the classification threshold that maximizes the F_1 score, which yields a score of $F_1 = 0.93$. Hence, this result can already be obtained without any training, using only the 'cold-start' embeddings of the pre-trained SIM model. The additional MLP classifier should enable further improvement of this result.

4.2. Classifying Agree-Disagree-Discuss

To address the task of the second classification stage, where we assume the input head-body pair to be related, we propose two models: Top-kNet and AgreemNet. Both are based on the following reasoning. By comparing the SIM-head embedding to the SIM-body embeddings, we can infer which body sentences are most similar to the head, and are thus most informative of the relation between the body and head. Then, we can consider the NLI-body embeddings of these sentences and the NLI-head embedding to classify the stance. The two crucial underlying assumptions of this method are (i) that body sentences which convey the central message of the article are close to the head in SIM embedding space, even though these sentences may contradict each other, and (ii) that when considering these strongly related sentences, their NLI embeddings will contain relevant information to determine their specific relation.

To support these assumptions and therefore underpin the argument for our models in general, consider the following head and body sentences, where the first two come from a training sample pair that is marked *agree* and the third from a *disagree* pair.

H: *Hundreds of Palestinians flee floods in Gaza as Israel opens dams.*
B1: *Hundreds of Palestinians were evacuated from their homes Sunday morning after Israeli authorities opened a number of dams near the border, flooding the Gaza Valley in the wake of a recent severe winter storm.*
B2: *Locals have continued to use it to dispose of their waste for lack of other ways to do so, however, creating an environmental hazard.*
B3: *Easily the most important part of this story is the fact that there are no such dams in Gaza.*

Table 2 shows the cosine similarity between the embeddings of the head and each of the body sentences in both NLI and SIM space. We observe that SIM embeddings correctly inform us that sentences B1 and B3 are (relatively) important w.r.t. the headline, whereas B2 is not. Moreover, in NLI embedding space, the headline is close to B1, but quite far from B3, corresponding to the respective *agree* and *disagree* labels.

	SIM	NLI
H, B1	0.802	0.826
H, B2	0.087	0.156
H, B3	0.632	0.389

Table 2. Cosine similarity between SIM and NLI embeddings of a head and three body sentences

4.2.1. Top-kNet: Most Similar Sentences

This model operates by first finding the k body sentences that are most similar to the head, and then using a concatenation of the NLI embeddings for those sentences as input to a Multi-layer Perceptron (MLP) to perform the *Agree-Disagree-Discuss* classification task. The MLP classifier consists of 5 fully-connected layers, with dropout (Srivastava et al., 2014) and ReLU activations used on each of the hidden layers. The dimensions for the first three hidden layers are controlled by a hyper-parameter $hdim_a$, and, similarly, the dimension of the final hidden layer is controlled by $hdim_b$. The other hyper-parameters for this model are the dropout probability, and k, the number of body sentences selected for the classifier. Figure 2 illustrates the TopKNet architecture.

4.2.2. AgreemNet: Averaging Over Sentences

This model takes a different approach, computing a similarity-weighted average of NLI-body embeddings as a representation of the body. This is done as in equation 1, where α_i is the similarity score in $[0, 1]$ between the i^{th} body sentence and the head sentence, $\mathbf{b}_i$ is the NLI embedding for the i^{th} body sentence, and there are $|B|$ sentences in the body in total.

$$\mathbf{b} = \sum_{i=1}^{|B|} \alpha_i \mathbf{b}_i \tag{1}$$

The body representation, $\mathbf{b}$, is then a vector in NLI embedding space, that can be directly compared to the NLI embedding of the head. This weighted average is computed using an attention layer, in which the SIM-head embedding is a *query*, the SIM-body embeddings are *keys*, and the NLI-body embeddings are *values*. Additional modelling flexibility is offered by the attention layer through linear transformations of the query, keys, and values, and the output embedding. Multiple body representations of this form are computed in parallel by AgreemNet using the multi-head scaled dot-product attention proposed in (Vaswani et al., 2017), which has been implemented in `pytorch`[5]. The concatenation of the body embeddings produced by the attention layer, the NLI-head embedding, and the cosine similarities between the NLI-head and each body embedding, is then passed to an MLP for classification. The MLP used in AgreemNet is similar to that of Top-kNet, but uses 4 fully-connected layers, with ReLUs and dropout after the hidden layers. The hyper-parameters for this model are then $hdim_a$, specifying the dimensions of the first two hidden layers, $hdim_b$ of the final hidden layer, the dropout probability, and the number of attention heads. See Figure 3 for a schematic illustration of the AgreemNet model architecture.

4.3. BaIT: Complete Hierarchical Stance Detection

Finally, to perform complete hierarchical stance detection at test time, we input the SIM and NLI embeddings of a head-body pair into RelatedNet. If this predicts the pair to be *unrelated*, this is returned as output and the procedure is terminated. Otherwise, the embeddings are fed into either AgreemNet or Top-kNet to obtain a final stance prediction of either *agree*, *disagree* or *discuss*. We referred to this combined model as BaIT, and an overview of the architecture is presented in Figure 4. As described above, the models for the two stages are trained separately, and the complete BaIT model is only applied at test time.

4.4. Data Augmentation

In order to combat the imbalanced nature of the FNC-1 data set, a number of data augmentation strategies were employed. The aim in applying these methods is mainly to improve accuracy on the under-represented *disagree* class, thus improving the recall on news articles that spread fake news.

4.4.1. Weighted Loss Function

This method involves adding a class-dependent weighting to the loss function so that mis-classifying some classes in-

[5] https://pytorch.org/docs/stable/generated/torch.nn.MultiheadAttention.html

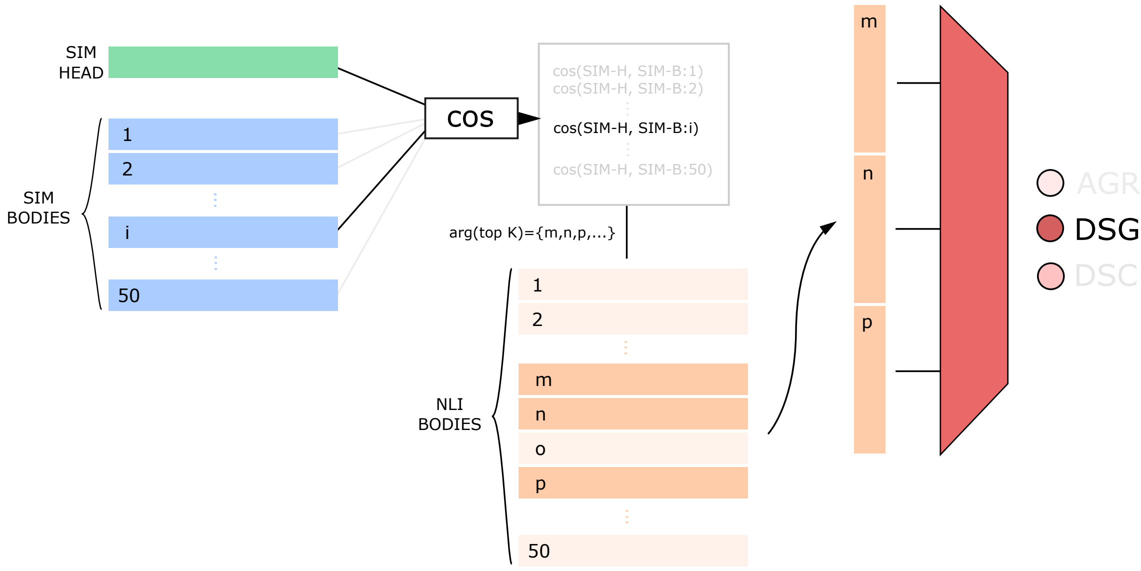

Figure 2. Schematic for TopKNet model

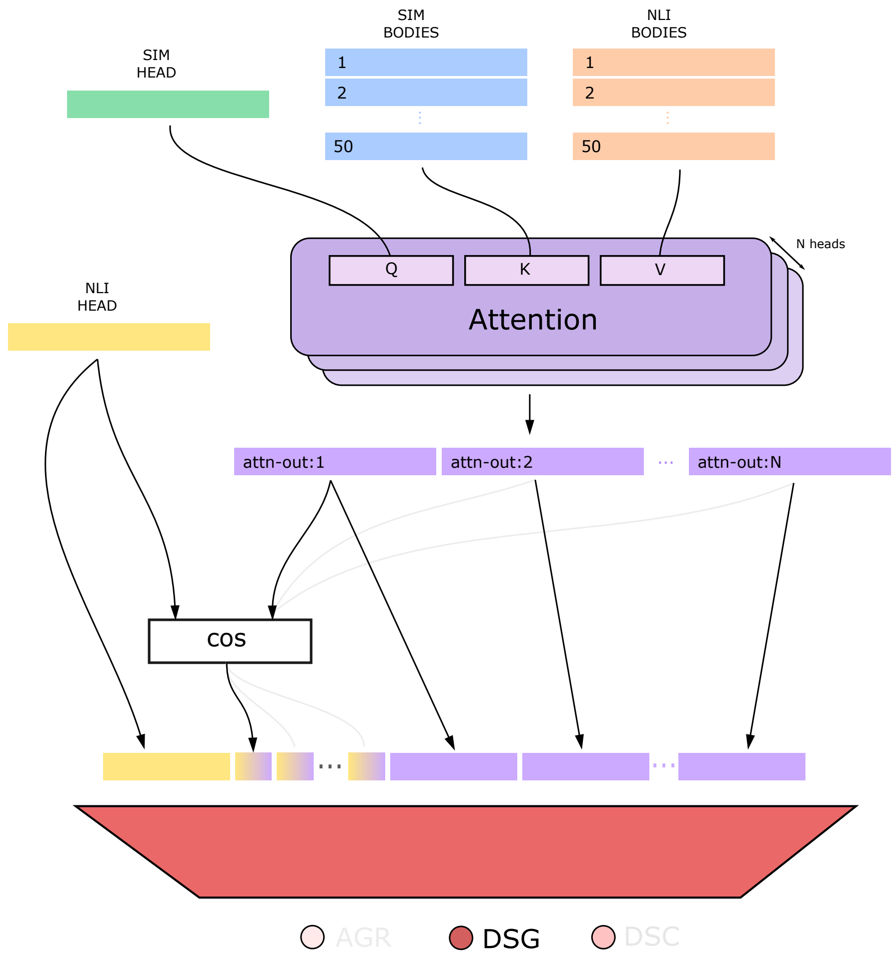

Figure 3. Schematic for AgreemNet model

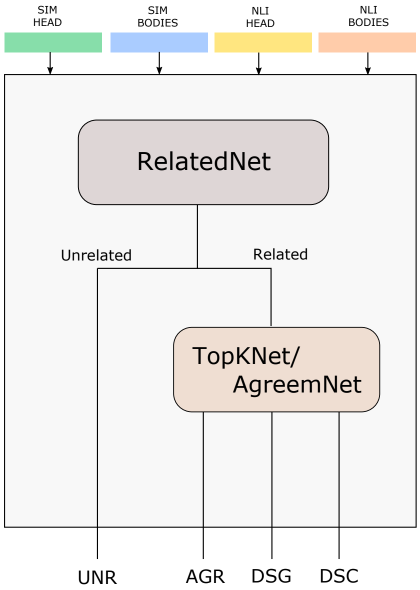

Figure 4. Schematic for hierarchical BaIT model

curs more loss than others. Assigning a higher weighting to under-represented classes in the dataset then helps prevent the model from over-predicting common classes, resulting in more balanced per-class accuracies. The class weights were computed using scikit-learn's (Pedregosa et al., 2011) `compute_class_weight` function [6].

4.4.2. Synthesis of Negated Headlines

An algorithm was developed to semantically negate headlines with the *agree* or *disagree* label, such that the correct label would flip, enabling us to generate more *disagree* samples from existing *agree* samples. This algorithm operates by sequentially attempting three methods of sentence negation, and returning the negated sentence produced by the first applicable method. The Stanford CoreNLP package was used for dependency parsing (Chen & Manning, 2014). The first method checks for the presence of the word *not*. If this *not* is functioning as a negation modifier (De Marneffe & Manning, 2008), then it is removed. The second method checks if the root verb of the sentence has an auxiliary verb and, if so, adds the word *not* afterwards. The third method finds the set of antonyms for the root verb using WordNet (Miller, 1995) and swaps the root with the antonym that results in the headline giving the highest language model probability, according to a distilled GPT-2 model (Radford et al., 2019). This method, when applied to the training set, generated an additional 1068 *disagree* samples, more than doubling the number disagree samples for training.

4.4.3. Extending the train set with ARC

In a retrospective analysis of the FNC-1 competition, it is suggested to evaluate the robustness and generalizability of the developed models by applying them to a different but similar data set (Hanselowski et al., 2018). Specifically, the authors suggest using the Argument Reasoning Comprehension (ARC) data set, which is constructed from the debate section of the New York Times, consisting of user posts relating to 188 debate topics (Habernal et al., 2017). To adapt this set to the stance detection task, a user post is taken as the article body and one of two associated opposing claims is taken as the headline. If the post is labelled as supporting the selecting claim, the sample is marked *agree*, if it supports the opposing claim the sample is *disagree*, and when it is labelled as supporting neither the sample is *discuss*. Similar to the FNC-1 set, *unrelated* samples are generated by pairing user posts with claims from different topics. An overview of this data set is presented in Table 3. Although there is again a majority of *unrelated* samples, the other stances have a much more balanced representation, containing substantially more *disagree* samples.

We investigate if we can improve the performance of our models by additional training on the ARC set. Although this implies additional training samples of the previously under-represented classes, it should be noted that the language used in the ARC set differs significantly from the FNC-1 set. The claims are generally much more factual, simpler sentences rather than news headlines (e.g. "*Same-sex colleges are still relevant*" vs "*The Greater Gaza Plan: Is Israel trying to force Palestinians into Sinai?*"), and the users posting to a debate section write differently than journalists. The ARC set is similar enough to FNC, however, that we believe training on a new dataset consisting of both FNC and ARC can help improve performance on the FNC task, particularly on the *disagree* class, which is much better represented in ARC. Note that this augmentation method, unlike the two methods described in Sections 4.4.1 and 4.4.2, uses external data, and therefore does not qualify for the FNC-1 challenge.

Articles	Samples	AGR	DSG	DSC	UNR
4,488	17,792	8.9%	10.0%	6.1%	75.0%

Table 3. Size and stance distribution of ARC complete data set (training and test). Each article consists of one head and body.

5. Experiments

A number of experiments were carried out to evaluate the performance of the models proposed in Sections 4.1 and 4.2. Details on the hyper-parameter tuning process and initial results on the test set are provided in Section 5.1. Further experimental results provided in Section 5.2 evaluate the effectiveness of each data augmentation method described in Section 4.4.

5.1. Model Selection

The FNC-1 training set was split into two distinct subsets for the purpose of hyper-parameter tuning: a train set, and a validation set. The validation set was created by taking

[6]https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

samples associated with a selected 30% of the set of training headlines, such that there is no overlap in headlines between the train and validation set. Hyper-parameter tuning was then carried out using Bayesian optimization, in which the relationship between the hyper-parameters and the unweighted average class accuracy is modelled as a Gaussian Process, with an *acquisition function* choosing new hyper-parameter values that are likely to improve the performance metric (Snoek et al., 2012). In addition to the hyper-parameters mentioned for each model in Section 4, the batch size and learning rate were also tuned. The parameter values selected for each model by the Bayesian optimization procedure are presented in Table 4. Using these hyper-parameters, the BaIT model was trained on the full FNC-1 training set and then evaluated on the test set using both TopKNet and AgreemNet (BaIT_K and BaIT_A, respectively) for comparison. Evaluation results are reported in Table 5, where they are presented alongside the results from the Simple but Tough To Beat (STTB) model (Riedel et al., 2017), which came 3rd on the official FNC-1 leaderboard. Confusion matrices for both BaIT_K and BaIT_A are presented in Figures 5 and 6. It is noteworthy that while the overall accuracy of STTB is higher, BaIT_K and BaIT_A achieve a higher accuracy on the *disagree* class, which is integral to identifying fake news in this task. Note also that BaIT_K has ~5 times fewer parameters than STTB: BaIT_K has 195,543 parameters, while STTB has 1,000,500.

	RelatedNet	TopKNet	AgreemNet
Batch size	32	64	128
Learning rate	10^{-4}	10^{-3}	10^{-3}
Dropout	0.277	0.301	0.105
$hdim_a$	600	60	60
$hdim_b$	600	60	20
k	4	3	–
Attn heads	–	–	11
Params.	2,235,602	195,543	2,203,679

Table 4. Optimized hyper-parameters for each model.

	BaIT_K	BaIT_A	STTB
AGR	36.8	36.5	44.0
DSG	11.3	9.9	6.6
DSC	53.6	53.9	81.4
UNR	95.5	95.5	97.9
Overall	81.4	81.4	88.5
FNC-1	68.2	68.2	81.7

Table 5. Per-class accuracy (%), overall accuracy (%), and FNC-1 score for our models and the STTB model (Riedel et al., 2017).

5.2. Data Augmentation

Each data augmentation method described in Section 4.4 was applied to both BaIT_K and BaIT_A models, using the optimized hyper-parameters specified in Table 4. The FNC-1 test scores achieved using these methods are given in Table 6 for the two BaIT variants. In general, the data augmentation methods were successful in improving the performance on the under-represented *disagree* class. The most effective augmentation method was including the ARC set in the training data, doubling the *disagree* accuracy for BaIT_K and almost tripling it for BaIT_A. Note, however, that this method involves adding external data, and so is in general far more costly than the other methods. The next biggest improvement resulted from using synthetic data with BaIT_A, which caused a doubling in performance. Weighting the *disgree* loss also hugely improved performance here, again almost doubling the *disagree* accuracy. Note that while the weighted loss resulted in slightly lower *disagree* accuracy than synthetic data, it had less negative impact on the overall accuracy, reducing it by ~1%, rather than ~3%. Interestingly, the weighted loss and synthetic data methods caused a decrease in the *disagree* accuracy in the BaIT_K model, perhaps due to a lack of modelling flexibility resulting from its small size. Using these methods, we can achieve 3 times the *disagree* accuracy of STTB with only 7% less overall accuracy, by using synthetic data and the BaIT_K model, which only requires the original FNC-1 training data and thus fits the original criteria for the challenge. By extending the dataset using ARC, this improvement grows to 4 times the *disagree* accuracy of STTB.

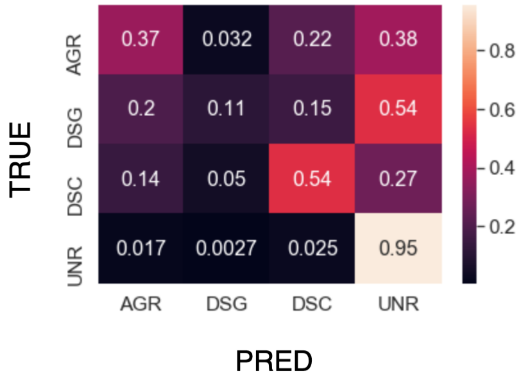

Figure 5. Confusion matrix for BaIT_K model.

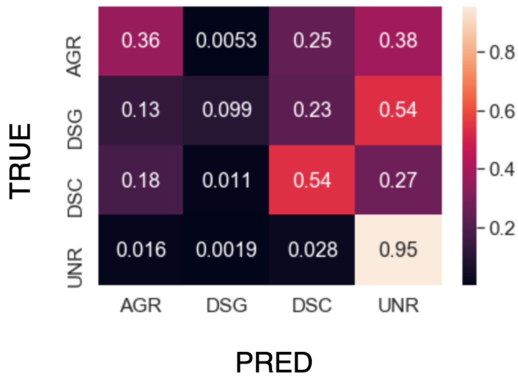

Figure 6. Confusion matrix for BaIT_A model.

	BaIT$_A$			BaIT$_K$		
	Weighted Loss	Synthetic Data	+ARC	Weighted Loss	Synthetic Data	+ARC
AGR	43.5	36.5	54.7	44.1	47.4	39.5
DSG	9.8	1.0	17.4	17.8	18.2	24.4
DSC	46.0	32.2	63.7	41.0	30.4	73.0
UNR	95.5	94.4	90.8	95.5	94.4	90.8
Overall	80.5	78.0	81.3	79.9	77.5	82.0
FNC-1	66.8	64.4	75.6	65.7	63.6	76.7

Table 6. Per-class accuracy (%), overall accuracy (%), and FNC-1 score for the two BaIT variants, using a number of data augmentation strategies.

6. Conclusions

Proposed new method of approaching FNC task using pre-trained encoders, which greatly reduces the number of trainable parameters required. Proposed two architectures, one simple smaller model, BaIT$_K$, and one larger more complex model, BaIT$_A$, both of which achieve similar performance on the initial task, and competitive results with existing baselines, particularly on the *disagree* class, which is key to this challenge. Explored a number of methods in data augmentation in attempt to further improve *disagree* performance and overcome class imbalance in the dataset. This included the proposal of a novel data augmentation method, involving an algorithm that generates negated versions of headlines to produce new samples of the under-represented class. These methods were broadly successful, particularly when used with the more flexible BaIT$_A$ model. Future work could look towards experimenting with other SIM and NLI pre-trained models to compare effectiveness of knowledge transfer to the FNC task. Additionally, more sophisticated methods in negated sentence generation could further tackle class imbalance in such datasets.

References

Augenstein, Isabelle, Rocktäschel, Tim, Vlachos, Andreas, and Bontcheva, Kalina. Stance detection with bidirectional conditional encoding. *arXiv preprint arXiv:1606.05464*, 2016.

Baird, Sean, Sibley, Doug, and Pan, Yuxi. Talos targets disinformation with fake news challenge victory. *Fake News Challenge*, 2017.

Bhatt, Gaurav, Sharma, Aman, Sharma, Shivam, Nagpal, Ankush, Raman, Balasubramanian, and Mittal, Ankush. On the benefit of combining neural, statistical and external features for fake news identification. *arXiv preprint arXiv:1712.03935*, 2017.

Bird, SG and Loper, Edward. Nltk: the natural language toolkit. Association for Computational Linguistics, 2004.

Borges, Luís, Martins, Bruno, and Calado, Pável. Combining similarity features and deep representation learning for stance detection in the context of checking fake news. *Journal of Data and Information Quality (JDIQ)*, 11(3): 1–26, 2019. Publisher: ACM New York, NY, USA.

Busioc, Costin, Ruseti, Stefan, and Dascalu, Mihai. A Literature Review of NLP Approaches to Fake News Detection and Their Applicability to RomanianLanguage News Analysis. *Revista Transilvania*, (10), 2020.

Chen, Danqi and Manning, Christopher D. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.

De Marneffe, Marie-Catherine and Manning, Christopher D. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

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

Faulkner, Adam. Automated classification of stance in student essays: An approach using stance target information and the Wikipedia link-based measure. In *The Twenty-Seventh International Flairs Conference*, 2014.

Ferreira, William and Vlachos, Andreas. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL, 2016.

Habernal, Ivan, Wachsmuth, Henning, Gurevych, Iryna, and Stein, Benno. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. *arXiv preprint arXiv:1708.01425*, 2017.

Hanselowski, Andreas, Avinesh, P. V. S., Schiller, Benjamin, and Caspelherr, Felix. Description of the system developed by team athene in the fnc-1. *Fake News Challenge*, 2017.

Hanselowski, Andreas, PVS, Avinesh, Schiller, Benjamin, Caspelherr, Felix, Chaudhuri, Debanjan, Meyer, Christian M, and Gurevych, Iryna. A retrospective analysis of the fake news challenge stance detection task. *arXiv preprint arXiv:1806.05180*, 2018.

Hasan, Kazi Saidul and Ng, Vincent. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the sixth international joint conference on natural language processing*, pp. 1348–1356, 2013.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jiang, Nanjiang and de Marneffe, Marie-Catherine. Evaluating bert for natural language inference: A case study on the commitmentbank. 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)*, 2019.

Kiros, Ryan, Zhu, Yukun, Salakhutdinov, Russ R, Zemel, Richard, Urtasun, Raquel, Torralba, Antonio, and Fidler, Sanja. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

Küçük, Dilek and Can, Fazli. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1):1–37, 2020. Publisher: ACM New York, NY, USA.

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

Miller, George A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Pedregosa, Fabian, Varoquaux, Gaël, Gramfort, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier, Blondel, Mathieu, Prettenhofer, Peter, Weiss, Ron, Dubourg, Vincent, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, Sutskever, Ilya, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Reimers, Nils and Gurevych, Iryna. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

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

Sepúlveda-Torres, Robiert, Vicente, Marta, Saquete, Estela, Lloret, Elena, and Palomar, Manuel. Exploring summarization to enhance headline stance detection. In *International Conference on Applications of Natural Language to Information Systems*, pp. 243–254. Springer, 2021.

Slovikovskaya, Valeriya. Transfer learning from transformers to fake news challenge stance detection (fnc-1) task. *arXiv preprint arXiv:1910.14353*, 2019.

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vosoughi, Soroush, Roy, Deb, and Aral, Sinan. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018. Publisher: American Association for the Advancement of Science.

Wang, Wenhui, Wei, Furu, Dong, Li, Bao, Hangbo, Yang, Nan, and Zhou, Ming. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

7. Appendix

All code can be found at https://github.com/OisinNolan/fakenewschallenge.

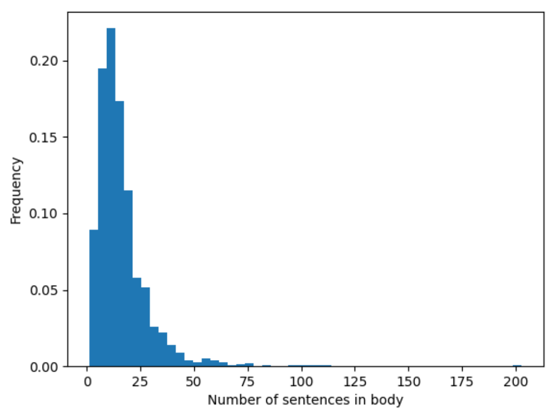

Figure 7. Histogram of number of sentences in article bodies