Stance Detection with Bidirectional Conditional Encoding

Isabelle Augenstein and **Tim Rocktäschel**
Department of Computer Science
University College London
i.augenstein@ucl.ac.uk, t.rocktaschel@cs.ucl.ac.uk

Andreas Vlachos and **Kalina Bontcheva**
Department of Computer Science
University of Sheffield
{a.vlachos, k.bontcheva}@sheffield.ac.uk

Abstract

Stance detection is the task of classifying the attitude expressed in a text towards a target such as *Hillary Clinton* to be "positive", "negative" or "neutral". Previous work has assumed that either the target is mentioned in the text or that training data for every target is given. This paper considers the more challenging version of this task, where targets are not always mentioned and no training data is available for the test targets. We experiment with conditional LSTM encoding, which builds a representation of the tweet that is dependent on the target, and demonstrate that it outperforms encoding the tweet and the target independently. Performance is improved further when the conditional model is augmented with bidirectional encoding. We evaluate our approach on the SemEval 2016 Task 6 Twitter Stance Detection corpus achieving performance second best only to a system trained on semi-automatically labelled tweets for the test target. When such weak supervision is added, our approach achieves state–of-the-art results.

1 Introduction

The goal of stance detection is to classify the attitude expressed in a text towards a given target, as "positive", "negative", or "neutral". Such information can be useful for a variety of tasks, e.g. Mendoza et al. (2010) showed that tweets stating actual facts were affirmed by 90% of the tweets related to them, while tweets conveying false information were predominantly questioned or denied. In this paper we focus on a novel stance detection task, namely tweet stance detection towards previously unseen targets (mostly entities such as politicians or issues of public interest), as defined in the SemEval Stance Detection for Twitter task (Mohammad et al., 2016). This task is rather difficult, firstly due to not having training data for the targets in the test set, and secondly, due to the targets not always being mentioned in the tweet. For example, the tweet "@realDonaldTrump is the only honest voice of the @GOP" expresses a positive stance towards the target *Donald Trump*. However, when stance is annotated with respect to *Hillary Clinton* as the implicit target, this tweet expresses a negative stance, since supporting candidates from one party implies negative stance towards candidates from other parties.

Thus the challenge is twofold. First, we need to learn a model that interprets the tweet stance towards a target that might not be mentioned in the tweet itself. Second, we need to learn such a model without labelled training data for the target with respect to which we are predicting the stance. In the example above, we need to learn a model for *Hillary Clinton* by only using training data for other targets. While this renders the task more challenging, it is a more realistic scenario, as it is unlikely that labelled training data for each target of interest will be available.

To address these challenges we develop a neural network architecture based on conditional encoding (Rocktäschel et al., 2016). A long-short term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) is used to encode the target, followed by a second LSTM that encodes the tweet using the encoding of the target as its initial state. We show that this approach achieves better F1 than an SVM baseline, or an independent LSTM encoding of the tweet and the target. Results improve further (0.4901 F1) with a bidirectional version of our model, which takes into account the context on either side of the word being encoded. In the context of the shared task, this would have been the second best result, except for an approach which uses automatically labelled tweets for the test targets (F1 of 0.5628). Lastly, when our bidirectional conditional encoding model is trained on such data, it achieves state-of-the-art performance (0.5803 F1).

2 Task Setup

The SemEval 2016 Stance Detection for Twitter shared task (Mohammad et al., 2016) consists of two subtasks, Task A and Task B. In Task A the goal is to detect the stance of tweets towards targets given labelled training data for all test targets (*Climate Change is a Real Concern*, *Feminist Movement*, *Atheism*, *Legalization of Abortion* and *Hillary Clinton*). In Task B, which is the focus of this paper, the goal is to detect stance with respect to an unseen target, *Donald Trump*, for which labeled training/development data is not provided.

Systems need to classify the stance of each tweet as "positive" (FAVOR), "negative" (AGAINST) or "neutral" (NONE) towards the target. The official metric reported for the shared task is F1 macro-averaged over the classes FAVOR and AGAINST. Although the F1 of NONE is not considered, systems still need to predict it to avoid precision errors for the other two classes.

Even though participants were not allowed to manually label data for the test target *Donald Trump*, they were allowed to label data automatically. The two best-performing systems submitted to Task B, pkudblab (Wei et al., 2016) and LitisMind (Zarrella and Marsh, 2016) made use of this, thus changing the task to weakly supervised seen target stance detection, instead of an unseen target task. Although the goal of this paper is to present stance detection methods for targets for which no training data is available, we show that they can also be used successfully in a weakly supervised framework and outperform the state-of-the-art on the SemEval 2016 Stance Detection for Twitter dataset.

3 Methods

A common stance detection approach is to treat it as a sentence-level classification task similar to sentiment analysis (Pang and Lee, 2008; Socher et al., 2013). However, such an approach cannot capture the stance of a tweet with respect to a particular target, unless training data is available for each of the test targets. In such cases, we could learn that a tweet mentioning *Donald Trump* in a positive manner expresses a negative stance towards *Hillary Clinton*. Despite this limitation, we use two such baselines, one implemented with a Support Vector Machine (SVM) classifier and one with an LSTM network, in order to assess whether we are successful in incorporating the target in stance prediction.

A naive approach to incorporate the target in stance prediction would be to generate features concatenating the target with words from the tweet. Ignoring the issue that such features would be rather sparse, a classifier could learn that some words have target-dependent stance weights, but it still assumes that training data is available for each target.

In order to learn how to combine the stance target with the tweet in a way that generalises to unseen targets, we focus on learning distributed representations and ways to combine them. The following sections develop progressively the proposed bidirectional conditional LSTM encoding model, starting from independently encoding the tweet and the target using LSTMs.

3.1 Independent Encoding

Our initial attempt to learn distributed representations for the tweets and the targets is to encode the target and tweet independently as k-dimensional dense vectors using two LSTMs (Hochreiter and Schmidhuber, 1997).

$$
\begin{aligned}
\mathbf{H} &= \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \\
\mathbf{i}_t &= \sigma(\mathbf{W}^i \mathbf{H} + \mathbf{b}^i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^f \mathbf{H} + \mathbf{b}^f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o \mathbf{H} + \mathbf{b}^o) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}^c \mathbf{H} + \mathbf{b}^c) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}
$$

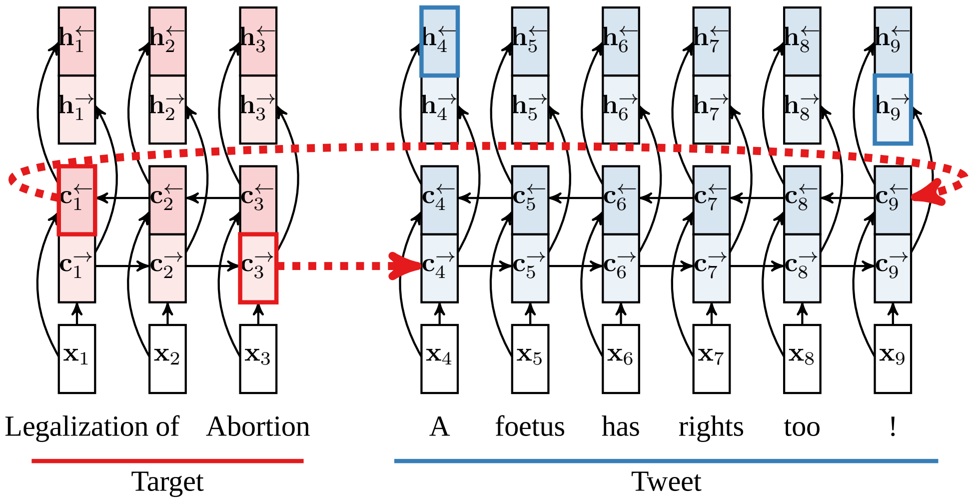

Figure 1: Bidirectional encoding of tweet conditioned on bidirectional encoding of target ($[\mathbf{c}_3^{\rightarrow}\ \mathbf{c}_1^{\leftarrow}]$). The stance is predicted using the last forward and reversed output representations ($[\mathbf{h}_9^{\rightarrow}\ \mathbf{h}_4^{\leftarrow}]$).

Here, $\mathbf{x}_t$ is an input vector at time step t, $\mathbf{c}_t$ denotes the LSTM memory, $\mathbf{h}_t \in \mathbb{R}^k$ is an output vector and the remaining weight matrices and biases are trainable parameters. We concatenate the two output vector representations and classify the stance using the softmax over a non-linear projection

$$\text{softmax}(\tanh(\mathbf{W}^{\text{ta}}\mathbf{h}_{\text{target}} + \mathbf{W}^{\text{tw}}\mathbf{h}_{\text{tweet}} + \mathbf{b}))$$

into the space of the three classes for stance detection where $\mathbf{W}^{\text{ta}}, \mathbf{W}^{\text{tw}} \in \mathbb{R}^{3\times k}$ are trainable weight matrices and $\mathbf{b} \in \mathbb{R}^3$ is a trainable class bias. This model learns target-independent distributed representations for the tweets and relies on the non-linear projection layer to incorporate the target in the stance prediction.

3.2 Conditional Encoding

In order to learn target-dependent tweet representations, we use conditional encoding as previously applied to the task of recognising textual entailment (Rocktäschel et al., 2016). We use one LSTM to encode the target as a fixed-length vector. Then, we encode the tweet with another LSTM, whose state is initialised with the representation of the target. Finally, we use the last output vector of the tweet LSTM to predict the stance of the target-tweet pair.

Formally, let $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ be a sequence of target word vectors, $(\mathbf{x}_{T+1}, \ldots, \mathbf{x}_N)$ be a sequence of tweet word vectors and $[\mathbf{h}_0\ \mathbf{c}_0]$ be a start state of zeros. The two LSTMs map input vectors and a previous state to a next state as follows:

$$\begin{aligned}
[\mathbf{h}_1\ \mathbf{c}_1] &= \text{LSTM}^{\text{target}}(\mathbf{x}_1, \mathbf{h}_0, \mathbf{c}_0)\\
&\ldots\\
[\mathbf{h}_T\ \mathbf{c}_T] &= \text{LSTM}^{\text{target}}(\mathbf{x}_T, \mathbf{h}_{T-1}, \mathbf{c}_{T-1})\\
[\mathbf{h}_{T+1}\ \mathbf{c}_{T+1}] &= \text{LSTM}^{\text{tweet}}(\mathbf{x}_{T+1}, \mathbf{h}_0, \mathbf{c}_T)\\
&\ldots\\
[\mathbf{h}_N\ \mathbf{c}_N] &= \text{LSTM}^{\text{tweet}}(\mathbf{x}_N, \mathbf{h}_{N-1}, \mathbf{c}_{N-1})
\end{aligned}$$

Finally, the stance of the tweet w.r.t. the target is classified using a non-linear projection

$$\mathbf{c} = \tanh(\mathbf{W}\mathbf{h}_N)$$

where $\mathbf{W} \in \mathbb{R}^{3\times k}$ is a trainable weight matrix. This effectively allows the second LSTM to read the tweet in a target-specific manner, which is crucial since the stance of the tweet depends on the target (recall the Donald Trump example above).

3.3 Bidirectional Conditional Encoding

Bidirectional LSTMs (Graves and Schmidhuber, 2005) have been shown to learn improved representations of sequences by encoding a sequence from left to right and from right to left. Therefore, we adapt the conditional encoding model from Section 3.2 to use bidirectional LSTMs, which represent the target and the tweet using two vectors for each of them, one obtained by reading the target and then the tweet left-to-right (as in the conditional LSTM encoding) and one obtained by reading them right-to-left. To achieve this, we initialise the state of the bidirectional LSTM that reads the tweet by the last state of the forward and reversed encoding of the target (see Figure 1). The bidirectional encoding allows the model to construct target-dependent representations of the tweet such that when a word is considered, both its left- and the right-hand side context are taken into account.

3.4 Unsupervised Pretraining

In order to counter-balance the relatively small amount of training data available (5,628 instances in total), we employ unsupervised pre-training by initialising the word embeddings used in the LSTMs with an appropriately trained word2vec model (Mikolov et al., 2013). Note that these embeddings are used only for initialisation, as we allow them to be optimised further during training.

In more detail, we train a word2vec model on a corpus of 395,212 unlabelled tweets, collected with the Twitter Keyword Search API[1] between November 2015 and January 2016, plus all the tweets contained in the official SemEval 2016 Stance Detection datasets (Mohammad et al., 2016). The unlabelled tweets are collected so that they contain the targets considered in the shared task, using up to two keywords per target, namely "hillary", "clinton", "trump", "climate", "femini", "aborti". Note that Twitter does not allow for regular expression search, so this is a free text search disregarding possible word boundaries. We combine this large unlabelled corpus with the official training data and train a skip-gram word2vec model (dimensionality 100, 5 min words, context window of 5).

Tweets and targets are tokenised with the Twitter-adapted tokeniser twokenize[2]. Subsequently, all tokens are lowercased, URLs are removed, and stopword tokens are filtered (i.e. punctuation characters, Twitter-specific stopwords ("rt", "#semst", "via").

As it will be shown in our experiments, unsupervised pre-training is quite helpful, since it is difficult to learn representations for all the words using only the relatively small training datasets available.

[1] `https://dev.twitter.com/rest/public/search`

[2] `https://github.com/leondz/twokenize`

Corpus	**Favor**	**Against**	**None**	**All**
TaskA_Tr+Dv	1462	2684	1482	5628
TaskA_Tr+Dv_HC	224	722	332	1278
TaskB_Unlab	-	-	-	278,013
TaskB_Auto-lab*	4681	5095	4026	13,802
TaskB_Test	148	299	260	707
Crawled_Unlab*	-	-	-	395,212

Table 1: Data sizes of available corpora. TaskA_Tr+Dv_HC is the part of TaskA_Tr+Dv with tweets for the target Hillary Clinton only, which we use for development. TaskB_Auto-lab is an automatically labelled version of TaskB_Unlab. Crawled_Unlab is an unlabelled tweet corpus collected by us.

Finally, to ensure that the proposed neural network architectures contribute to the performance, we also use the word vectors from word2vec to develop a Bag-of-Word-Vectors baseline (BOWV), in which the tweet and target representations are fed into a logistic regression classifier with L2 regularization (Pedregosa et al., 2011).

4 Experiments

Experiments are performed on the SemEval 2016 Task 6 corpus for Stance Detection on Twitter (Mohammad et al., 2016). We report experiments for two different experimental setups: one is the *unseen target* setup (Section 5), which is the main focus of this paper, i.e. detecting the stance of tweets towards previously unseen targets. We show that conditional encoding, by reading the tweets in a target-specific way, generalises to unseen targets better than baselines which ignore the target. Next, we compare our approach to previous work in a *weakly supervised framework* (Section 6) and show that our approach outperforms the state-of-the-art on the SemEval 2016 Stance Detection Subtask B corpus.

Table 1 lists the various corpora used in the experiments and their sizes. TaskA_Tr+Dv is the official SemEval 2016 Twitter Stance Detection TaskA training and development corpus, which contain instances for the targets *Legalization of Abortion*, *Atheism*, *Feminist Movement*, *Climate Change is a Real Concern* and *Hillary Clinton*. TaskA_Tr+Dv_HC is the part of the corpus which contains only the *Hillary Clinton* tweets, which we use for development purposes. TaskB_Test is the TaskB test corpus on which we report results containing *Donald Trump* testing instances. TaskB_Unlab is an unlabelled corpus containing *Donald Trump* tweets supplied by the task organisers, and TaskB_Auto-lab* is an automatically labelled version of a small portion of the corpus for the weakly supervised stance detection experiments reported in Section 6. Finally, Crawled_Unlab* is a corpus we collected for unsupervised pre-training (see Section 3.4).

For all experiments, the official task evaluation script is used. Predictions are post processed so that if the target is contained in a tweet, the highest-scoring non-neutral stance is chosen. This was motivated by the observation that in the training data most target-containing tweets express a stance, with only 16% of them being neutral. The code used in our experiments is available from `https://github.com/sheffieldnlp/stance-conditional`.

Method	Stance	P	R	F1
BoWV	FAVOR	0.2444	0.0940	0.1358
	AGAINST	0.5916	0.8626	0.7019
	Macro			0.4188
TweetOnly	FAVOR	0.2127	0.5726	0.3102
	AGAINST	0.6529	0.4020	0.4976
	Macro			0.4039
Concat	FAVOR	0.1811	0.6239	0.2808
	AGAINST	0.6299	0.4504	0.5252
	Macro			0.4030
TarCondTweet	FAVOR	0.3293	0.3649	0.3462
	AGAINST	0.4304	0.5686	0.4899
	Macro			0.4180
TweetCondTar	FAVOR	0.1985	0.2308	0.2134
	AGAINST	0.6332	0.7379	0.6816
	Macro			0.4475
BiCond	FAVOR	0.2588	0.3761	0.3066
	AGAINST	0.7081	0.5802	0.6378
	Macro			**0.4722**

Table 2: Results for the *unseen target* stance detection development setup.

4.1 Methods

We compare the following baseline methods:

- SVM trained with word and character tweet n-grams features (SVM-ngrams-comb) Mohammad et al. (2016)
- a majority class baseline (Majority baseline), reported in (Mohammad et al., 2016)
- bag of word vectors (BoWV) (see Section 3.4)
- independent encoding of tweet and the target with two LSTMs (Concat) (see Section 3.1)
- encoding of the tweet only with an LSTM (TweetOnly) (see Section 3.1)

to three versions of conditional encoding:

- target conditioned on tweet (TarCondTweet)
- tweet conditioned on target (TweetCondTar)
- a bidirectional encoding model (BiCond)

5 Unseen Target Stance Detection

As explained earlier, the challenge is to learn a model without any manually labelled training data for the test target, but only using the data from the Task A targets. In order to avoid using any labelled data for *Donald Trump*, while still having a (labelled) development set to tune and evaluate our models, we used the tweets labelled for *Hillary Clinton* as a development set and the tweets for the remaining four targets as training. We refer to this as the *development setup*, and all models are tuned using this setup. The labelled *Donald Trump* tweets were only used in reporting our final results.

For the final results we train on all the data from the development setup and evaluate on the official Task B test set, i.e. the *Donald Trump* tweets. We refer to this as our *test setup*.

Based on a small grid search using the development setup, the following settings for LSTM-based models were chosen: input layer size 100 (equal to the word embedding dimension), hidden layer size of 60, training for max 50 epochs with initial learning rate 1e-3 using ADAM (Kingma and Ba, 2014) for optimisation, dropout 0.1. Models were trained using cross-entropy loss. The use of one, relatively small hidden layer and dropout help to avoid overfitting.

5.1 Results and Discussion

Results for the unseen target setting show how well conditional encoding is suited for learning target-dependent representations of tweets, and crucially, how well such representations generalise to unseen targets. The best performing method on both de-

Method	Stance	P	R	F1
BoWV	FAVOR	0.3158	0.0405	0.0719
	AGAINST	0.4316	0.8963	0.5826
	Macro			0.3272
TweetOnly	FAVOR	0.2767	0.3851	0.3220
	AGAINST	0.4225	0.5284	0.4695
	Macro			0.3958
Concat	FAVOR	0.3145	0.5270	0.3939
	AGAINST	0.4452	0.4348	0.4399
	Macro			0.4169
TarCondTweet	FAVOR	0.2322	0.4188	0.2988
	AGAINST	0.6712	0.6234	0.6464
	Macro			0.4726
TweetCondTar	FAVOR	0.3710	0.5541	0.4444
	AGAINST	0.4633	0.5485	0.5023
	Macro			0.4734
BiCond	FAVOR	0.3033	0.5470	0.3902
	AGAINST	0.6788	0.5216	0.5899
	Macro			**0.4901**

Table 3: Results for the *unseen target* stance detection test setup.

EmbIni	NumMatr	Stance	P	R	F1
Random	Sing	FAVOR	0.1982	0.3846	0.2616
		AGAINST	0.6263	0.5929	0.6092
		Macro			0.4354
	Sep	FAVOR	0.2278	0.5043	0.3138
		AGAINST	0.6706	0.4300	0.5240
		Macro			0.4189
PreFixed	Sing	FAVOR	0.6000	0.0513	0.0945
		AGAINST	0.5761	0.9440	0.7155
		Macro			0.4050
	Sep	FAVOR	0.1429	0.0342	0.0552
		AGAINST	0.5707	0.9033	0.6995
		Macro			0.3773
PreCont	Sing	FAVOR	0.2588	0.3761	0.3066
		AGAINST	0.7081	0.5802	0.6378
		Macro			**0.4722**
	Sep	FAVOR	0.2243	0.4103	0.2900
		AGAINST	0.6185	0.5445	0.5792
		Macro			0.4346

Table 4: Results for the *unseen target* stance detection development setup using BiCond, with single vs separate embeddings matrices for tweet and target and different initialisations

velopment (Table 2) and test setups (Table 3) is BiCond, which achieves an F1 of 0.4722 and 0.4901 respectively. Notably, Concat, which learns an independent encoding of the target and the tweets, does not achieve big F1 improvements over TweetOnly, which learns a representation of the tweets only. This shows that it is not sufficient to just take the target into account, but is is important to learn target-dependent encodings for the tweets. Models that learn to condition the encoding of tweets on targets outperform all baselines on the test set.

It is further worth noting that the Bag-of-Word-Vectors baseline achieves results comparable with TweetOnly, Concat and one of the conditional encoding models, TarCondTweet, on the dev set, even though it achieves significantly lower performance on the test set. This indicates that the pre-trained word embeddings on their own are already very useful for stance detection. This is consistent with findings of other works showing the usefulness of such a Bag-of-Word-Vectors baseline for the related tasks of recognising textual entailment Bowman et al. (2015) and sentiment analysis Eisner et al. (2016).

Our best result in the test setup with BiCond is the second highest reported result on the Twitter Stance Detection corpus, however the first, third and fourth best approaches achieved their results by automatically labelling *Donald Trump* training data. BiCond for the unseen target setting outperforms the third and fourth best approaches by a large margin (5 and 7 points in Macro F1, respectively), as can be seen in Table 7. Results for weakly supervised stance detection are discussed in Section 6.

Pre-Training Table 4 shows the effect of unsupervised pre-training of word embeddings with a word2vec skip-gram model, and furthermore, the results of sharing of these representations between the tweets and targets, on the development set. The first set of results is with a uniformly Random embedding initialisation in $[-0.1, 0.1]$. PreFixed uses the pre-trained skip-gram word embeddings, whereas PreCont initialises the word embeddings with ones from SkipGram and continues training them during LSTM training. Our results show that, in the absence of a large labelled training dataset, pre-training of word embeddings is more helpful than random initialisation of embeddings. Sing vs Sep shows the difference between using shared vs two separate embeddings matrices for looking up the word embeddings. Sing means the word representations for tweet and target vocabularies are shared, whereas Sep means they are different. Using shared

Method	inTwe	Stance	P	R	F1
Concat	Yes	FAVOR	0.3153	0.6214	0.4183
		AGAINST	0.7438	0.4630	0.5707
		Macro			0.4945
	No	FAVOR	0.0450	0.6429	0.0841
		AGAINST	0.4793	0.4265	0.4514
		Macro			0.2677
TweetCondTar	Yes	FAVOR	0.3529	0.2330	0.2807
		AGAINST	0.7254	0.8327	0.7754
		Macro			0.5280
	No	FAVOR	0.0441	0.2143	0.0732
		AGAINST	0.4663	0.5588	0.5084
		Macro			0.2908
BiCond	Yes	FAVOR	0.3585	0.3689	0.3636
		AGAINST	0.7393	0.7393	0.7393
		Macro			**0.5515**
	No	FAVOR	0.0938	0.4286	0.1538
		AGAINST	0.5846	0.2794	0.3781
		Macro			0.2660

Table 5: Results for the *unseen target* stance detection development setup for tweets containing the target vs tweets not containing the target.

Method	Stance	P	R	F1
BoWV	FAVOR	0.5156	0.6689	0.5824
	AGAINST	0.6266	0.3311	0.4333
	Macro			0.5078
TweetOnly	FAVOR	0.5284	0.6284	0.5741
	AGAINST	0.5774	0.4615	0.5130
	Macro			0.5435
Concat	FAVOR	0.5506	0.5878	0.5686
	AGAINST	0.5794	0.4883	0.5299
	Macro			0.5493
TarCondTweet	FAVOR	0.5636	0.6284	0.5942
	AGAINST	0.5947	0.4515	0.5133
	Macro			0.5538
TweetCondTar	FAVOR	0.5868	0.6622	0.6222
	AGAINST	0.5915	0.4649	0.5206
	Macro			0.5714
BiCond	FAVOR	0.6268	0.6014	0.6138
	AGAINST	0.6057	0.4983	0.5468
	Macro			**0.5803**

Table 6: Stance Detection test results for weakly supervised setup, trained on automatically labelled pos+neg+neutral Trump data, and reported on the official test set.

embeddings performs better, which we hypothesise is because the tweets contain some mentions of targets that are tested.

Target in Tweet vs Not in Tweet Table 5 shows results on the development set for BiCond, compared to the best unidirectional encoding model, TweetCondTar and the baseline model Concat, split by tweets that contain the target and those that do not. All three models perform well when the target is mentioned in the tweet, but less so when the targets are not mentioned explicitly. In the case where the target is mentioned in the tweet, biconditional encoding outperforms unidirectional encoding and unidirectional encoding outperforms Concat. This shows that conditional encoding is able to learn useful dependencies between the tweets and the targets.

6 Weakly Supervised Stance Detection

The previous section showed the usefulness of conditional encoding for unseen target stance detection and compared results against internal baselines. The goal of experiments reported in this section is to compare against participants in the SemEval 2016 Stance Detection Task B. While we consider an *unseen target* setup, most submissions, including the three highest ranking ones for Task B, pkudblab (Wei et al., 2016), LitisMind (Zarrella and Marsh, 2016) and INF-UFRGS (Dias and Becker, 2016) considered a different experimental setup. They automatically annotated training data for the test target *Donald Trump*, thus converting the task into weakly supervised seen target stance detection. The pkudblab system uses a deep convolutional neural network that learns to make 2-way predictions on automatically labelled positive and negative training data for *Donald Trump*. The neutral class is predicted according to rules which are applied at test time.

Since the best performing systems which participated in the shared task consider a weakly supervised setup, we further compare our proposed approach to the state-of-the-art using such a weakly supervised setup. Note that, even though pkudblab, LitisMind and INF-UFRGS also use regular expressions to label training data automatically, the resulting datasets were not available to us. Therefore, we had to develop our own automatic labelling method and dataset, which are publicly available from our code repository.

Weakly Supervised Test Setup For this setup, the unlabelled *Donald Trump* corpus TaskB_Unlab is annotated automatically. For this purpose we created a small set of regular expressions[3], based on inspection of the TaskB_Unlab corpus, expressing positive and negative stance towards the target. The regular expressions for the positive stance were:

- make(?)america(?)great(?)again
- trump(?)(for|4)(?)president
- votetrump
- trumpisright
- the truth
- #trumprules

The keyphrases for negative stance were: #dumptrump, #notrump, #trumpwatch, racist, idiot, fired

A tweet is labelled as positive if one of the positive expressions is detected, else negative if a negative expressions is detected. If neither are detected, the tweet is annotated as neutral randomly with 2% chance. The resulting corpus size per stance is shown in Table 1. The same hyperparameters for the LSTM-based models are used as for the *unseen target* setup described in the previous section.

6.1 Results and Discussion

Table 6 lists our results in the weakly supervised setting. Table 7 shows all our results, including those using the unseen target setup, compared against the state-of-the-art on the stance detection corpus. Table 7 further lists baselines reported by Mohammad et al. (2016), namely a majority class baseline (Majority baseline), and a method using 1 to 3-gram bag-of-word and character n-gram features (SVM-ngrams-comb), which are extracted from the tweets and used to train a 3-way SVM classifier.

Bag-of-word baselines (BoWV, SVM-ngrams-comb) achieve results comparable to the majority baseline (F1 of 0.2972), which shows how difficult the task is. The baselines which only extract features from the tweets, SVM-ngrams-comb and TweetOnly perform worse than the baselines which also learn representations for the targets (BoWV, Concat). By training conditional encoding models on automatically labelled stance detection data we achieve state-of-the-art results. The best result (F1 of 0.5803) is achieved with the bi-directional conditional encoding model (BiCond). This shows that such models are suitable for unseen, as well as seen target stance detection.

Method	**Stance**	**F1**
SVM-ngrams-comb (*Unseen Target*)	FAVOR	0.1842
	AGAINST	0.3845
	Macro	0.2843
Majority baseline (*Unseen Target*)	FAVOR	0.0
	AGAINST	0.5944
	Macro	0.2972
BiCond (*Unseen Target*)	FAVOR	0.3902
	AGAINST	0.5899
	Macro	**0.4901**
INF-UFRGS (*Weakly Supervised**)	FAVOR	0.3256
	AGAINST	0.5209
	Macro	0.4232
LitisMind (*Weakly Supervised**)	FAVOR	0.3004
	AGAINST	0.5928
	Macro	0.4466
pkudblab (*Weakly Supervised**)	FAVOR	0.5739
	AGAINST	0.5517
	Macro	0.5628
BiCond (*Weakly Supervised*)	FAVOR	0.6138
	AGAINST	0.5468
	Macro	**0.5803**

Table 7: Stance Detection test results, compared against the state of the art. SVM-ngrams-comb and Majority baseline are reported in Mohammad et al. (2016), pkudblab in Wei et al. (2016), LitisMind in Zarrella and Marsh (2016), INF-UFRGS in Dias and Becker (2016)

7 Related Work

Stance Detection: Previous work mostly considered target-specific stance prediction in debates (Hasan and Ng, 2013; Walker et al., 2012) or student essays (Faulkner, 2014). The task considered in this paper is more challenging than stance detection in debates because, in addition to irregular language, the Mohammad et al. (2016) dataset is offered without any context, e.g., conversational structure or tweet metadata. The targets are also not always mentioned in the tweets, which is an additional challenge (Augenstein et al., 2016) and distinguishes this task from target-dependent (Vo and Zhang, 2015; Zhang et al., 2016; Alghunaim et al., 2015) and

[3]Note that "|" indiates "or", (?) indicates optional space

open-domain target-dependent sentiment analysis (Mitchell et al., 2013; Zhang et al., 2015). Related work on rumour stance detection either requires training data from the same rumour (Qazvinian et al., 2011), i.e., target, or is rule-based (Liu et al., 2015) and thus potentially hard to generalise. Finally, the target-dependent stance detection task tackled in this paper is different from that of Ferreira and Vlachos (2016), which while related concerned with the stance of a statement in natural language towards another statement.

Conditional Encoding: Conditional encoding has been applied to the related task of recognising textual entailment (Rocktäschel et al., 2016), using a dataset of half a million training examples (Bowman et al., 2015) and numerous different hypotheses. Our experiments here show that conditional encoding is also successful on a relatively small training set and when applied to an unseen testing target. Moreover, we augment conditional encoding with bidirectional encoding and demonstrate the added benefit of unsupervised pre-training of word embeddings on unlabelled domain data.

8 Conclusions and Future Work

This paper showed that conditional LSTM encoding is a successful approach to stance detection for unseen targets. Our unseen target bidirectional conditional encoding approach achieves the second best results reported to date on the SemEval 2016 Twitter Stance Detection corpus. In the weakly supervised seen target scenario, as considered by prior work, our approach achieves the best results to date on the SemEval Task B dataset. We further show that in the absence of large labelled corpora, unsupervised pre-training can be used to learn target representations for stance detection and improves results on the SemEval corpus. Future work will investigate further the challenge of stance detection for tweets which do not contain explicit mentions of the target.

Acknowledgments

This work was partially supported by the European Union under grant agreement No. 611233 PHEME[4] and by Microsoft Research through its PhD Scholarship Programme.

[4] `http://www.pheme.eu`

References

[Alghunaim et al.2015] Abdulaziz Alghunaim, Mitra Mohtarami, Scott Cyphers, and Jim Glass. 2015. A Vector Space Approach for Aspect Based Sentiment Analysis. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 116–122, Denver, Colorado, June. Association for Computational Linguistics.

[Augenstein et al.2016] Isabelle Augenstein, Andreas Vlachos, and Kalina Bontcheva. 2016. USFD: Any-Target Stance Detection on Twitter with Autoencoders. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California.

[Bowman et al.2015] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September. Association for Computational Linguistics.

[Dias and Becker2016] Marcelo Dias and Karin Becker. 2016. INF-UFRGS-OPINION-MINING at SemEval-2016 Task 6: Automatic Generation of a Training Corpus for Unsupervised Identification of Stance in Tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June.

[Eisner et al.2016] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning Emoji Representations from their Description. In *Proceedings of the International Workshop on Natural Language Processing for Social Media*, SocialNLP '16, Austin, Texas.

[Faulkner2014] Adam Faulkner. 2014. Automated Classification of Stance in Student Essays: An Approach Using Stance Target Information and the Wikipedia Link-Based Measure. In William Eberle and Chutima Boonthum-Denecke, editors, *FLAIRS Conference*. AAAI Press.

[Ferreira and Vlachos2016] 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*, pages 1163–1168, San Diego, California, June. Association for Computational Linguistics.

[Graves and Schmidhuber2005] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610.

[Hasan and Ng2013] Kazi Saidul Hasan and Vincent Ng. 2013. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *IJCNLP*, pages 1348–1356. Asian Federation of Natural Language Processing / ACL.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Kingma and Ba2014] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

[Liu et al.2015] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time Rumor Debunking on Twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1867–1870, New York, NY, USA. ACM.

[Mendoza et al.2010] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. 2010. Twitter Under Crisis: Can We Trust What We RT? In *Proceedings of the First Workshop on Social Media Analytics (SOMA'2010)*, pages 71–79, New York, NY, USA. ACM.

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

[Mitchell et al.2013] Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open Domain Targeted Sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654, Seattle, Washington, USA, October. Association for Computational Linguistics.

[Mohammad et al.2016] Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 Task 6: Detecting stance in tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June.

[Pang and Lee2008] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

[Pedregosa et al.2011] 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.

[Qazvinian et al.2011] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor Has It: Identifying Misinformation in Microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1589–1599.

[Rocktäschel et al.2016] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about Entailment with Neural Attention. In *International Conference on Learning Representations (ICLR)*.

[Socher et al.2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

[Vo and Zhang2015] Duy-Tin Vo and Yue Zhang. 2015. Target-Dependent Twitter Sentiment Classification with Rich Automatic Features. In Qiang Yang and Michael Wooldridge, editors, *IJCAI*, pages 1347–1353. AAAI Press.

[Walker et al.2012] 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.

[Wei et al.2016] 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 International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June.

[Zarrella and Marsh2016] Guido Zarrella and Amy Marsh. 2016. MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June.

[Zhang et al.2015] Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015. Neural Networks for Open Domain Targeted Sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621, Lisbon, Portugal, September. Association for Computational Linguistics.

[Zhang et al.2016] Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated Neural Networks for Targeted Sentiment Analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, February. Association for the Advancement of Artificial Intelligence.