Tribrid: Stance Classification with Neural Inconsistency Detection

Song Yang
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands
s.yang@student.vu.nl

Jacopo Urbani
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands
jacopo@cs.vu.nl

Abstract

We study the problem of performing automatic stance classification on social media with neural architectures such as BERT. Although these architectures deliver impressive results, their level is not yet comparable to the one of humans and they might produce errors that have a significant impact on the downstream task (e.g., fact-checking). To improve the performance, we present a new neural architecture where the input also includes automatically generated negated perspectives over a given claim. The model is jointly learned to make simultaneously multiple predictions, which can be used either to improve the classification of the original perspective or to filter out doubtful predictions. In the first case, we propose a weakly supervised method for combining the predictions into a final one. In the second case, we show that using the confidence scores to remove doubtful predictions allows our method to achieve human-like performance over the retained information, which is still a sizable part of the original input.

1 Introduction

The spreading of unverified claims on social media is an important problem that affects our society at multiple levels (Vlachos and Riedel, 2014; Ciampaglia et al., 2015; Hassan et al., 2017). A valuable asset that we can exploit to fight this problem is the set of perspectives that people publish about such claims. These perspectives reveal the users' stance and this information can be used to help an automated framework to determine more accurately the veracity of rumors (Castillo et al., 2011; Bourgonje et al., 2017).

The stance can be generally categorized either as supportive or opposing. For instance, consider the claim "*The elections workers in Wisconsin illegally altered absentee ballot envelopes*". A tweet with a supportive stance is "*The number of people who took part in the election in Wisconsin exceeded the total number of registered voters*" while one with an opposed stance is "*An extra zero was added as votes accidentally but it was quickly fixed after state officials noticed it*".

Being able to automatically classify the stance is necessary for dealing with the large volume of data that flows through social networks. To this end, earlier solutions relied on linguistic features, such as n-grams, opinion lexicons, and sentiment (Somasundaran and Wiebe, 2009; Anand et al., 2011; Hasan and Ng, 2013; Sridhar et al., 2015) while more recent methods additionally include features that we can extract from networks like Twitter (Chen and Ku, 2016; Lukasik et al., 2016; Sobhani et al., 2017; Kochkina et al., 2017). The current state-of-the-art relies on advanced neural architectures such as BERT (Devlin et al., 2019) and returns remarkable performance. For instance, STANCY (Popat et al., 2019) can achieve an F_1 of 77.76 with the PERSPECTRUM (PER) dataset against the F_1 of 90.90 achieved by humans (Chen et al., 2019).

Although these results are encouraging, the performance has not yet reached a level that it can be safely applied in contexts where errors must be avoided at all costs. Consider, for instance, the cases when errors lead to a misclassification of news about a catastrophic event, or when they trigger wrong financial operations. In such contexts, we argue that it is better that the AI abstains from returning a prediction unless it is very confident about it. This requirement clashes with the design of current solutions, which are meant to "blindly" make a prediction for any input. Therefore, we see a gap between the capabilities of the state-of-the-art and the needs of some realistic use cases.

In this paper, we address this problem with a new BERT-based neural network, which we call *Tribrid* (TRIplet Bert-based Inconsistency Detection), that is designed not only to produce a reliable and accurate classification, but also to test its confidence. This test is implemented by including a "negated"

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6831–6843
November 7–11, 2021.

version of the original perspective as part of the input, following the intuition that a prediction is more trustworthy if the model produces the opposite outcome with the negated perspective. If that is not the case, then the model is inconsistent and the prediction should be discarded.

Testing the consistency of the model with negated perspectives is a task that can be done simply by computing two independent predictions, one with the original perspective and one with the negated one. However, this is suboptimal because existing state-of-the-art methods are trained only with the principle that supportive perspectives should be similar to their respecting claims in the latent space (Popat et al., 2019) and the similarity might be sufficiently high even if there are keywords (e.g., "not") that negate the stance. To overcome this problem, we propose a new neural architecture that processes *simultaneously* both original and negated perspectives, using a siamese BERT model and a loss function that maximises the distance between the two perspectives. In this way, the model learns to distinguish more clearly supportive and opposite perspectives.

To cope with the large volume of information that flows through social networks, it is important that the negated perspectives are generated automatically or at least with a minimal human intervention. To this end, two types of techniques have been presented in the literature. One consists of attaching a fixed phrase which negates the meaning (Bilu et al., 2015) while the other adds or removes the first occurrences of tokens like "not" (Niu and Bansal, 2018; Camburu et al., 2020). *Tribrid* implements this task in a different way, namely using simple templates that negate both with keywords (e.g., "not") and antonyms.

The prediction scores obtained with *Tribrid* can be used either to determine more accurately the stance of the original perspective or to discard low-quality predictions. We consider both cases: In the first one, we propose an approach where multiple classifiers are constructed from the scores and a final weakly supervision model combines them. In the second one, we propose several approaches to establish the confidence and describe how to use them to discard low-quality predictions. Our experiments show that our method is competitive in both cases. For instance, in the second case, our approach was able to achieve a F_1 of 87.43 on PER by excluding only 39.34% of the perspectives. Moreover, the score increased to 91.26 when 30% more is excluded. Such performance is very close to the one of humans and this opens the door to an application in contexts where errors are very costly.

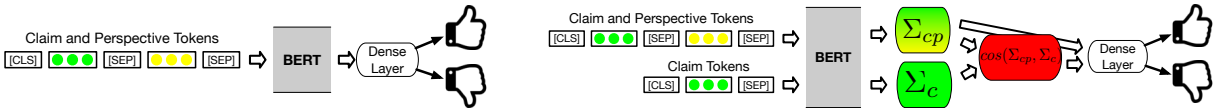
The source code and other experimental data can be found at https://github.com/karmaresearch/tribrid.

2 Background and Related Work

Stance Classification aims to determine the stance of a input perspective that supports or opposes another given claim. In earlier studies, the research mainly focused on online debate posts using traditional classification approaches (Thomas et al., 2006; Murakami and Raymond, 2010; Walker et al., 2012). Afterwards, other approaches focused on spontaneous speech (Levow et al., 2014), and on student essays (Faulkner, 2014). Thanks to the rapid development of social media, the number of studies on tweets has increased substantially (Rajadesingan and Liu, 2014; Chen and Ku, 2016; Lukasik et al., 2016; Sobhani et al., 2017; Kochkina et al., 2017), especially boosted by dedicated SemEval challenges (Mohammad et al., 2016; Kochkina et al., 2017) and benchmarks (Bar-Haim et al., 2017; Chen et al., 2019).

Ealier methods for stance classification employed various traditional classifiers, which include rule-based algorithms (Anand et al., 2011); supervised classifiers like SVM (Hasan and Ng, 2013), naïve Bayes (Rajadesingan and Liu, 2014), boosting (Levow et al., 2014), decision tree and random forest (Misra and Walker, 2013), Hidden Markov Models (HMM) and Conditional Random Fields (Hasan and Ng, 2013); graph algorithms such as MaxCut (Murakami and Raymond, 2010), and other approaches such as Integer Linear Programming (Somasundaran and Wiebe, 2009) and Probabilistic Soft Logic (Sridhar et al., 2014). Popular features include cue/topic words, argument-related and sentiment/subjectivity features, and frame-semantic features. Other features like tweets reply, rebuttal information, and retweets are known to improve the performance (Sobhani et al., 2019).

In more recent work, the NLP community investigated on how to use deep neural network to improve the performance. Some representatives are LSTM-based approaches (Du et al., 2017; Sun et al., 2018; Wei et al., 2018), RNN-based approaches (Sobhani et al., 2019; Borges et al., 2019), CNN approaches (Wei et al., 2016; Zhang et al.,



(a) BERT$_{base}$ (b) STANCY

Figure 1: BERT used for stance classification by Chen et al. (2019) and Popat et al. (2019)

2017) and more recently BERT-based approaches (Chen et al., 2019; Popat et al., 2019; Schiller et al., 2020). Techniques like attention mechanisms (Du et al., 2017), memory networks (Mohtarami et al., 2018), lexical features (Riedel et al., 2017; Hanselowski et al., 2018), transfer learning and multi-task learning (Schiller et al., 2020) can also improve the performance.

All these approaches focus on identifying the most effective method to achieve the highest possible performance using syntactic and semantic features from the input. In contrast, our goal is to improve the performance by injecting background knowledge in the form of negated text, encouraging the model to produce more consistent predictions. As far as we know, we are the first that study this form of optimization to improve the performance of stance classification.

The generation of negated perspectives can be viewed as an instance of the broader problem of constructing adversarial examples, which is drawing more attention in NLP research in recent years (Zhang et al., 2020; Wang et al., 2019). The main focus of adversarial generation is to change the text (with character/words changes or removals) to train more robust models. Most of the works focus on changes that do not alter the semantics of the original input (Belinkov and Bisk, 2018; Xiao et al., 2017; Iyyer et al., 2018; Cheng et al., 2020) while works that negate the semantics are many fewer. In this context, some initial works have manually constructed some small-scale tests (Isabelle et al., 2017; Mahler et al., 2017). More recently, Gardner et al. (2020) suggested that datasets should be perturbed by experts with small changes to the test instances. In this way, empirical evaluations can test more accurately the true linguistic capabilities of the models. In their evaluation, they selected PER, one of the datasets that we also consider, and showed that models such as BERT perform significantly worse on the perturbed dataset. Also Ribeiro et al. (2020) consider the problem that accuracy on a held-out dataset may overestimate the performance on a real scenario. To counter this, they propose a new methodology that involves tests with certain perturbations (like negation), but they do not consider stance classification. These works further motivate our effort to develop models that are more consistent when presented with negated inputs. Moreover, another goal of our work is to use consistency as a proxy to measure uncertainty and to discard low-quality predictions. A similar objective was pursued by Kochkina and Liakata (2020) for the problem of rumor verification.

Since manually creating additional test instances is time consuming, some works propose automatic procedures to generate them. Bilu et al. (2015) proposes to add a fixed phrase at the end ("but this is not true") while Niu and Bansal (2018) adds the token "not" before the first verb in the sentence or replaces it with its antonym. Finally, Camburu et al. (2020) suggests a simpler alternative to remove the first occurrence of "not". In contrast to them, we use templates in the form of *if-then* rules.

3 Our Approach

First, we provide a short description on how BERT has been used to achieve the state-of-the-art for this problem (Section 3.1). Then, we describe our proposed neural network (Section 3.2) and how we can interpret its output to classify the stance (Section 3.3). Finally, we discuss the generation of negated perspectives using templates (Section 3.4).

3.1 BERT Base and STANCY

Our input is a sentence pair $\langle C, P \rangle$ where C is the input claim and P is the perspective. In 2019, Chen et al. proposed to concatenate C and P using two tokens `[CLS]` and `[SEP]` to delimit the claim and the perspective, respectively, and to feed the resulting string to BERT (see Figure 1a). We call this approach BERT$_{base}$. A few months later, Popat et al. (2019) proposed an improvement based on the assumption that the latent representation of a perspective should be similar if the perspective support the claim and vice versa. The resulting network is called STANCY (see Figure 1b). The main idea is to compute a latent representation of the claim

and perspective (Σ_{cp}) and one for the claim alone (Σ_c). The two are compared with cosine similarity, denoted with $cos(\cdot)$, and passed to a final dense layer that performs the classification.

3.2 Tribrid: Neural Architecture

Current solutions deploy *one* BERT model passing, as input, the claim followed by the perspective to obtain the latent representation (see, e.g., Figure 1a). This approach is not ideal for us because we would like to pass more information as input (the negated perspective), and this might lead to a string that is too long. We address this problem by using *multiple* BERT models that share the same parameters, thus creating a network that is often labeled as a *siamese* network (Bromley et al., 1993).

A schematic view of *Tribrid* is shown in Figure 2. As input, *Tribrid* receives the triplet $\langle C, P, NP\rangle$ where NP is the negated perspective. We also provide a simpler architecture where the input is the pair $\langle C, P\rangle$, which we call *Tribrid*$_{pos}$.

In *Tribrid*, each component of the input triplet is fed to a BERT model that shares the parameters with the other two models. The three BERT models compute latent representations for C, P, and NP, henceforth written as Σ_c, Σ_p, and $\Sigma_{\neg p}$. In the second stage, the network concatenates them into a single representation. We experimented with the concatenation techniques proposed in Sentence-BERT (Reimers and Gurevych, 2019), InferSent (Conneau and Kiela, 2018) and Universal Sentence Encoder (Cer et al., 2018), namely $(|\Sigma_c - X|, \Sigma_c * X)$, $(\Sigma_c, X, \Sigma_c * X)$, $(\Sigma_c, X, |\Sigma_c - X|)$ and $(\Sigma_c, X, |\Sigma_c - X|, \Sigma_c * X)$ where $X \in \{\Sigma_p, \Sigma_{\neg p}\}$, $|\dots|$ is the element-wise distance, and $*$ is the element-wise multiplication. We selected $(\Sigma_c, X, |\Sigma_c - X|, \Sigma_c * X)$ as it slightly outperformed the others in our experiments and further concatenated $cos(\Sigma_c, X)$ to it because their similarity is valuable for predicting the stance (Popat et al., 2019).

Finally, the result of the concatenation is passed to a final dense layer that returns two logits λ_s and λ_o, which estimate the likelihood of supporting and opposing stances, respectively.

To train the model, we introduce the following loss function $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_e + \mathcal{L}_d$, described below (with *Tribrid*$_{pos}$, $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_e$). The first component $\mathcal{L}_c$ is a standard cross entropy loss:

$$\mathcal{L}_c = -log\left(\frac{exp(\hat{y})}{exp(\lambda_s) + exp(\lambda_o)}\right) \quad (1)$$

where $\hat{y} \in \{\lambda_s, \lambda_o\}$ is the logit of the true stance.

The second part is the cosine embedding loss:

$$\mathcal{L}_e = \begin{cases} 1 - cos(\Sigma_c, \Sigma_p) & \text{if } y = 1 \\ max(0, cos(\Sigma_c, \Sigma_p)) & \text{if } y = -1 \end{cases} \quad (2)$$

where $y = 1$ if the perspective supports the claim and -1 if it is opposed to it.

The third component $\mathcal{L}_d$ is added because $\mathcal{L}_c$ and $\mathcal{L}_e$ do not take into account the fact that the perspective that supports the claim should be "closer" to the claim than the perspective that opposes it. To this end, we add a triplet loss (Schroff et al., 2015). Let $\Sigma^+ = \Sigma_p$ and $\Sigma^- = \Sigma_{\neg p}$ if the input perspective supports the claim or $\Sigma^+ = \Sigma_{\neg p}$ and $\Sigma^- = \Sigma_p$ otherwise. Then,

$$\mathcal{L}_d = max(\gamma + \delta^+ - \delta^-, 0) \quad (3)$$

with γ being the margin, $\delta^+ = |\Sigma_c - \Sigma^+|$, and $\delta^- = |\Sigma_c - \Sigma^-|$.

3.3 Stance Classification

We can make use of four signals to predict the stance. The first two are the logits λ_s and λ_o. The third one is $\delta_p = ||\Sigma_c - \Sigma_p||$, i.e., the distance between the claim and the input perspective. The fourth one is $\delta_{\neg p} = ||\Sigma_c - \Sigma_{\neg p}||$, i.e., the distance to the negated perspective. The values of these signals can be combined in different ways in order to compute a final binary decision. We define two possible alternative procedures.

The first procedure consists of picking the logit with the highest value as the final label, e.g., if $\lambda_s > \lambda_o$, then the stance should be support. In contrast, the second procedure looks at the distance values. In this case, it chooses the final label depending on which perspective has the closest distance, i.e., the system should return "support" if $\delta_p < \delta_{\neg p}$ or "oppose" otherwise.

In both cases, the confidence of the model can be quantified by the difference between the two signals. If difference is at least τ, where τ is a given threshold, then we can accept the outcome trusting that the system is sufficiently confident. Otherwise, we abstain from making a prediction. Following this principle, we introduce the decision procedures K^τ and Λ^τ, defined as follows:

$$K^\tau = \begin{cases} \mathsf{S} & |\lambda_s - \lambda_o| \geq \tau \text{ and } \lambda_s \geq \lambda_o \\ \mathsf{O} & |\lambda_s - \lambda_o| \geq \tau \text{ and } \lambda_s < \lambda_o \\ \mathsf{A} & \text{otherwise} \end{cases} \quad (4)$$

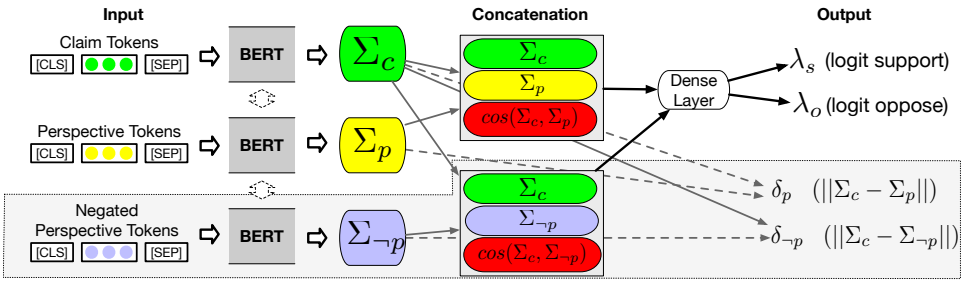


Figure 2: *Tribrid* network: The components in the light gray area are the ones that process the negated perspectives

and

$$\Lambda^{\tau} = \begin{cases} \mathsf{S} & |\delta_p - \delta_{\neg p}| \geq \tau \text{ and } \delta_p < \delta_{\neg p} \\ \mathsf{O} & |\delta_p - \delta_{\neg p}| \geq \tau \text{ and } \delta_p \geq \delta_{\neg p} \\ \mathsf{A} & \text{otherwise} \end{cases} \quad (5)$$

where S stands for support, O for oppose, and A for abstain.

The advantage of these procedures is that we can select the minimum amount of acceptable confidence by choosing an appropriate τ. In practice, we can use a small validation dataset or pick τ so that at most $X\%$ of the data is excluded.

In case the user is not willing to discard any prediction, then we propose a third decision procedure where the system never abstains. In essence, our proposal consists of creating multiple K^{τ} and Λ^{τ} classifiers with different τ which are fed to an ensemble method that makes the final prediction.

A simple example of an ensemble method is majority voting, but this technique does not consider latent correlations between the classifiers. To take those into account, we can rely on weak supervision. In particular, we can use the state-of-the-art method proposed by Fu et al. (2020), which is called *FlyingSquid*. As far as we know, methods like the one of Fu et al. have not yet been considered for stance classification. We show here that they lead to an improvement of accuracy.

The main goal of *FlyingSquid* is to learn a model that is able to compute a probabilistic label (which is the stance in our case) with a set of noisy labeling functions (in our case the K^{τ} and Λ^{τ} classifiers) given as input. This method is particularly interesting to us for two reasons: 1) It does not need to access ground truth annotations (which are scarce in our context), and 2) it can find the optimal model's parameters quickly, without iterative procedures like gradient descent.

We proceed as follows. First, we create n classifiers K^i and Λ^j where $i \in \{\tau_1^K, \ldots, \tau_n^K\}$ and $j \in \{\tau_1^{\Lambda}, \ldots, \tau_n^{\Lambda}\}$. For a given pair $\langle C, P, NP \rangle$, these classifiers produce $2n$ labels that can be either $\{\mathsf{S}, \mathsf{O}, \mathsf{A}\}$. These labels form the input for *FlyingSquid*, which learns a model from the labels' correlations and return a final label $l \in \{\mathsf{S}, \mathsf{O}\}$ for every $\langle C, P, NP \rangle$.

To recap, we proposed three approaches for stance classification with our neural model. The first is the K^{τ} classifier, the second is Λ^{τ}, while the third is a weak supervision model (*FlyingSquid*) built from multiple K^{τ} and Λ^{τ} classifiers. The first two classifiers might abstain if the model is not confident while the third one always returns a binary output. Henceforth, we refer to them as *Tribrid*$_l$, *Tribrid*$_d$, and *Tribrid*$_w$, respectively.

3.4 Templates for Automatic Negation

To create negated perspectives, we use templates that are encoded as if-then rules of the form $A \Rightarrow B$. The rules contain instructions on how to change the text. In case more rules apply to the same perspective, then only the first application is kept.

For our purpose, the templates should be relatively simple so that they can applied to large volumes of text and do not capture biases in some datasets. Moreover, the templates should make only few changes to the text because we would like to teach the model to pay attention to specific tokens, or combinations of words, that can potentially change the stance.

With this desiderata in mind, we randomly picked some perspectives from multiple datasets and negated them by encoding meaningful changes into rules. This process returned a list of about 60 templates. From this list, we extracted 14 templates which are enough to cover about 90% of the cases (see Section 4.1 for more details about the coverage and the appendix for the list of all templates). A few examples of templates are shown below:

P_1 :	[X] is/was/... [Y]	$\Rightarrow$	[X] is/was/... not [Y]
P_2 :	[X] more [Y]	$\Rightarrow$	[X] less [Y]
P_3 :	[X] help [Y]	$\Rightarrow$	[X] spoil [Y]

Split	Train	Dev	Test	Total
S pairs	3603	1051	1471	6125
O pairs	3404	1045	1032	5751
Total	7007	2096	2773	11876

(a) Perspectrum (PER)

Train	Test	Total
625	700	1325
414	655	1069
1039	1355	2394

(b) IBMCS

Table 1: Dataset Statistics; S=Support, O=Oppose

As we can see from the list, these patterns are fairly simple and mostly reduce to a strategical position of "not" or to replace words with antonyms.

4 Evaluation

We tested our approach on the datasets PER (Chen et al., 2019) and IBMCS (Bar-Haim et al., 2017), which are the main datasets previously used by our competitors. PER is a set of claims and perspectives constructed from online debate websites while IBMCS is a similar dataset released by IBM. Statistics on both datasets are in Table 1a and 1b. We did not use the datasets proposed in SemEval-2016, Task 6 (Mohammad et al., 2016) and SemEval2017, Task 8 (Derczynski et al., 2017) for the predictions because our work focuses on a binary classification while these datasets also include additional classes such as "neutral", "query", or "comment". Extending our method to predict more than two classes should be seen as future work.

We implemented our approach using PyTorch 1.4.0, and a BERT BASE model with 12 layers, 768 hidden size, and 12 attention heads. We fine-tuned BERT with grid search optimizing the F_1 on a validation dataset with a learning rates {1, 2, 3, 4, 5} $\times 10^{-5}$, batch size {24, 28, 32}, and the Adam optimizer. For our experiments, we used a machine with a TitanX GPU with 12GB RAM.

In the following, we first discuss the results using the confidence based K^τ and Λ^τ classifiers (*Tribrid*$_l$ and *Tribrid*$_d$). Then, we study the performance with our weakly supervised approach. Finally, we provide an analysis on the coverage of the templates for negating the perspectives.

Data Collection	Cover Cases	Total Cases	Cover Rate
PER train claim	6514	7007	0.930
PER train perspective	6270	7007	0.895
PER test claim	2618	2773	0.944
PER test perspective	2527	2773	0.911
PER dev claim	1866	2096	0.890
PER dev perspective	1881	2096	0.897
IBMCS train claim	1039	1039	1.000
IBMCS train perspective	927	1039	0.892
IBMCS test claim	1274	1355	0.940
IBMCS test perspective	1192	1355	0.880
ARC train set	13878	14233	0.975
ARC test set	3478	3559	0.977
SemEval2016 train set	2280	2814	0.810
SemEval2016 test set	1013	1249	0.811

Table 2: Number of perspectives (cases) that can be negated at least by one of our templates vs. the total number of perspectives

Dataset	F1 (Precision, Recall)	Coverage
PER	83.54 (82.53, 84.56)	86.62%
IBMCS	73.06 (73.80, 72.33)	86.83%

Table 3: Percentage of cases when the logits and distance values agree, and F_1, precision and recall on this subset of cases

4.1 Confidence-based Stance Classification

To establish how general our templates are, we have applied them to the text in PER and IBMCS and in the dataset of SemEval2016, Task 6. Moreover, we also considered datasets which are used for other NLP tasks, namely ARC (Habernal et al., 2018; Hanselowski et al., 2018). As we can see in Table 2, the templates generalize well as they can negate more than 90% of the perspectives both in PER and IBMCS and between 81.0% and 100% in the other cases. For now, we restrict our analysis to the subset of perspectives for which there is a negation. In the next section, we will also consider the (few) remaining cases.

First, it is interesting to look at the percentage of cases when the four signals, i.e., λ_s, λ_o, δ_p, and $\delta_{\neg p}$, agree. If this occurs for a certain input, then we can interpret it as an hint that the model is confident about the prediction. For instance, if $\lambda_s > \lambda_o$ and $\delta_p < \delta_{\neg p}$, then we are confident that the output should be support. Table 3 reports the percentage of the input where the signals agree (column "Coverage") and the F_1 that we would obtain if we follow this strategy for deciding the stance. We see that the number of cases when there is an agreement is large (86% on PER) and the performance is fairly high (F_1 of about 83.5) and superior to

Filtered Percentage	10%	20%	30%	40%	50%	60%	70%	80%	90%
BERT$_{base}$	71.94	73.98	76.44	78.40	80.61	81.51	77.89	67.10	7.14
STANCY	79.41	81.46	82.91	82.75	79.92	69.89	1.48	0.00	0.00
Tribrid$_l$	82.87	85.49	86.88	87.60	87.51	85.54	85.31	88.56	86.71
Tribrid$_d$	80.65	83.01	84.69	86.42	87.58	89.54	91.26	93.18	96.02

Table 4: F_1 with *Tribrid*$_l$ and *Tribrid*$_d$ varying the threshold values on PER

the one that we would obtain with, e.g. STANCY over the entire dataset (77.76, Table 5). From this, we conclude that this is a rather simple strategy to improve the performance without discarding many cases. However, notice that with this approach we cannot choose which is the minimum level of acceptable confidence. For this, we can use our proposed *Tribrid*$_l$ or *Tribrid*$_d$.

In general, if we want to accept only the cases with high confidence, then we can use either *Tribrid*$_l$ or *Tribrid*$_d$ with a high τ. In this way, however, it is likely that we will discard many cases. To study what the tradeoff would be in our benchmarks, we present the results of an experiment where we pick τ such that only $X\%$ of the data will be missed. Two natural baselines for comparing our performance consist of applying the K^τ decision procedure using the logits returned by BERT$_{base}$ and STANCY. In this way, we can also study the behaviour using other methods. Notice that in this experiment τ is chosen independently for each method. This means that the value of τ with STANCY can be different than the value of τ with BERT$_{base}$ when, for instance, 10% of the predictions are filtered out. In this way, the comparison is fair since it is done on subsets of predictions which have similar sizes.

Table 4 reports the results of this experiments while Figure 3 plots the same numbers in a graph. As expected, we observe the that F_1 increases as we increase τ. However, notice that with BERT$_{base}$ and STANCY the F_1 drastically decreases after we filter approximately more than 70-80% of the cases. That point marks the maximum F_1 that we can achieve with those methods. Instead, with our method the F_1 keeps increasing to higher values, which indicates that our system is capable of producing much higher-quality predictions.

We make *three* main observations. First, with a similar discard rate, our approach outperforms both BERT$_{base}$ and STANCY (the difference is statistically significant with a p-value of 0.0379 as per paired t-test between *Tribrid*$_l$ and STANCY and 0.0441 between *Tribrid*$_d$ and STANCY). This

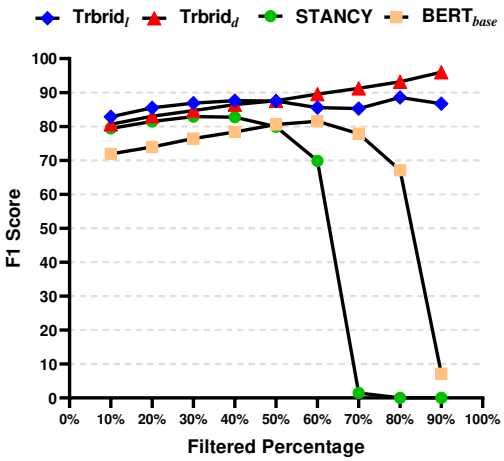


Figure 3: F_1 varying τ with various approaches on PER

shows that our proposed neural architecture, which is trained and used including negated perspectives, is able to return higher-quality predictions than existing methods. Second, *Tribrid*$_d$ (Λ^τ) can achieve a very high F_1; above 95 in the most selective case. However, if we are not willing to sacrifice a large part of the input, then *Tribrid*$_l$ returns better performance. For instance, with a discard rate of 30%, then *Tribrid*$_l$ returns an F_1 of 86.88 vs. 84.69 obtained with *Tribrid*$_d$. Finally, it is remarkable that both *Tribrid*$_d$ and *Tribrid*$_l$ can achieve a very high accuracy while retaining a sizable part of the input. We argue that in scenarios such as social media even if we remove 30-40% of the available perspectives then we are still left with enough data for the downstream task (e.g., fact-checking). Clearly, this does not hold in contexts where all data is needed. In this case, *Tribrid*$_w$ is more appropriate.

4.2 Stance Classification with *Tribrid*$_w$

To exploit the weakly supervised model used in *Tribrid*$_w$ (*FlyingSquid*), we created five classifiers with different threshold values. Then, we performed grid search and feature ablation using the validation dataset and F_1 as metric to optimize. For Λ^τ, we considered threshold values in the range $[0.01, 2]$ while for K^τ the range was $[1, 20]$. We then picked the best performed settings, namely

five τ with the values $\langle 0.01, 0.2, 1.3, 1.5, 1.9\rangle$ for Λ^{τ}, and five K^{τ} classifiers setting τ with the values $\langle 5, 5.5, 8.5, 11, 13\rangle$. We applied each classifier to the input and construct a feature vector with $2n$ labels for every $\langle C, P, NP\rangle$. To train the probabilistic model of *FlyingSquid*, we used the labels obtained from the claims and perspectives in the training set.

Table 5 reports the results obtained with *Tribrid*$_w$ and with the simpler variant *Tribrid*$_{pos}$, which makes no use of negation. Moreover, it also reports the results with several alternatives. First, we considered BERT$_{base}$, STANCY, and majority voting as alternative to weak supervision. Then, we trained additional BERT$_{base}$ and STANCY models considering only the negated perspectives. We used the logits produced by these models and the ones produced by the models trained with the original perspectives to create 10 K^{τ} classifiers. In this way, we could evaluate our weak supervision approach using BERT$_{base}$ and STANCY instead of our neural model. We call these last two baselines BERT$_{ba}$/N and STANC/N, respectively.

Notice that Table 5 does not include the results obtained with the method by Schiller et al. (2020) because it uses BERT large and external datasets with transfer learning. Thus, it cannot be directly compared to our approach. For fairness, we mention that their best result is a F_1 of about 84 on PER. This makes transfer learning a promising extension of our work, but this deserves a dedicated study. Finally, notice that if it is not possible to negate the input perspective, then *Tribrid*$_w$ applies the fallback strategy of executing *Tribrid*$_{pos}$. Therefore, the results presented in Table 5 were obtained considering the entire testsets.

Models	**PER** **F1 (Prec., Rec.)**	**IBMCS** **F1 (Prec., Rec.)**
Random	50.11	48.64
Majority	34.66	34.06
BERT$_{base}$	70.80 (70.50, 71.10)	63.99 (64.13, 63.86)
STANCY	77.76 (81.14, 74.65)	-
BERT$_{ba}$/N	73.83 (65.85, 84.16)	-
STANC/N	78.13 (70.19, 88.10)	-
Tribrid$_{pos}$	80.40 (80.33, 80.49)	70.75 (69.02, 72.57)
Tribrid$_w$	**81.35** (81.00, 81.71)	**71.16** (71.92, 70.42)
Human	90.90 (91.30, 90.60)	-

Table 5: *Tribrid*$_{pos}$ and *Tribrid*$_w$ vs random baseline (Random), majority baseline (Majority), BERT$_{base}$, STANCY, BERT$_{base}$ with negation (BERT$_{ba}$/N), STANCY with negation (STANC/N), and human performance (Human)

Popular Templates	**Coverage**
[X] is/was/are/were [Y]	28.10%
[X] will/would/can/could/shall/ should/may/might/must [Y]	26.28%
[X] to do (or any verb) [Y]	8.18%
[X] have/has [Y]	7.00%
[X] benefit/help [Y]	1.53%
...	...
Total	91.36%

Table 6: Most popular templates on PER. The coverage shows the % of cases where the template matches (for conciseness, we report only the template's premise)

Negation	**Coverage**	**BERT**$_{base}$		**STANCY**		***Tribrid***$_{pos}$	
		#Cases	F_1	#Cases	F_1	#Cases	F_1
AppSuff	**100%**	935	76.2	974	81.3	1130	85.4
DelNot	15.58%	200	51.8	232	60.7	269	74.4
Ours	91.36%	**1309**	**78.0**	**1332**	**82.3**	**1298**	**87.3**

Table 7: Comparison about negating the text on PER

After looking at Table 5, we make *two* main observations. First, *Tribrid*$_{pos}$ slightly outperforms STANCY on PER (80.40 vs 77.76). This means that our strategy of processing the claim and perspective separately, instead concatenating them is beneficial. We argue that this is because with our approach BERT receives shorter strings, thus it is able to produce latent representations of higher quality. Besides, the way we used for concatenating the representations could also lead to an additional increase of the performance.

Second, *Tribrid*$_w$ further outperforms *Tribrid*$_{pos}$, which was the second best, with a difference that is statistically significant (p-value of 3.610e-16 with the McNemar test). Moreover, if we compare the performance of BERT$_{base}$ and STANCY with and without the negated perspectives (BERT$_{base}$ vs. BERT$_{ba}$/N and STANCY vs. STANC/N), then we observe that the F_1 increases also in these cases. This suggests that the strategy of including negative perspectives and to post-process the output in a weakly supervised fashion is a viable solution to improve the performance over the entire input.

4.3 Templates for Negating Perspectives

Our approach heavily depends on the quality of the templates. For us, a good template is not necessarily a template that alters the meaning in a way that is considered optimal by a human. Instead, it is a template that alters the text in a way that improves stance prediction. Moreover, a good template should not be too specific so that it can be applied to as much text as possible.

Table 6 reports a list of the most popular templates on PER. As we can see, a very simple template like the top one matches a large number of cases. One may wonder what the performance would be if we use instead one of the other known techniques for negating the text. Table 7 shows what would happen if we use the methodologies proposed by Bilu et al. (2015) and by Camburu et al. (2020), which are the two most prominent approaches in the current literature. The first method appends the suffix "but this is not true" while the second removes the token "not". Therefore, we call them "AppSuff" and "DelNot", respectively. Since the goal of negating the perspectives is to recognize dubious predictions, we focus on the number of "flipped" cases, that is the number of cases where the outcome changes if we pass the negated text. For instance, suppose that the outcome with perspective A is S. Then, we expect that if we provide $\neg A$, then the output is O. If this happens, then we count this case as "flipped".

Table 7 reports the number of flipped cases and the F_1 that is obtained on the subset with such cases. Notice that here we consider only models that have not been trained with negated information to avoid that they learn some biases. As we can see, the number of flipped cases and F_1 are superior with our templates than with the other two methods. This shows that negating using templates produces sentences that can be recognized more easily by BERT. Because of this, BERT can return an opposite prediction in a larger number of cases and with a better accuracy.

We conclude mentioning a couple of "too hard" cases for *Tribrid*, even with a high τ. The first relates to the claim "We should drop the sanctions against Cuba" and perspective "Sanctions are not working". We suspect that the problem here is that "not" produces a double negation that confuses the model. The second is the claim "Animal testing should be banned" and perspective "Animals do not have rights, therefore it is acceptable to experiment on them". In this case, computing the semantics of the perspective requires some entailment that is likely to be too complex for the model.

5 Conclusion

In this paper, we introduced a new method to classify the stance of messages about a given claim. The main idea is to "inject" negated perspectives that are automatically generated into a BERT-based model so that we can filter out dubious predictions or to improve the overall accuracy.

If we filter out dubious predictions, then we can improve the performance to a point where the F_1 reaches a human-like level without sacrificing a large part of the input. We believe that discarding a (small) percentage of the input is not a major issue in data-intensive environments (e.g., social networks) where many users express their perspectives. However, if the use case is such that we must always make a prediction, then we have shown how we can leverage weak supervision to make a judicious prediction based on the confidence of the model. Also this approach is competitive against the state-of-the-art on standard benchmark datasets.

Our work opens the door to several follow-up studies. A natural continuation is to explore whether we can achieve similar results if we negate the claim instead of the perspective. Moreover, it is interesting to see whether we can construct paraphrases instead of negated text and modify the architecture accordingly. If we increase the number of sequences that we pass as inputs, our "siamese" approach may no longer work. If this occurs, then future work is needed to find some alternatives. Finally, more sophisticated ways to negate the text may lead to further improvements.

In general, we believe that critically assessing the output of a BERT model using negated text is a promising technique to evaluate the model's confidence. Therefore, it can also bring some improvements in other tasks like sentiment analysis, entity linking, or word sense disambiguation.

References

Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats Rule and Dogs Drool!: Classifying Stance in Online Debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance Classification of Context-Dependent Claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic Claim Negation: Why, How and When. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93.

Luís Borges, Bruno Martins, and Pável Calado. 2019. 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.

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.

Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.

Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2020. Make Up Your Mind! Adversarial Generation of Inconsistent Natural Language Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4157–4165.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information Credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, page 675–684.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 542–557.

Wei-Fan Chen and Lun-Wei Ku. 2016. UTCNN: a Deep Learning Model of Stance Classification on Social Media Text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1635–1645.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3601–3608.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational Fact Checking from Knowledge Networks. *PLOS ONE*, 10(6):e0128193.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

L Derczynski, K Bontcheva, M Liakata, R Procter, GWS Hoi, and A Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining Rumour Veracity and Support for Rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance Classification with Target-specific Neural Attention Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 3988–3994.

Adam Faulkner. 2014. 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*, pages 174–179.

Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. 2020. Fast and Three-rious: Speeding Up Weak Supervision with Triplet Methods. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119, pages 3280–3291.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1307–1323.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Volume 1 (Long Papers), pages 1930–1940.

Andreas Hanselowski, PVS Avinesh, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. ClaimBuster: The first-ever End-to-end Fact-checking System. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A Challenge Set Approach to Evaluating Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

Elena Kochkina and Maria Liakata. 2020. Estimating Predictive Uncertainty for Rumour Verification Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6964–6981.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 475–480.

Gina-Anne Levow, Valerie Freeman, Alena Hrynkevich, Mari Ostendorf, Richard Wright, Julian Chan, Yi Luan, and Trang Tran. 2014. Recognition of Stance Strength and Polarity in Spontaneous Speech. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 236–241.

Michal Lukasik, P. K. 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 the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–398.

Taylor Mahler, Willy Cheung, Micha Elsner, David King, Marie-Catherine de Marneffe, Cory Shain, Symon Stevens-Guille, and Michael White. 2017. Breaking NLP: Using Morphosyntax, Semantics, Pragmatics and World Knowledge to Fool Sentiment Analysis Systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 33–39.

Amita Misra and Marilyn Walker. 2013. Topic Independent Identification of Agreement and Disagreement in Social Media Dialogue. In *Proceedings of the SIGDIAL 2013 Conference*, pages 41–50.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic Stance Detection Using End-to-End Memory Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 767–776.

Akiko Murakami and Rudy Raymond. 2010. Support or Oppose? Classifying Positions in Online Debates from Reply Activities and Opinion Expressions. In *Coling 2010: Posters*, pages 869–875.

Tong Niu and Mohit Bansal. 2018. Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 486–496.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2019. STANCY: Stance Classification Based on Consistency Cues. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6414–6419.

Ashwin Rajadesingan and Huan Liu. 2014. Identifying Users with Opposing Opinions in Twitter Debates. In *International conference on social computing, behavioral-cultural modeling, and prediction*, pages 153–160.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

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*.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2020. Stance Detection Benchmark: How Robust Is Your Stance Detection? *arXiv preprint arXiv:2001.01565*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A Dataset for Multi-target Stance Detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2019. Exploring deep neural networks for multi-target stance detection. *Computational Intelligence*, 35(1):82–97.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing Stances in Online Debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234.

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint Models of Disagreement and Stance in Online Debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 116–125.

Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective Stance Classification of Posts in Online Debate Forums. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 109–117.

Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance Detection with Hierarchical Attention Network. In *Proceedings of the 27th international conference on computational linguistics*, pages 2399–2409.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get Out the Vote: Determining Support or Opposition from Congressional Floor-debate Transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335.

Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task Definition and Dataset Construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22.

Marilyn Walker, Pranav Anand, Rob 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*, pages 592–596.

Wenqi Wang, Benxiao Tang, Run Wang, Lina Wang, and Aoshuang Ye. 2019. A Survey on Adversarial Attacks and Defenses in Text. *arXiv preprint arXiv:1902.07285*.

Penghui Wei, Junjie Lin, and Wenji Mao. 2018. Multi-target Stance Detection via a Dynamic Memory-augmented Network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1229–1232.

Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. 2016. pkudblab at SemEval-2016 Task 6: A Specific Convolutional Neural Network System for Effective Stance Detection. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 384–388.

Baicen Xiao, Hossein Radha Poovendran, et al. 2017. Deceiving Google's Cloud Video Intelligence API Built for Summarizing Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–5.

Shaodian Zhang, Lin Qiu, Frank Chen, Weinan Zhang, Yong Yu, and Noémie Elhadad. 2017. We Make Choices we Think are Going to Save Us: Debate and Stance Identification For Online Breast Cancer CAM Discussions. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1073–1081.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

A Templates

The list of 14 templates used to negate the perspectives is reported below.

Templates		
[X] A [Y]	⇒	*[X] A not [Y]*
[X] B [Y]	⇒	*[X] B not [Y]*
[X] C [Y]	⇒	*[X] not C [Y]*
[X] D [Y]	⇒	*[X] don't/doesn't D [Y]*
[X] benefit/help [Y]	⇒	*[X] harm [Y]*
[X] allow [Y]	⇒	*[X] disallow [Y]*
[X] not/n't [Y]	⇒	*[X] [Y]*
[X] more [Y]	⇒	*[X] less [Y]*
[X] need [Y]	⇒	*[X] don't need [Y]*
[X] E [Y]	⇒	*[X] protect [Y]*
[X] cause [Y]	⇒	*[X] cause no [Y]*
[X] help [Y]	⇒	*[X] spoil [Y]*
[X] increase [Y]	⇒	*[X] decrease [Y]*
[X] everyone [Y]	⇒	*[X] no one [Y]*

A=is/was/are/were
B=will/would/can/could/shall/should/may/might/must
C=to do (or any verb)
D=have/has
E=hurt/harm/damage