Document-Level Stance Classification for Fake News Detection

Anonymous ACL submission

Abstract

Document-level stance classification is a crucial first step of fake news detection. In this problem setting, the system should decide if a given document ”agrees”, ”disagrees”, ”discusses” or is ”unrelated” to a given text snippet that is to be validated. The recently launched Fake News Challenge has stressed upon the task by attempting to provide a large-scale dataset for training and evaluating the corresponding systems. The challenge attracted much attention from the community: over 50 registered participants. In this paper, we critically assess high performing models on the task, the dataset itself, present a model which achieves state-of-the-art results and evaluate the performance of successful models on a second dataset.

1 Introduction

Stance detection can be generally defined as the problem of determining the relative perspective of a source text entity with respect to a target text entity. The source text entity may ”agree” or ”disagree” with the target text entity or do not express a stance at all. Stance detection is helpful for a variety of different tasks such as the analysis of online debates (Walker et al., 2012; Sridhar et al., 2014; Somasundaran and Wiebe, 2010) or determining the veracity of rumors on twitter (Lukasik et al., 2016; Derczynski et al., 2017). Moreover, stance detection is also considered as an important first step in fake news detection, and it was therefore chosen as the first task to be tackled in the Fake News Challenge (FNC) (Pomerleau and Rao, 2017). The FNC was launched in order to foster the development of AI technology to help solve the fake news problem (Pomerleau, 2017).

Figure 1: Sample Headline, and text snippets from document bodies with respective stances.

The FNC has received much attention in the NLP community and 50 teams from academia and industry have participated in the stage one of the challenge (FNC-1). In the competition, the stance of the body text of an news article had to be determined with respect to a headline. As shown in an example in Figure 1, the body text may ”agree” or ”disagree” with the headline, only ”discuss” the topic of the headline, or be completely unrelated to it. Compared to other stance detection problem settings, in which the stance of a tweet with respect to a target entity (Mohammad et al., 2016), a premise with respect to a claim (Stab and Gurevych, 2017), or a blog post with respect to a target entity (Walker et al., 2012) needs to be determined, the FNC-1 stance detection task is more difficult, as the stance of the whole document needs to be identified. The document may contain opposing statements and only as a whole lean towards a certain stance.

In this paper, we test and analyze numerous models and features for the document-level stance detection task. Based on these insights, we propose a new model which reaches a new state-of-the-art. Moreover, we crucially assess high performing models, the problem setting, the dataset itself, and evaluate successful models on a second dataset. Based on our analyses, we report the following findings.

We have found that even the best performing systems on the FNC dataset, which reach about 0.82 on the FNC metric, achieve a relatively low F1 macro score of about 0.6. On the basis of our analysis we conclude that the FNC metric is problematic, since it does not take the unbalanced distribution for all classes of the FNC dataset into account. We have analyzed why the systems reach a relatively low F1 macro score and identified three main causes. 1. The dataset is unbalanced and there are only few instances for certain classes. Experiments on a more balanced corpus have shown that the models can distinguish between "agree" and "disagree" instances more successfully. 2. The task is challenging and the human upper bound is relatively low with 0.754 F1 macro. 3. The best performing models are using mostly similarity based features and are therefore not able to resolve more difficult cases, such as complex negation instances.

Our code including the new introduced models, the implementation of the features and the corpora is publicly available[1].

2 Related Work

The stance detection problem is broadly defined and it encompasses a number of problem settings, in which the stance of a source text entity with respect to a target text entity is determined.

Stance detection has been used in (Walker et al., 2012; Sridhar et al., 2014; Somasundaran and Wiebe, 2010), for the analysis of online debates, where the relative perspective of user posts with respect to a certain topic is determined. In these studies, structural and linguistic features, sentiment polarity features, and a lexicon with positive/negative arguing expressions, such as "I am convinced" or "certainly not", are used for the classification.

Within the field of computational argumentation, Stab and Gurevych (2017) address the problem of identifying argumentative relations, such as "support" or "attack", between premises and claims. They have identified unigrams, syntactic features, discourse features, and shared nouns between premise and claim to be most valuable for the task.

In SemEval-2016 Task 6a (Mohammad et al., 2016), the stance of the author of a tweet with respect to a target entity had to be classified as "against", "neutral" or "in favor". Zarrella and Marsh (2016) proposed the best system using an LSTM (Hochreiter and Schmidhuber, 1997) with word2vec embeddings (Mikolov et al., 2013). However, no team was able to beat the SVM baseline, using word/character n-grams as features.

Ferreira and Vlachos (2016) derived a dataset from the digital journalism project Emergent, which was also used for the construction of the FNC dataset. They used logistic regression classifier with hand-engineered features for the detection of the stance of article headlines with respect to a claim. Their system outperforms the textual entailment platform Excitement (Magnini et al., 2014), which was considered as a reasonable baseline for the task.

The discussed studies have focused on different stance detection problems, however, none have addressed the problem of detecting the stance of a whole document w.r.t a statement, which is discussed in this paper. Even though there are a number of publications concerned with the FNC-1 (Riedel et al., 2017; Thorne et al., 2017; Bourgonje et al., 2017; Stanford, 2017), the authors have mostly focused on model development without analyzing the document-level stance detection task or the FNC dataset at depth.

3 Stance detection corpora

In this study, we consider, the FNC dataset and the Argument Reasoning Comprehension (ARC) dataset (Habernal et al., 2017).

[1] https://github.com/... (We are going to publish the code with the paper)

Dataset	topics	documents	instances	agree	disagree	discuss	unrelated
FNC Train	200	1683	49972	7.4%	1.7%	17.8%	73.1%
FNC Test	100	904	25413	7.5%	2.7%	17.6%	72.2%
ARC	188	4448	17792	8.9%	10.0%	6.1%	75.0%

Table 1: Corpus statistics & label distribution for the FNC and ARC datasets

3.1 FNC dataset

The FNC corpus was almost entirely derived from the Emergent project (Silverman, 2017). The corpus consists of 300 claims, for each of which 5 to 20 related articles have been collected, resulting in a corpus of 2,595 documents. Since each claim discusses a different issue, the corpus can be viewed as representing information about 300 topics. The journalists hand-annotated the stances of the articles with respect to the claim as "agree", "disagree" and "discuss" and summarized each article into a headline.

The FNC organizers further modified the corpus in order to adjust it to the FNC-1 problem setting. For each claim, they matched every related article with every related headline. If both headline and body were agreeing with the claim, they were labeled as agreeing with each other. The agree label was also given if both disagreed with the claim. If the stance of the headline was opposite to the stance of the body, the pair was labeled as disagree. If either the headline or the body was labeled as discuss, the pair was labeled as discuss.

The dataset was split into 200 claims (topics) with associated headlines and bodies as the training dataset and 100 claims (topics) with its headlines and bodies as the testing dataset. To generate the unrelated class, headlines and bodies belonging to different claims are randomly matched; the data from the testing and the training set was kept separate to avoid the same headlines or bodies appearing in both sets. Thus, there is no overlap between the topics in the two datasets. In order to prevent teams from using any unfair means, by using the labels of the testing set from the Emergent project (which is publicly available), the organizers additionally created 266 instances. The statistics and the label distribution of the corpus are illustrated in Table 1.

3.2 ARC dataset

In order to evaluate the best-performing system on a second corpus, we select the dataset introduced by Habernal et al. (2017).The corpus was built by manually selecting 188 debates with popular questions from the user debate section of the New York Times. For each debate they created two opposing claims about the discussed topic and collected high-ranked comments. The corpus was annotated by crowd workers, who had to choose for each comment between the two opposing claims or select the no-stance option.

Example from the original ARC dataset:

Topic	Do same-sex colleges play an important role in education, or are they outdated?
Comment	Only 40 women's colleges are left in the U.S. And, while there are a variety of opinions on their value, to the women who have attended ... them, they have been ... tremendously valuable. ...
Claim 1	Same-sex colleges are outdated
Claim 2	Same-sex colleges are still relevant
Label	Same-sex colleges are still relevant

Generated instance:

Stance	Headline	Article body
agree	Claim 1	Comment

Table 2: ARC dataset modification

In order to align the corpus to the FNC stance detection problem, we modified the ARC dataset. It was assumed that the comments are always related to the two opposing claims. One of the two claims has been randomly selected as the headline and the comment as the article body. In fact, typically, the comments express an opinion in several sentences and can therefore be considered as documents. If the randomly chosen claim was also selected by the workers, we consider the claim-comment pair as agreeing with each other. If the opposite claim was selected, we labeled the pair as disagree. If none of the claims were selected and the no-stance options was selected by the workers, the comment was considered as discussing the claim. An example of a generated instance is shown in Table 2.

In order to generate the unrelated instances, we randomly match the comments with claims, thereby avoiding that a comment being assigned to a claim from the same topic. The statistics of the resulting corpus are given in Table 1.

4 Performance evaluation

4.1 Evaluation metric

The performance measurement for the FNC-1 was defined hierarchically. Firstly, 0.25 points are given if the article was correctly classified as "related or "unrelated" to the headline. If the article is "related" to the headline, 0.75 additional point are assigned if the model correctly classified the article-headline pair as "agrees", "disagrees" or "discuss". Thus, the large number of unrelated instances is balanced by the weights. Nevertheless, the metric fails at taking into account the unbalanced distribution of the three related classes (Table 1). Thus, models, which perform well on the majority class and poorly on the minority classes are favored. In fact, if one correctly classifies the "related" and "unrelated" instances, which is not difficult as the best systems are reaching about 0.99 F1 score on the task, and then simply predicts the "discuss" class, which is the majority of the three related classes, one reaches an FNC score of 0.833. Using this approach it would be sufficient to win FNC-1. Therefore, for our experiments we report F1 scores.

4.2 Human upper bound

4.2.1 FNC dataset

In order to be able to compare human and machine performance, five subjects labeled 200 instances. The overall inter-annotator agreement is relatively high reaching 0.686 Fleiss' κ (Fleiss, 1971). However, when evaluating the agreement only for the three related classes, by simply dropping the unrelated instances, Fleiss' κ dramatically reduces to 0.218. This indicates that differentiating between the three related classes is difficult even for humans.

On the basis of the annotation, we have also determined the most probable labels according to MACE (Hovy et al., 2013), and compared them to the ground truth from the Emergent project. The agreement of the labels in this case is better, reaching an overall Fleiss' κ of 0.807 and 0.552 for the three related classes. On the basis of the annotation according to MACE, we have computed the human upper bound which is reported in Table 3. However, this only can be an approximate limit, as our subjects are not expert annotators.

	agr	dsg	dsc	unr	F1m
FNC	.588	.667	.765	.997	.754
ARC	.710	.857	.571	.954	.773

Table 3: Human performance on the FNC and ARC dataset, agr = agree, dsg = disagree, dsc = discuss, unr = unrelated, F1m = F1 macro

4.2.2 ARC corpus

Also for the ARC dataset, subjects hand-annotated 200 samples to determine an approximate human upper bound. Even though the overall Fleiss' κ score of 0.614 is slightly lower compared to the FNC corpus, the agreement for the related class is higher with a Fleiss' κ score of 0.383. Also in this case, we determine the most probable labels according to MACE and compare them with the ground truth. The resulting overall Fleiss' κ score is 0.708, and for the three related classes it is 0.481. The class-wise F1 scores and F1 macro are displayed in Table 3.

5 Development of models and features

We propose models and features for the document-level stance detection task, which are evaluated in the subsequent section.

5.1 Features

To capture the characteristics of the headlines and bodies, we developed features based on related work on fake news detection, as well as stance detection. Some of the features are taken from the baseline implementation of the organizers of the FNC-1. The features are split into several groups, which are briefly explained below, with a detailed description in the supplement material at A.1.

BoW/BoC features: We use bag-of-words (BoW) 1- and 2-grams and add a negation tag to words that appear after a special negation keyword, based on a technique by Das and Chen (2007). For the bag-of-characters (BoC) 3-grams are used. For the BoW/BoC features, we create TF vectors for headline and body and concatenate them. The FNC-1 baseline feature *co-occurrence* counts occurrences of word n-grams, character n-grams, and stop words of the headline.

Topic model features: We use non-negative matrix factorization (NMF) (Lin, 2007), latent semantic indexing (LSI) (Deerwester et al., 1990), and latent Dirichlet allocation (LDA) (Blei et al., 2003) to create topic models. For each topic model a different feature is created. We extract 300 topics, compute the similarity of the headline and body to the found topics, and use the resulting vectors as features by either concatenating or calculating the cosine similarity between them.

Lexicon-based features: These features are based on the NRC Hashtag Sentiment and Sentiment140 lexicon (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014), on the MPQA lexicon (Wilson et al., 2005), MaxDiff Twitter lexicon (Rosenthal et al., 2015; Kiritchenko et al., 2014), and the EmoLex lexicon (Mohammad and Turney, 2010, 2013). The lexicons hold values signaling the sentiment/polarity for each word. For headline and body separately, we implement eight different features proposed by Mohammad et al. (2013). For the EmoLex lexicon, we count the emotions listed for each word of the headline/body that is found in the lexion. Lastly, the FNC-1 baseline features *polarity words* and *refuting words* are added. The first one counts refuting words (e.g. ”fake”, ”hoax”), divides the counter by two, and takes the remainder as a feature signaling the polarity of headline or body. The latter one sets a binary feature for each refuting word (e.g. ”fraud”, ”deny”) appearing in the texts.

Readability features: We measure the readability of headline and body with SMOG grade , Flesch-Kincaid grade level, Flesch reading ease, and Gunning fog index (Štajner et al., 2012), Coleman-Liau index (Coleman and Liau, 1975), automated readability index (Senter and Smith, 1967), LIX and RIX (Anderson, 1983), McAlpine EFLAW Readability Score (McAlpine, 1997), Strain Index (Solomon, 2006).

Lexical features: As lexical features we implement the type-token-ratio (TTR) and the measure of textual lexical diversity (MTLD) (McCarthy, 2005) for the body, and only TTR for the headline. The FNC-1 baseline feature *word overlap* divides the cardinality of the intersection of unique words in headline and body by the cardinality of the union of unique words in headline and body.

POS features: The POS features include counters for different POS-tags, and also the percentage of stop words and the number of verb phrases, which showed good results in the work of Horne and Adali (2017). For the *word-similarity* feature, we calculated average word embeddings (pre-trained word2vec model[2]) for all verbs (retrieved with Stanford Core NLP toolkit[3]) of headline/body separately. The cosine similarity between the averaged embeddings of headline and body is taken as a feature, as well as the Hungarian distance between verbs of headline and body based on the paraphrase database[4]. The same computation is repeated for the nouns.

Structural features: The structural features contain the average word length of the headline and body, and the number of paragraphs and average paragraph length of the body.

5.2 Models

5.2.1 Baseline models

The following two models reach highest performance on the FNC dataset and therefore serve as a baseline for our experiments.

Talos Intelligence model (TalosComb) Baird et al. (2017) reached state-of-the-art results on the FNC dataset according to the FNC-metric. They use 50/50 weighted average of a deep convolutional neural network (TalosCNN) and gradient-boosted decision trees model (TalosTree). TalosCNN is based on pre-trained word2vec embeddings[2] which are passed through several convolutional layers followed by three fully-connected layers and a final output layer with four neurons for classification. TalosTree is based on word count features, TF-IDF features, singular-value decomposition features, pre-trained word2vec embeddings[2] and sentiment features.

UCL-model (uclMLP) Riedel et al. (2017) implemented a multi-layer perceptron (MLP) with one hidden layer which also reaches high performance on the FNC dataset. As features they use BoW unigrams by creating a vocabulary of the 5,000 most important words from the development set and defining TF vectors of headline and body with this vocabulary. Also, they define another BoW unigram feature, but add the tokens of the test set and use TF-IDF instead of TF in order to find the most important words. The resulting TF

[2] https://code.google.com/archive/p/word2vec/
[3] https://stanfordnlp.github.io/CoreNLP/
[4] http://www.cis.upenn.edu/ ccb/ppdb/

feature vectors of headline and body are concatenated and a single-value entry is added representing the cosine similarity of the two TF-IDF vectors.

5.2.2 Implemented models

Feature based MLP (featMLP): We constructed a MLP using as an initial configuration the hyperparameters suggested by Davis and Proctor (2017). Based on this initial configuration, we performed a random search on the development set in order to further optimize the hyperparameters with regard to the developed features. The identified hyperparameters are as follows: Optimizer: Adam (Kingma and Ba, 2014), learning rate: .001, batch size: 188, 7 hidden layers: 362, 942, 1071, 870, 318, 912, 246 units per layer, dropout: none, bias initialization: .001, weight initialization: method proposed by He et al. (2015).

Stacked LSTM model (stackLSTM): We implemented a stacked LSTM with Keras (Chollet et al., 2015). For this model, we use 100-d GloVe word embeddings[5] (Pennington et al., 2014), concatenate the LSTM's output with the final features determined in section 6.1, and add three dense layers with 600 neurons each before computing the class probabilities.

Avg. Pooled CNN (avgCNN): This CNN architecture consists of average-pooled layers after a 1-D convolution with filter sizes of 3, 5, and 7. It is optimized with batch-normalization using an Adam Optimizer with a learning rate of 0.0001.

Weighted MLP (weightMLP): This is a hierarchical-weighted densely connected custom architecture. The average sentence embedding of the headline is used to weight the sentences from the document bodies. The weighted body embeddings are concatenated with the headline embedding and feed into a densely connected hidden-layer of size 2000. The network is optimized using batch-normalization, and, since the classes are imbalanced, we use weighted-categorical cross-entropy as a loss function to optimize the parameters of the model.

Additionally, we are using the classifiers Naive Bayes (NaiveB), Gradient Boost (GradBoost), Logistic Regression (LogReg), and SVM from the *sklearn* library (Pedregosa et al., 2011).

[5] http://nlp.stanford.edu/data/glove.twitter.27B.zip

	agr	dsg	dsc	unr	F1m
Baselines:					
major. vote	0.0	0.0	0.0	.835	.209
FNC-1	.241	.047	.738	.970	.499
Only:					
BoW/BoC	**.772**	.601	.874	.991	.796
Topic	.637	.571	.838	.983	.757
POS	0.0	0.0	.731	.964	.425
All w/o:					
BoW/BoC	.665	.530	.841	.982	.754
Topic	.714	.598	.863	.989	.791
POS	.722	**.616**	**.876**	**.995**	**.802**
All feat. †	.713	.573	.870	.993	.787
All feat.	.675	.455	.835	.989	.738

Table 4: Results of the feature ablation test on the development set with 10-fold cross-validation. Baseline *FNC-1* is calculated with gradient boosting classifier and all FNC-1 baseline features. † states that only the preselected features are used (see Table 6 in A.1). (agr = agree, dsg = disagree, dsc = discuss, unr = unrelated, F1m = F1 macro).

6 Experiments

In this section, we perform experiments with the implemented models and features in order to identify the best performing configuration.

6.1 Feature selection

Preliminary experiments have shown that the MLP model outperforms all the other models. We therefore use the MLP for the feature ablation test in order to find the best feature set for our experiments. All tests are performed on the development set with 10-fold cross-validation. We grouped the features according to the feature type in eight different groups. Features that have much lower scores than others in their group are taken out and listed individually. On the basis of preliminary tests, we decided that features more than 15% below the FNC-1 baseline should be omitted. We have found that they mostly just predict the majority class and thus lower the score. We mark all features that are used for the following feature ablation test with † in Table 6 of A.1.

The results of the ablation test (see Table 4) reveal that the BoW/BoC features have the biggest impact, and the performance can be further improved by the topic features. Adding the POS features lowers the score. Hence, the final feature set will consist of the BoW/BoC and topic model features.

6.2 Model experiments

In Table 5 our implemented models are compared with *sklearn* classifier, which are using the best feature set described above, and various baselines. Here we only report F1 scores on the testing set (FNC metric scores can be found in the appendix A.4). It has been observed that the performance of the systems decreases from about 0.8 on the development set, to 0.6 F1 Macro on the test set. The drop of performance is most likely because of the 100 new topics represented in the testing set. As can be observed, the TalosComb model is in this case not superior and is slightly outperformed by the uclMLP. The analyses of the confusion matrix has shown that the model mostly predicts for the majority classes, which is also the reason why the performance on the ”disagree” class is low. The same problem could be observed for the *sklearn* classifiers which are therefore not competitive in terms of F1 macro. From our models, stackLSTM performs best, outperforming the strongest baseline model uclMLP by more than two percentage points. Nevertheless, stackLSTM is not significantly better than the featMLP. The advantage of the two models is that they better perform on the ”disagree” class. An ensemble of the featMLP, TalosComb, and uclMLP could not further significantly improve performance.

As it can be noticed in the table, all models have difficulties predicting the ”disagree” class, which is probably because of the few number of instances for this class. To address this issue, we have applied different under-sampling and over-sampling techniques. However, this did not help to improve performance.

6.3 Experiments on the ARC dataset

In order to analyze how far the developed models are able to generalize to a similar problem settings, we investigate the performance of the models on the ARC corpus. For experiments on the ARC we have chosen only our featMLP and the two baseline models uclMLP and TalosComb. The results, listed in Table 5, show that the performance of all models decreases. Nevertheless, they are still better able to distinguish between ”agree” and ”disagree” instances compared to the FNC-1 corpus. We assume this is because the corpus is more balanced. However, here, the classification of the discuss instances is more difficult. This is because, even though the user comments are related to the claim, they often do not explicitly refer to it. On this corpus, the TalosComb outperforms the other models on all classes. We assume, the difference

Model Experiments:

	agr	dsg	dsc	unr	F1m
Baselines:					
major. vote	0.0	0.0	0.0	.839	.210
TalosTM	.520	.003	.762	.994	.570
TalosCNN	.258	.092	0.0	.882	308
TalosComb	**.539**	.035	.760	.994	.582
uclMLP	.479	.114	.747	.989	.583
Class.:					
NaiveB	.180	.024	.350	.576	.283
GradBoost	.365	.027	.750	.983	.531
LogReg	.449	.003	**.773**	.979	.551
SVM	.497	.022	.738	.984	.561
Proposed:					
avgCNN	.202	.144	.325	.747	.355
weightMLP	.460	.002	.673	.963	.525
featMLP	.530	.151	.766	.982	.607
stackLSTM	.501	**.180**	.757	**.995**	**.609**
upp. bound	.588	.667	.765	.997	.754

ARC dataset and cross-domain experiments:

	agr	dsg	dsc	unr	F1m
ARC-ARC					
major. vote	0.0	0.0	0.0	.857	.214
TalosComb	**.576**	**.584**	**.183**	**.944**	**.576**
uclMLP	.517	.503	.121	.932	.519
featMLP	.526	.506	.144	.934	.526
upp. bound	.710	.857	.571	.954	.773
ARC-FNC					
major. vote	0.0	0.0	0.0	.857	.214
TalosComb	**.376**	**.279**	**.113**	**.977**	**.376**
uclMLP	.288	.234	.109	.728	.288
featMLP	.322	.111	.033	.939	.351
upp. bound	.710	.857	.571	.954	.773
FNC-ARC					
major. vote	0.0	0.0	0.0	.839	.210
TalosComb	.348	0.0	**.188**	**.928**	.366
uclMLP	**.352**	**.258**	.063	.898	.352
featMLP	.321	.159	.171	.906	**.389**
upp. bound	.588	.667	.765	.997	.754

Table 5: Model experiments, ARC dataset and cross-domain experiments reported in F1 (ARC-ARC: train and predict on ARC, ARC-FNC: train on ARC predict for FNC, FNC-ARC: train on FNC predict for ARC, agr = agree, dsg = disagree, dsc = discuss, unr = unrelated, F1m = F1 macro, upp. bound = human upper bound)

of the news domain genre of the FNC dataset with respect to the user debate forum genre from the ARC is one factor for the different performance.

The cross corpus experiments show that the performance of the models is substantially better than the majority vote baseline. It can be therefore concluded that the two problem settings are related and exhibit a common structure. The results suggest that TalosComb is best able to learn from the ARC corpus as it is also superior in the **ARC-FCN** setting. The featMLP, on the other hand, yields best results when trained on the FNC corpus as the **ARC-FCN** setting suggests.

6.4 Error analysis

In the error analysis, which was performed for the top performing models, we have made the following observations. If there is lexical overlap between headline and body, the models classify the instance as one of the related classes, even in cases in which the headline and body are unrelated (Appendix A.3 Example 1). If the body and the headline are ”related” but do not contain the same tokens but synonyms, the model often classifies the case as ”unrelated” (Appendix A.3 Example 2). If keywords like ”reports”, ”said”, ”allegedly” are detected, the systems classify the case as ”discuss” (Appendix A.3 Example 3). The ”disagree” class is difficult to determine as only few lexical indicators such as ”false”, ”hoax”, ”fake” are available as features. The disagreement is often expressed in complex terms which demands more sophisticated techniques (Appendix A.3 Example 4).

7 Discussion of the results

The experiments show that even the best performing models on the FNC-1 dataset reach a relatively low F1 macro score of about 0.6, even though scoring high on the FNC metric. From our perspective, the FNC metric is problematic, since it does not take the unbalanced class distribution for the three related classes into account. On the basis of our experiments we conclude that the low performance is caused by the following problems.

1. The class distribution is unbalanced and there are in particular very few instances for the ”disagree” class. The problem is substantial as oversampling and under-sampling experiments did not help to increase performance. However, the experiments on the ARC dataset suggest that the differentiation of the ”agree” and ”disagree” instances can be learned with reasonable performance if the dataset is balanced.

2. The human upper bound is relatively low, reaching only 0.754 F1 macro. The differentiation between ”agree”, ”disagree” and ”discuss” classes is very challenging even for humans, as we reach only 0.218 Fleiss’ κ inter-annotator agreement on these three classes.

3.The error analysis from Section 6.4 shows that the models exploit the similarity between the headline and the article body in terms of lexical overlap. Furthermore, lexical cue words, such as ”reports”, ”said”, ”false”, ”hoax” are important for classification. The systems fail when semantic relations between words need to be taken into account, complex negation instances are encountered, or the understanding of propositional content in general is required.

8 Conclusion

In our experiments, we have tested numerous models and features for the FNC document-level stance detection task. Moreover, we crucially assessed the successful models, the problem setting, the dataset itself, and evaluated the performance of the models on a second dataset. Based on these insights, we have developed a new model which reaches a new state-of-the-art. Nevertheless, we have also found that even the best performing models reach relatively low F1 macro scores of about 0.6. We further analyzed why the systems reach low performances and have identified three main causes. 1. The dataset is unbalanced and there are only few instances for certain classes. 2. The task is challenging and the human upper bound is relatively low with 0.754 F1 macro. 3. The best performing models use mostly similarity based features and are therefore unable to resolve difficult instances.

Based on these findings, we conclude that in order to improve the performance of machine learning methods on the document-level stance detection task, a better balanced corpus with a higher inter annotator agreement is required. Moreover, similarity based approaches appear to reach their limit on the task. Thus, more sophisticated machine learning techniques are needed, which are better able to deal with complex negation instances, have a deeper semantic understanding, and are able to determine the stance on the basis of propositional content.

References

Jonathan Anderson. 1983. Lix and rix: Variations on a little-known readability index. *Journal of Reading* 26(6):490–496.

Sean Baird, Doug Sibley, and Pan Yuxi. 2017. Talos targets disinformation with fake news challenge victory. http://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html. Accessed: 2017-12-2.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022. http://dl.acm.org/citation.cfm?id=944919.944937.

Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. 2017. From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*. pages 84–89.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60(2):283.

Sanjiv R Das and Mike Y Chen. 2007. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science* 53(9):1375–1388.

Richard Davis and Chris Proctor. 2017. Fake news, real consequences: Recruiting neural networks for the fight against fake news .

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41(6):391.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972* .

William Ferreira and Andreas Vlachos. 2016. 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.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. pages 1026–1034.

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

Benjamin D. Horne and Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *CoRR* abs/1703.09398. http://arxiv.org/abs/1703.09398.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning Whom to Trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*. Atlanta, GA, USA, pages 1120–1130. http://www.aclweb.org/anthology/N13-1132.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50:723–762.

Chih-Jen Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural Computation* 19(10):2756–2779. https://doi.org/10.1162/neco.2007.19.10.2756.

Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 393–398.

Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Günter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. 2014. The excitement open platform for textual inferences. In *ACL (System Demonstrations)*. pages 43–48.

Rachel McAlpine. 1997. *Global english for global business*. Longman.

Philip M McCarthy. 2005. An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (mtld). *Dissertation Abstracts International* 66:12.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*. Atlanta, Georgia, USA.

Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, pages 26–34.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon 29(3):436–465.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Dean Pomerleau. 2017. Time to challenge fake news with ai. `https://medium.com/@deanpomerleau_24908/time-to-challenge-fake-news-with/-ai-7036a1f22c0d`. Accessed: 2017-12-2.

Dean Pomerleau and Delip Rao. 2017. The fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. `http://www.fakenewschallenge.org/`. Accessed: 2017-10-20.

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

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 451–463. http://www.aclweb.org/anthology/S15-2078.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, CINCINNATI UNIV OH.

Craig Silverman. 2017. Emergent: A real-time rumor tracker. `http://www.emergent.info/`. Accessed: 2017-12-13.

N. Watson Solomon. 2006. Strain index: A new readability formula.

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

Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective stance classification of posts in online debate forums.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics* .

Sanja Štajner, Richard Evans, Constantin Orasan, and Ruslan Mitkov. 2012. What can readability measures really tell us about text complexity. In *Proceedings of the the Workshop on Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*. pages 14–21.

Stanford. 2017. Cs224n: Natural language processing with deep learning, course project reports for 2017. `http://web.stanford.edu/class/cs224n/reports.html`. Accessed: 2017-12-13.

James Thorne, Mingjie Chen, Giorgos Myrianthous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. 2017. Fake news stance detection using stacked ensemble of classifiers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*. pages 80–83.

Marilyn A Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 592–596.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '05, pages 347–354. https://doi.org/10.3115/1220575.1220619.

Guido Zarrella and Amy Marsh. 2016. Mitre at semeval-2016 task 6: Transfer learning for stance detection. *arXiv preprint arXiv:1606.03784* .

Xiaodan Zhu, Svetlana Kiritchenko, and Saif M Mohammad. 2014. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. Citeseer, pages 443–447.

A Supplemental Material

A.1 Features: Detailed description

BoW/BoC features We use bag-of-words (BoW) 1- and 2-grams with 5,000 tokens vocabulary for the headline as well as the body. For the BoW feature, based on a technqiue by Das and Chen (2007), we add a negation tag "_NEG" as prefix to every word between special negation keywords (e.g. "not", "never", "no") until the next punctuation mark appears. For the bag-of-characters (BoC) 3-grams are chosen with 5,000 tokens vocabulary, too. For the BoW/BoC feature we use the TF to extract the vocabulary and to build the feature vectors of headline and body. The resulting TF vectors of headline and body get concatenated afterwards. Feature *co-occurrence* (FNC-1 baseline feature) counts how many times word 1-/2-/4-grams, character 2-/4-/8-/16-grams, and stop words of the headline appear in the first 100, first 255 characters of the body, and how often they appear in the body overall.

Topic models We use non-negative matrix factorization (NMF) (Lin, 2007), latent semantic indexing (LSI) (Deerwester et al., 1990), and latent Dirichlet allocation (LDA) (Blei et al., 2003) to create topic models out of which we create independent features. For each topic model, we extract 300 topics out of the headline and body texts. Afterwards, we compute the similarity of headlines and bodies to the found topics separately and either concatenate the feature vectors (NMF, LSI) or calculate the cosine distance between them as a single valued feature (NMF, LDA).

Lexicon-based features These features are based on the NRC Hashtag Sentiment and Sentiment140 lexicon (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014), as well as for the MPQA lexicon (Wilson et al., 2005) and MaxDiff Twitter lexicon (Rosenthal et al., 2015; Kiritchenko et al., 2014). All named lexicons hold values that signal the sentiment/polarity for each word. The features are computed separately for headline and body, and constructed as proposed by Mohammad et al. (2013): First, we count how many words with positive, negative, and without polarity are found in the text. Two features sum up the positive and negative polarity values of the words in the texts and another two features are set by finding the word with the maximum positive and negative polarity value in the text. Finally, the last word in the text with negative or positive polarity is taken as a feature. Since the MaxDiff Twitter lexicon also contains 2-grams, we decide to take them into account as well, whereas for the other lexicons only 1-grams incorporated. Additionally, we base features on the EmoLex lexicon (Mohammad and Turney, 2010, 2013). For all its words, it holds up to eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, disgust), based on the context they frequently appear in. For headline and body respectively, the emotions for all words are counted as a feature vector. The resulting vectors for headline and body are then concatenated. Lastly, the baseline features *polarity words* and *refuting words* are added. The first one counts refuting words (e.g. "fake", "hoax"), divides the sum by two, and takes the remainder as a feature signaling the polarity of headline or body. The latter one sets a binary feature for each refuting word (e.g. "fraud", "deny") appearing in the headline or body.

Readability features We measure the readability of headline and body with SMOG grade (only body), Flesch-Kincaid grade level, Flesch reading ease, and Gunning fog index (Štajner et al., 2012), Coleman-Liau index (Coleman and Liau, 1975), automated readability index (Senter and Smith, 1967), LIX and RIX (Anderson, 1983), McAlpine EFLAW Readability Score (McAlpine, 1997), Strain Index (Solomon, 2006). The SMOG grade is only valid if a text has at least 30 sentences, and thus is only implemented for the bodies.

Lexical features As lexical features we implement the type-token-ratio (TTR) and the measure of textual lexical diversity (MTLD) (McCarthy, 2005) for the body, and only type-token-ratio for the headline, since MTLD needs at least 50 tokens to be valid. Also, the baseline feature *word overlap* belongs to this group. It divides the cardinality of the intersection of unique words in headline and body by the cardinality of the union of unique words in headline and body.

POS features The POS features amongst others include counters for nouns, personal pronouns, verbs and verbs in past tense, adverbs, nouns and proper nouns, cardinal numbers, punctuations, the ratio of quoted words, and also the frequency of the three least common words in the text. The headline feature also contains a value for the percentage of stop words and the number of verb phrases, which showed good results in the work of Horne and Adali (2017). For the *word-similarity* feature, [which are mainly based on Ferreira and Vlachos (2016) we calculated average word embeddings (pre-trained word2vec model[6]) for all verbs (retrieved with Stanford Core NLP toolkit[7]) of headline/body separately. The cosine similarity between the averaged embeddings of headline and body is taken as a feature, as well as the hungarian distance between verbs of headline and body based on the parapharse database[8]. The same computation is done for all nouns of headline and body. Additionally the average sentiment of the headline and the average sentiment of the body is used as a feature. A count of negating words of the headline and the body is added to the feature vector as well as the distance from the negated word to the root of the sentence. The number of average words per sentence of headline and body is another feature. The aforementioned features are improved by only selecting a predefined number of sentences of body and headline. Therefore the sentences are ordered by TF-IDF score.

Structural features The structural features contain the average word length of the headline and body, and the number of paragraphs and average paragraph length of the body.

[6]https://code.google.com/archive/p/word2vec/
[7]https://stanfordnlp.github.io/CoreNLP/
[8]http://www.cis.upenn.edu/ ccb/ppdb/

A.2 Features tested separately

Features	FNC	F1 macro
Baselines		
Majority vote	.3877	.2877
FNC-1 features	.7929	.4990
Topic models		
LSI 300 †	.8834	.7502
NMF 300 †	.8563	.7016
NMF 300 cos-sim. †	.8210	.4361
LDA 300 cos-sim. †	.7419	.4081
BoW/BoC features		
BoW 1-/2-grams 5,000 †	.9015	.7782
BoC 3-grams 5,000 †	.9034	.7729
Co-occurrence †	.7729	.4701
POS features		
Wordsim †	.7708	.4233
NRC Hashtag POS	.5342	.3427
Lexicon-based features		
EmoLex 1-grams	.4816	.3490
Sentiment140 1-grams	.4471	.2913
NRC Hashtag 1-grams	.4319	.2718
MPQA 1-grams	.3932	.2226
Polarity features	.3877	.2088
MaxDiff 1-/2-grams	.3877	.2088
Refuting features	.3877	.2088
Readability features		
Readability_features	.4430	.2842
Structural features		
Structural_features	.3959	.2197
Lexical features		
Lexical_features	.6918	.3854

Table 6: Features tested with the tuned multi-layer perceptron. Some of the features of the different groups are listed separately in order to show their high variances in score. Before the feature ablation test is done, some of the low-scoring features shown separately are removed. Only features marked with † are considered.

A.3 Misclassified examples identified the error analysis

Example 1.
(stance ”unrelated”, system predicts ”agree”)
Headline: CNN: Doctor Took Mid-Surgery Selfie

with Unconscious Joan Rivers

Body: ”A TEENAGER woke up during brain surgery to ask doctors how it was going. Iga Jasica, 19, was having an op to remove a tumour at when the anaesthetic wore off and she struck up a conversation with the medics still working on her.”

Example 2.
(stance ”agree”, system predicts ”unrelated”)
Headline: Three Boobs Are Most Likely Two Boobs and a Lie

Body: The woman who claimed she had a third breast has been proved a hoax.

Example 3.
(stance ”disagree”, system predicts ”discuss”)
Headline: Woman pays 20,000 for third breast to make herself LESS attractive to men

Body: The woman who reported that she added a third breast was most likely lying.

Example 4.
(stance ”disagree”, system predicts ”agree”)
Headline: Disgusting! Joan Rivers Doc Gwen Korovins Sick Selfie EXPOSED Last Photo Of Comic Icon, When She Was Under Anesthesia

Body: If the bizarre story about Joan Rivers’ doctor pausing to take a ”selfie” in the operating room minutes before the 81-year-old comedienne went into cardiac arrest on August 29 sounded outlandish, that’s because it was.

A.4 FNC score for the models experiments

Models	FNC	F1 macro
Baselines:		
major. vote	.394	.210
maj. v. dsc	**.833**	.444
TalosTM	.830	.570
TalosCNN	.502	308
TalosComb	.820	.582
uclMLP	.817	.583
Class.:		
NaiveB	.471	.283
GradBoost	.811	.531
LogReg	.815	.551
SVM	.819	.561
Proposed:		
avrgCNN	.472	.355
weightMLP	.745	.525
featMLP	.827	.607
LSTM	.821	**.609**
upp. bound	.859	.754

Table 7: FNC-scores and F1 macro scores for the analyzed models