IDI@NTNU at SemEval-2016 Task 6: Detecting Stance in Tweets Using Shallow Features and GloVe Vectors for Word Representation

Henrik Bøhler and **Petter Fagerlund Asla** and **Erwin Marsi** and **Rune Sætre**
Norwegian University of Science and Technology
Sem Sælands vei 9
Trondheim, 7491, NORWAY
{henriboh,pettefas}@stud.ntnu.no, {emarsi,satre}@idi.ntnu.no

Abstract

This paper describes an approach to automatically detect stance in tweets by building a supervised system combining shallow features and pre-trained word vectors as word representation. The word vectors were obtained from several collections of large corpora using GloVe, an unsupervised learning algorithm. We created feature vectors by selecting the word vectors relevant to the data and summing them for each unique word. Combining multiple classifiers into a voting classifier, representing the best of both approaches, shows a significant improvement over the baseline system.

1 Introduction

This paper describes our submission to the SemEval 2016 competition Task 6A - *Detecting Stance in Tweets*. The goal of the task is to classify a tweet into one of the three classes – *against*, *favor* or *none* in regard to a certain topic. These classes represents the tweet's stance towards the given target.

Twitter, and other microblogging platforms, have in recent years become popular arenas to apply natural language processing tasks. One of the most popular tasks has been sentiment analysis. Stance detection differs from sentiment analysis because the sentiment of a text – generally positive or negative – does not necessarily agree with its stance regarding a certain topic of debate. For example, a tweet like "all those climate-deniers are morons" is negative in its overall sentiment, but positive with regard to the stance that climate change is a real concern. We refer the reader to the official SemEval 2016[1] website for a detailed task description.

Our approach to detect stance is based on shallow features (Kohlschütter et al., 2010; Hagen et al., 2015; Walker et al., 2012) and the use of GloVe word vectors (Pennington et al., 2014). During the development phase we explored several approaches by implementing features such as sentiment detection (Hutto and Gilbert, 2014), number of tokens and number of capital words. The experiments later in this paper show that not all features enhanced the performance of the implemented system.

The feature that turned out to boost performance the most, in combination with basic shallow features, was the use of pre-trained GloVe vectors (Pennington et al., 2014). The vector representations of tweets were created by summing the pre-trained word vectors for each unique word. No additional data was added to the training set used for our final submission, although we explored the possibility of gathering and automatically labelling additional tweets by using label propagation (Zhu and Ghahramani, 2002; Zhou et al., 2004). This did enhance our baseline system performance slightly, but not in combination with other features.

2 System description

To predict the stance in tweets we built a supervised machine learning system using the scikit-learn machine learning library (Pedregosa et al., 2011). Our system consists of a soft voting classifier that predicts the class label on the basis of the best re-

[1] http://alt.qcri.org/semeval2016/task6/

Proceedings of SemEval-2016, pages 457–462,
San Diego, California, June 16-17, 2016.

sults out of the three classifiers described in subsection 2.3.

2.1 Resources

Our system used a limited number of resources. It relies on the annotated training data consisting of 2814 tweets divided into five different topics: *Atheism*, *Climate Change is a Real Concern*, *Feminist Movement*, *Hillary Clinton*, and *Legalization of Abortion*. In addition, it uses pre-trained word vectors[2] created by Pennington et al. (2014).

2.1.1 Bootstrapping attempts

The labels for the climate change target showed a highly skewed distribution where only 3.8% were labelled *against*. Skewed data distributions in machine learning are a common problem. Monard and Batista (2002), Provost (2000) and Tang et al. (2009) discuss this problem and suggests several solutions, such as data under- and over-sampling. We did not have time to investigate the effects of these methods, but Elkan (2001) suggest that changing the balance of negative and positive training samples has little effect on learned classifiers.

In an attempt to even out the distribution of the climate change data, we searched for ways to add additional tweets. The most promising approach explored was label propagation (Zhu and Ghahramani, 2002; Zhou et al., 2004), a semi-supervised learning algorithm. Thousands of tweets were fetched based on the most common hashtags found in the climate topic data. We hand-picked a small portion of tweets that seemed relevant to the climate topic (e.g. same language and containing a statement). These tweets were then automatically labelled using label propagation. The label propagation was performed with a (small) representative sample of the labelled training data together with the collected, hand picked, unlabelled tweets. We found that adding more data to our system did not result in substantial improvement. An explanation could be that the gathered tweets were not meaningful enough to be effective. The additional data was therefore not used in subsequent experiments.

[2] http://nlp.stanford.edu/projects/glove/

2.2 Features

The submitted system used the following features, generated from the raw data supplied in the training set.

1. **Word bigrams:** All pairs of consecutive words
   - Punctuation ignored
2. **Character trigram:** All triples of consecutive characters
   - Punctuation ignored
   - Converted to lowercase
   - Ignored terms that had a document frequency strictly lower than 5 (cut-off)
3. **GloVe vectors:** Word embeddings for all words in a tweet
   - Punctuation ignored
   - Converted to lowercase
   - Removed stop words

In addition, we experimented with the following features, which were not included in the final system. They were left out as they did not improve the systems performance (section 3 will provide more details on this).

- **Negation:** Presence of negation in the sentence
- **Length of tweets:** Number of characters divided by the maximum length (140 characters)
- **Capital words:** Number of capital words in the tweet
- **Repeated punctuation:** Number of occurrences of non-single punctuation (e.g. !?)
- **Exclamation mark last:** Exclamation mark found last in non-single punctuation (e.g. ?!)
- **Lengthening of words:** Number of lengthened words (e.g. smoooth)
- **Sentiment:** Detecting sentiment in tweet using the Vader system (Hutto and Gilbert, 2014)
- **Number of tokens:** Count of total number of tokens in the tweet

2.2.1 GloVe

GloVe (Pennington et al., 2014) is an unsupervised learning algorithm for obtaining vector representations of words. It creates word vectors based on the distributional statistics of words, in particular how frequently words co-occur within a certain window in a large text corpus such as the Gigaword corpus (Parker et al., 2011). The resulting word vectors can be used to measure semantic similarity between word pairs, following the hypothesis that similar words tend to have similar distributions. The Euclidean or Cosine distance between two word vectors can thus be used as a measure of their semantic similarity. For the word *frog*, for example, we can find related words such as *frogs, toad, litoria, leptodactylidae, rana, lizard, eleutherodactylu*.

In order to measure the semantic similarity between tweets, rather than isolated words, we needed a way to obtain vector representations of documents. Mitchell and Lapata (2010) looked at the possibility to use word vectors to represent the meaning of word combinations in a vector space. They suggest, among other things, to use vector composition, operationalized in terms of additive (or multiplicative) functions. Accordingly we created vector representations of tweets by combining the vectors of their words. We used pre-trained word vectors created by Pennington et al. (2014) trained on Wikipedia 2014 + Gigaword 5[3] and Twitter data[4]. The word vectors come in several versions with a different number of dimensions (25, 50, 100, 200, 300) that supposedly capture different granularities of meaning. The resulting features (from here on called GloVe features) were obtained by summing the GloVe vectors, per dimension, for all unique terms in a tweet.

2.3 Models

To detect stance we constructed separate models for each of the five topics, each in the form of a soft voting classifier from scikit-learn (Pedregosa et al., 2011). The voting classifiers took input from the following three classifiers:

1. **Multinomial Naive Bayes** trained on word bigrams
2. **Multinomial Naive Bayes** trained on character trigrams
3. **Logistic Regression** trained on GloVe features

The soft voting classifier is – in contrast to a hard voting classifier – able to exploit prediction probabilities from the separate classifiers. For each sample, the soft voting classifier predicts the class based on the argmax of the sums of the predicted probabilities from the input classifiers.

In the task description it was stated that it was not necessary to predict stance for every tweet in the test set, leaving the uncertain ones with an *unknown* label. We decided to use a threshold value, using the extracted probabilities, to prevent predictions with low confidence. Labels predicted with a probability below the threshold were thus changed into *unknown*. Details of the selection of the threshold value are presented at the end of section 4.

Due to the imbalanced distribution of labels in climate change data, our system had a low prediction rate of *against* stances on this target. For that reason we included a second slightly different model for the climate change target. The difference between the first and second model was that the second used a hard (majority rule) voting classifier, which performed slightly better on the against labels in the climate data. The combination of the two models was implemented in a way such that for each of the *against* predictions in the hard voting model, we overwrote the soft model's prediction, labelling the tweet *against*. Our submitted system thus consisted of two models for predicting the climate class, giving a total of six models.

To summarize, the system contains six models, where five of them consist of a soft voting classifier with input from the three different classifiers introduced above. The sixth is a hard voting model that supplements the soft voting model for the climate change target.

3 Results on Development Data

To measure the system performance we conducted multiple experiments using the training data to examine the effects of various shallow features and the use of GloVe features with a varying number of dimensions. All experiments in this paper were conducted using stratified five-fold cross-validation and

[3] http://nlp.stanford.edu/data/glove.6B.zip

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

the results were measured with macro F-score based on precision and recall on the class labels *favor* and *against*. Our system used supervised machine learning algorithms supplied by the scikit-learn library (Pedregosa et al., 2011).

3.1 Baseline

The first experiment was set up to gain insight in the performance of different classifiers and their parameters. We chose a basic approach using only word unigrams (bag of words approach). The best of the resulting models was chosen as the baseline, serving as an indication of the performance of a simplistic system. The models were trained on the entire data set, not divided by individual targets. We chose to perform the experiment with two different Support Vector Machines (SVM) and one Naive Bayes (MNB) classifier with different parameters[5].

One of the hyperparameters we optimized was *C*, which is a regularization term for misclassifications of each sample. Higher values will do a better job correctly labelling the training data during training (smaller hyperplane margin), but are more likely to overfit. Conversely, lower values may have more misclassifications because it will ignore more outliers (larger hyperplane margin), but are less likely to overfit. We also used the *decision function shape* parameter to decide whether to use one-vs-one (ovo) or one-vs-rest (ovr) as decision function. Ovo constructs one classifier per pair of classes. At prediction time, a vote is performed and the class which receives the most votes is selected. The ovr strategy consist of fitting one classifier for each class. The table below displays the results from the experiments.

Classifiers	Parameter specification	Macro F
Multinomial NB	[alpha=0.01]	0.5513
SVM	[kernel='linear', C=0.37]	0.5701
LinearSVM	[kernel='linear', C=0.28]	**0.5819**

Table 1: Average macro F-scores from five-fold CV experiments with different classifiers on the entire training set.

LinearSVM scored highest and established the baseline with the macro F-score of 0.5819. However, the LinearSVM classifier was not beneficial in later experiments when trained individually per target[6] and therefore only used as a performance baseline.

3.2 Improved system

In the development phase, the data set was divided by the individual targets creating five respective data sets. The development experiments began by including more and more shallow features. We started off by applying various forms of n-grams (uni-, bi- and trigram of words and characters). The classifier that achieved the highest cross-validated macro F-score from these experiments was MNB using character trigram. The achieved score was 0.6290. The experiments continued by adding features (listed in section 2.2) to the MNB in addition to the character trigram feature. Results of these experiments can be seen in table 2.

Shallow Features	Macro F	Change
Trigram characters	0.6290	
.+negation	0.6308	(+ 0.0018)
.+length of tweets	0.6311	(+ 0.0003)
.+capital words	0.6313	(+ 0.0002)
.+non-single punctuation	0.6356	(+ 0.0043)
.+exclamation mark last	0.6358	(+ 0.0002)
.+lengthening words	**0.6360**	(+ 0.0002)
.+sentiment	0.6352	(- 0.0008)
.+number of tokens	0.6264	(- 0.0088)

Table 2: Average macro F-scores for different sets of shallow features from five-fold CV experiments with MNB classifier on the entire training set.

Table 2 shows that adding shallow features yielded only a slight increase in macro F-score from 0.6290 to 0.6360. Based on this, relatively small, improvement it is difficult to imply that the addition of features gave any substantial performance boost of the system.

3.3 Final system

Subsequent experiments tested the use of a Logistic Regression classifiers with GloVe feature vectors. We used pre-trained word vectors from

[5] SVM with *kernel*=$[linear, rbf, poly]$, *C*=$numpy.logspace(-3, 3, 50)$, *decision_function_shape*=$[ovo, ovr]$ and LinearSVM with *C*=$numpy.logspace(-3, 3, 50)$. MultinomialNB with *alpha*=$numpy.logspace(-1, 1, 10)$).

[6] Average macro F-score over all targets:
LinearSVM with word bigram: 0.4955.
LinearSVM with character trigram: 0.5970.
LinearSVM with shallow features: 0.5974

Features	Overall	std (σ)	Atheism	Climate	Feminism	Hillary	Abortion
Baseline	0.5819	0.0494	-	-	-	-	-
Best shallow features	0.6360	0.0891	0.6601	0.5923	0.6246	0.6022	0.7006
Glove features	0.6067	0.0722	0.6516	0.6256	0.5553	0.5898	0.6102
Glove + best shallow	0.6048	0.0659	0.5775	0.5604/0.6754	0.6291	0.5479	0.7088
Glove + n-gram	**0.6751**	0.0704	0.7055	0.6540/0.6404	0.6537	0.6427	0.7204

Table 3: Average macro F-scores, both overall and per target, for different combinations of feature sets from five-fold CV experiments on the entire training set. Baseline model was not trained per target, therefore no individual scores are available. Where two scores are listed, there were two models used (soft/hard voting).

different corpora with a various number of dimensions (*corpus sizes* = $[(6Btokens, 400Kvocab), (27Btokens, 1.2Mvocab)]$ and *dimensions* = $[25, 50, 100, 200, 300]$). The various dimensions supposedly capture different granularities of meaning obtained from the corpora they were extracted from[7].

From table 3 we can observe that from the baseline score of 0.5819 the result increased to 0.6360 when applying the best shallow features. It also shows that using only the Logistic Regression classifier with GloVe vectors did not perform well. For this reason we decided to combine multiple classifiers. Initially we tried wrapping the Logistic Regression classifier and the MNB classifier from table 2 in a voting classifier. However, this new voting classifier did not improve the performance, instead a further drop in performance occurred. We later inspected the outcome of the combined classifiers when we reduced the feature set of the MNB classifier down to only applying versions of n-grams. This was more successful and our best result was achieved using the Logistic Regression classifier using GloVe features, MNB classifier using bigram words, and a MNB classifier with trigram characters wrapped inside a soft voting classifier. The final submission therefore included only n-gram features and the rest of the features were discarded. As seen in table 3 this scored 0.6751, which was a substantial improvement over the performance baseline.

[7]The final submission used the following word vectors: Atheism (*size*=$6B$, *dimension*=200), Climate Change (*size*=$27B$, *dimension*=200), Feminist Movement (*size*=$27B$, *dimension*=100), Hillary Clinton (*size*=$27B$, *dimension*=200), Legalization of Abortion (*size*=$27B$, *dimension*=100).

4 Results on Test Data

Our submitted approach achieved a macro F-score of **0.6247** on the test data, while the best system on task 6A achieved a score of 0.6782. After the gold labels were released, we ran the test ourselves in order to see how well we did on precision, recall, and F-score. Table 4 shows our final results. The high precision on the class *against* shows that predictions for this label were mostly correct, albeit with a relatively low recall.

Stance	Precision	Recall	F-score
Favor	0.5750	0.6053	0.5897
Against	0.8770	0.5287	0.6597
Overall macro F-score			0.6247

Table 4: Precision, recall and F-score of the official submission per class as well as overall macro F-score.

At the end of section 2.3, we mentioned that we established a threshold in our system. The threshold value was set at the last minute using a rule of thumb as we did not have time to perform experiments to determine the optimal setting, or even whether it was beneficial at all. Our intention was to use this approach only for the category *Climate Change is a Real Concern*, as this was the most skewed topic. However, by accident, it was applied to *all* topics. Comparing our best result in the development phase with the test result, we can observe a substantial drop in performance. This is a result of the threshold that was – by mistake – applied in all predictions. To measure how much this affected our system, we performed an overall test run where the threshold as used in the original submission was disregarded. This resulted in a macro F-score of 0.6660 - an increase of 0.0413 relative to our submission

score. The threshold proved to have lowered the recall for both *favor* and *against* and explains the low recall in the submitted system predictions.

Stance	Precision	Recall	F-score
Favor	0.5432	0.7237	0.6206
Against	0.8042	0.6378	0.7114
Overall macro F-score			0.6660

Table 5: Precision, recall and F-score of the submission without the applied threshold per class as well as overall macro F-score.

It is worth mentioning that even though the addition of all shallow features gave poor results during development phase, it performed a lot better on the test data, scoring 0.6939.

5 Conclusion

This paper summarizes our system created for SemEval 2016 task 6A - *Detecting Stance in Tweets*. Using shallow features alone performed well, but combining shallow features and word embeddings created from GloVe word vectors increased the score substantially.

With this system we finished 10th as we were able to detect stance in tweets with a macro F-score of 0.6247 on the test data, whereas the best system in task 6A scored 0.6782. Post-analysis revealed that the application of an ad-hoc threshold to prevent low-confidence predictions was a mistake, resulting in a 0.0413 loss in overall macro F-score. The threshold should have been set using cross-validation, or even better, not at all.

References

Charles Elkan. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. LAWRENCE ERLBAUM ASSOCIATES LTD.

Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. 2015. Twitter sentiment detection via ensemble classification using averaged confidence scores. In *Advances in Information Retrieval*, pages 741–754. Springer.

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, June.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Maria Carolina Monard and Gustavo EAPA Batista. 2002. Learning with skewed class distributions. *Advances in Logic, Artificial Intelligence, and Robotics: LAPTEC 2002*, 85:173.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition ldc2011t07. dvd. *Philadelphia: Linguistic Data Consortium*.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Foster Provost. 2000. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI2000 workshop on imbalanced data sets*, pages 1–3.

Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. 2009. Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):281–288.

Marilyn A Walker, Pranav Anand, Rob Abbott, Jean E Fox Tree, Craig Martell, and Joseph King. 2012. That is your evidence?: Classifying stance in online political debate. *Decision Support Systems*, 53(4):719–729.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schlkopf. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer.