USFD at SemEval-2016 Task 6: Any-Target Stance Detection on Twitter with Autoencoders

Isabelle Augenstein and **Andreas Vlachos** and **Kalina Bontcheva**
Department of Computer Science
University of Sheffield
i.augenstein@sheffield.ac.uk, a.vlachos@sheffield.ac.uk,
k.bontcheva@sheffield.ac.uk

Abstract

This paper describes the University of Sheffield's submission to the SemEval 2016 Twitter Stance Detection weakly supervised task (SemEval 2016 Task 6, Subtask B). In stance detection, the goal is to classify the stance of a tweet towards a target as "favor", "against", or "none". In Subtask B, the targets in the test data are different from the targets in the training data, thus rendering the task more challenging but also more realistic. To address the lack of target-specific training data, we use a large set of unlabelled tweets containing all targets and train a bag-of-words autoencoder to learn how to produce feature representations of tweets. These feature representations are then used to train a logistic regression classifier on labelled tweets, with additional features such as an indicator of whether the target is contained in the tweet. Our submitted run on the test data achieved an F1 of 0.3270.

1 Introduction

Stance detection is the task of assigning stance labels to a piece of text with respect to a topic, i.e. whether a piece of text is in favour of "abortion", neutral, or against. Previous work considered target-specific stance predictors in debates (Walker et al., 2012; Hasan and Ng, 2013) or news (Ferreira and Vlachos, 2016).

The variety of topics discussed on Twitter calls for developing methods that can generalise to any target, including targets not seen in the training data, which is the focus of Subtask B in Task 6 of SemEval 2016 (Mohammad et al., 2016). A further challenge is that the targets are not always mentioned in the tweets, which distinguishes this task from target-dependent sentiment analysis (Zhang et al., 2015; Zhang et al., 2016), and open-domain target-dependent sentiment analysis (Mitchell et al., 2013; Vo and Zhang, 2015).

The SemEval Stance Detection task is further related to that of textual entailment (Dagan et al., 2005; Bowman et al., 2015; Lendvai et al., 2016), i.e. we judge if a hypothesis (tweet in our task) entails, contradicts or is neutral towards a textual premise (target in our task). However, the premises in typical RTE datasets offer a richer context than the stance detection targets, i.e. they are full sentences instead of topic labels such as "atheism". Simple baselines such as textual overlap can achieve an F1 of >0.5 (Bowman et al., 2015), whereas for stance detection such baselines would not perform well, as the target is only mentioned in about half the tweets.

In our approach we learn a 3-way logistic regression classifier to perform stance detection. Apart from the standard bag-of-words features commonly used in sentiment analysis, we also use features from a trained bag-of-words autoencoder similar to the one used by Glorot et al. (2011). In our experiments we show that the bag-of-words autoencoder trained on a large amount of unlabelled tweets about the targets can help generalise to unseen targets better; on our development set it achieves an 8% increase over our best baseline. Further, tweets which contain the target are easier to classify correctly than tweets which do not contain the target. Such information can be useful for stance detection and we experiment with different ways of integrating it, finding that including a binary feature "targetContainedIn-

Tweet" outperforms including features extracted by applying the autoencoder to the target.

2 Method Description

At the core of our stance detection approach is a classifier trained on tweets stance-labelled with respect to a target. For this purpose we used the logistic regression classifier from scikit-learn with L2 regularisation (Pedregosa et al., 2011)[1]. In what follows we describe the various feature representations we used and the data pre-processing. Resources to reproduce our experiments are available on Github[2].

The stages of our approach are: a) unlabelled tweets about the targets; b) preprocess the data; c) train a bag-of-word autoencoder on all task data and unlabelled collected tweets, d) apply the autoencoder to all labelled training tweets to get a fixed-length feature vector; add a "does target appear in tweet" feature; and e) train a logistic regression model and apply it to the test tweets.

2.1 Autoencoder Training

After tweets are tokenised, a bag-of-word autoencoder is trained on them. To do so, a vocabulary of the 50000 most frequent words is constructed. The input to the autoencoder is a vector $input\_dim$ for each training example of size 50000. Each index i in $input\_dim[i]$ corresponds to a word in the vocabulary, $input\_dim[i]$ is 1 if the tweet contains the corresponding word in the vocabulary and 0 otherwise. During autoencoder training, an encoder, i.e. embedding function is learned which maps input of size $input\_dim$ to an embedding of size $output\_dim$, as well as a decoder which reconstructs the input. We apply the encoder to the training and test data to obtain features of size $output\_dim$ for supervised learning and disregard the decoder. While it would be possible to train an encoder which preserves word order, i.e. an LSTM (Li et al., 2015), we opt for a simpler bag-of-word autoencoder here, following Glorot et al. (2011).

The architecture of the autoencoder is as follows: $input\_dim$ is 50000, it has one hidden layer of dimensionality 100, and $output\_dim$ is of size 100. A dropout of 0.1 is added to the hidden layer (Srivastava et al., 2014). The autoencoder is trained with Adam (Kingma and Ba, 2014), using the learning rate 0.1, for 2600 iterations. In each iteration, 500 training examples are selected randomly.

Additional tweets are collected: 395212 tweets, tweeted between 18 November and 13 January, collected with the Twitter Keyword Search API[3] using up to two keywords per target (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.

2.2 Feature Extraction

The autoencoder is applied to the labelled data to get an 100-dimensional feature vector. For the final run, it was only applied to the tweets, but we also experiment with applying it to the target (see Section 3.2).

One additional binary feature is used for the final run, targetInTweet, which indicates if the name of the target is contained in the tweet. The following mapping was used for this purpose: 'Hillary Clinton' → 'hillary', 'clinton'; Donald Trump → 'trump'; 'Climate Change is a Real Concern' → 'climate'; 'Feminist Movement' → 'feminist', 'feminism'; 'Legalization of Abortion' → 'abortion', 'aborting'. Further features, which are not used for the final run, are discussed in Section 3.2.

2.3 Preprocessing

Twitter-based tokenisation is performed with twokenize[4]. Afterwards, tokens are normalised to lower case and stopwords are filtered, using the nltk[5] English stopword list, punctuation characters, plus Twitter-specific stopwords. The latter is manually created and consists of: "rt", "#semst", "thats", "im", "'s", "...", "via", "http". The first seven have to be an exact token match, the last one has to match the beginning of a token. Finally, phrases are detected, using an unsupervised method that creates 2-grams of commonly occurring expression such as "hillary clinton", "donald trump", "hate muslims"

[1] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[2] http://github.com/sheffieldnlp/stance-semeval2016

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

[4] https://github.com/leondz/twokenize

[5] http://www.nltk.org/

(Mikolov et al., 2013)[6]. The phrase detection model is trained on all tweets except the test tweets. At application time, if two subsequent tokens are identified as a phrase, those tokens are merged to one token (i.e. “donald”, “trump” → “donald_trump”).

3 Experiments

3.1 Experimental Setup

Our development setup is to train on all labelled tweets for the targets “Climate Change is a Real Concern”, “Feminist Movement” and “Legalization of Abortion”, then evaluate on “Hillary Clinton” tweets. The motivation for this is that Hillary Clinton is the most semantically related target to the Task B test target Donald Trump, since both entities are persons and politicians. For final submission we tuned all settings with this setup, then retrained on all data and applied the model to the test data.

3.2 Methods

Our goal is to determine if including the target is beneficial and if so, how best to include the it. To this end, the following features are evaluated:

- Aut-twe: the autoencoder is applied to the tweet only
- Aut-twe_tar: the autoencoder is applied to the tweet and the target, the target features are concatenated with the tweet features
- Aut-twe*tar: the autoencoder is applied to the tweet and the target, and the outer product of the tweet and target features is used
- InTwe: A boolean “targetInTweet” feature

We evaluate the impact of traditional sentiment analysis gazetteer features, extracted by assessing appearance of each word of the tweet in the gazetteers:

- Emo: emoticon recognition[7] One gazetteer/binary feature for each of: happy, sad, happy+sad, not_available
- Aff: WordNet Affect gazetteer features, one binary feature for each of: anger, disgust, fear, joy, sadness, surprise (Strapparava and Valitutti, 2004)[8]

Method	Stance	P	R	F1
BoW	FAVOR	0.1587	0.1709	0.1646
	AGAINST	0.5544	0.4020	0.4661
	Macro			0.3153
BoW+inTwe	FAVOR	0.2278	0.1538	0.1837
	AGAINST	0.5545	0.5700	0.5621
	Macro			0.3729
Word2Vec	FAVOR	0.2647	0.0769	0.1192
	AGAINST	0.5179	0.2570	0.3435
	Macro			0.2314
Aut-twe	FAVOR	0.1538	0.1538	0.1538
	AGAINST	0.5680	0.7328	0.6400
	Macro			0.3969
Aut-twe_tar	FAVOR	0.1652	0.1624	0.1638
	AGAINST	0.5503	0.6539	0.5977
	Macro			0.3807
Aut-twe*tar	FAVOR	0.0000	0.0000	0.0000
	AGAINST	0.5712	1.0000	0.7271
	Macro			0.3636
Aut-twe+inTwe	FAVOR	0.2388	0.1368	0.1739
	AGAINST	0.5709	0.7684	0.6551
	Macro			0.4145
Aut-twe_tar+inTwe	FAVOR	0.1731	0.0769	0.1065
	AGAINST	0.5487	0.7888	0.6472
	Macro			0.3768
Aut-twe+inTwe+Emo	FAVOR	0.2063	0.1111	0.1444
	AGAINST	0.5733	0.7964	0.6667
	Macro			0.4056
Aut-twe+inTwe+Aff	FAVOR	0.2113	0.1282	0.1596
	AGAINST	0.5701	0.7964	0.6645
	Macro			0.4121

Table 1: Stance Detection results, reported on the development set. The best results are achieved with Aut-twe+inTwe.

In addition we experiment with substituting the bag of word autoencoder with a word2vec model trained on the same data. We trained a skip-gram model with a dimensionality of 300, 10 min words and a context of 10 with the gensim implementation of word2vec[9]. Word vectors are combined by multiplication to get a fixed-length sentence-level vector. We also report a bag-of-words baseline, which disregards the unlabelled data and extracts unigram and bigram bag-of-word features from the training data. For the word2vec models as well as the bag-of-words baseline, the same preprocessing as for the autoencoder approach is used.

4 Results

Results are reported in Tables 1 and 2 for all the experiments above, using the dev setup (Hillary Clinton) and the test setup (Donald Trump). Overall re-

[6] `https://radimrehurek.com/gensim/models/phrases.html`

[7] `https://github.com/brendano/tweetmotif/blob/master/emoticons.py`

[8] `http://wndomains.fbk.eu/wnaffect.html`

[9] `https://radimrehurek.com/gensim/models/word2vec.html`

Method	Stance	P	R	F1
BoW	FAVOR	0.2933	0.1486	0.1973
	AGAINST	0.4116	0.6154	0.4933
	Macro			0.3453
BoW+inTwe	FAVOR	0.2308	0.0608	0.0963
	AGAINST	0.4121	0.7057	0.5203
	Macro			0.3083
Word2Vec	FAVOR	0.3500	0.0473	0.0833
	AGAINST	0.4155	0.8629	0.5609
	Macro			0.3221
Aut-twe	FAVOR	0.0889	0.0270	0.0415
	AGAINST	0.4266	0.8462	0.5673
	Macro			0.3044
Aut-twe_tar	FAVOR	0.2273	0.0676	0.1042
	AGAINST	0.4192	0.8161	0.5539
	Macro			0.3290
Aut-twe*tar	FAVOR	0.0000	0.0000	0.0000
	AGAINST	0.4229	1.0000	0.5944
	Macro F			0.2972
Aut-twe+inTwe	FAVOR	0.2857	0.0676	0.1093
	AGAINST	0.4087	0.8161	0.5446
	Macro			0.3270
Aut-twe_tar+inTwe	FAVOR	0.3077	0.0541	0.0920
	AGAINST	0.4250	0.8629	0.5695
	Macro			0.3307
Aut-twe+inTwe+Emo	FAVOR	0.4762	0.0676	0.1183
	AGAINST	0.4265	0.8829	0.5752
	Macro			0.3468
Aut-twe+inTwe+Aff	FAVOR	0.2667	0.0541	0.0899
	AGAINST	0.4107	0.8763	0.5592
	Macro			0.3246

Table 2: Stance Detection results, reported on official test. The submitted run is Aut-twe+InTwe.

sults for dev are significantly better than for test, and F1 for AGAINST is consistently higher than for FAVOR. Performance increases for test with respect to baselines are much smaller than for dev. The best results for the dev set are achieved with Aut-twe+inTwe, and it was chosen for the final run on the test set. However, the best results on the test set are achieved with Aut-twe+inTwe+Emo, which is almost on par with the BoW baseline.

The feature that contributes positively to both dev and test performance is inTwe. It was introduced because almost all tweets in the training data that contain the target either FAVOR or are AGAINST the target, but are rarely neutral towards the target. 363 out of 656 Hillary Clinton dev tweets contain the target, and 309 out of 689 Donald Trump test tweets contain the target. We observed that there is a significant difference in performance between tweets containing the target and tweets which do not contain the target (see Tables 3 and 4).

inTwe	Stance	P	R	F1
Yes	FAVOR	0.2830	0.1456	0.1923
	AGAINST	0.7072	0.7237	0.7154
	Macro			0.4538
No	FAVOR	0.0714	0.0714	0.0714
	AGAINST	0.4361	0.8529	0.5771
	Macro			0.3243

Table 3: Stance Detection results, reported on dev, split by tweets containing the target and not containing the target. The run used is Aut-twe+inTwe.

inTwe	Stance	P	R	F1
Yes	FAVOR	0.7778	0.0486	0.0915
	AGAINST	0.5201	0.8931	0.6574
	Macro			0.3745
No	FAVOR	0.0000	0.0000	0.0000
	AGAINST	0.3333	0.8286	0.4754
	Macro			0.2377

Table 4: Stance Detection results, reported on test, split by tweets containing the target and not containing the target. The run used is Aut-twe+inTwe.

Adding autoencoder features for the target did not improve results for dev. For test, tweet features aggregated with target features slightly outperform target features on their own. As for traditional sentiment analysis features, Emo improves macro F1 for test, but not but dev, and Aff does not improve macro F1 for either of them.

5 Conclusions and Future Work

To conclude, we showed that it is important to detect if the target is mentioned in the tweet, and that a bag-of-word autoencoder can help to detect stance towards unseen targets. Further, developing a stance detection method for new targets without any labelled training data is challenging - we found that there are some discrepancies between what features perform well for a development versus a test set. In future work we will investigate how to better incorporate the target for stance detection, as this target-dependence is crucial in capturing that the same tweet can have different stance with respect to different targets that are not mentioned in the tweet.

Acknowledgments

This work was partially supported by the European Union, grant agreement No. 611233 PHEME[10].

[10] http://www.pheme.eu

References

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. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In Joaquin Quionero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alch Buc, editors, *MLCW*, Lecture Notes in Computer Science, pages 177–190. Springer.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of NAACL*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, New York, NY, USA. ACM.

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.

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

Piroska Lendvai, Isabelle Augenstein, Kalina Bontcheva, and Thierry Declerck. 2016. Monolingual Social Media Datasets for Detecting Contradiction and Entailment. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia. European Language Resources Association (ELRA).

Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. A Hierarchical Neural Autoencoder for Paragraphs and Documents. 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 1106–1115, Beijing, China. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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. Association for Computational Linguistics.

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.

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.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Carlo Strapparava and Alessandro Valitutti. 2004. WordNet-Affect: An affective extension of WordNet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1083–1086. ELRA.

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.

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.

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. Association for Computational Linguistics.

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. Association for the Advancement of Artificial Intelligence.