INF-UFRGS-OPINION-MINING at SemEval-2016 Task 6: Automatic Generation of a Training Corpus for Unsupervised Identification of Stance in Tweets

Marcelo Dias and **Karin Becker**
Instituto de Informática (INF)
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, RS, Brazil
{marcelo.dias, karin.becker}@inf.ufrgs.br

Abstract

This paper describe a weakly supervised solution for detecting stance in tweets, submitted to the SemEval 2016 Stance Task. Our approach is based on the premise that stance can be exposed as positive or negative opinions, although not necessarily about the stance target itself. Our system receives as input n-grams representing opinion targets and common terms used to denote stance (e.g. hashtags), and use these features, together with the sentiment detection solutions, to automatically compose a large training corpus. Then, it applies a supervised learning algorithm to develop a stance prediction model.

1 Introduction

Sentiment analysis involves the automatic identification of opinions, feelings, evaluations, attitudes and emotions expressed by people in the written language. Popular lines of work in this field are opinion mining (Liu, 2012) and emotion mining (Mohammad, 2016). Stance detection is a less explored problem, addressed as part of SemEval-2016 (International Workshop on Semantic Evaluation 2016), Task 6[1]. Stance detection is defined in this task as the automatic determination from text whether its author is in favor of the given target, against the given target, or whether neither inference is likely.

The present work describes the solution developed for Task 6-b, which involves the unsupervised stance detection in tweets based solely in their content. The target is Donald Trump, a possible republican presidential candidate, for which two sets of non-annotated tweets were supplied: a domain corpus with 78,156 tweets and a test corpus with 707 tweets for task evaluation purpose. Another set of annotated tweets for stance detection was available as part of Task 6-a (supervised stance detection), which included 639 tweets about another possible candidate, Hillary Clinton. We used this annotated data to develop the proposed method, due to the similarity of problems. Details of the Task 6 can be found at (Mohammad et al., 2016).

The identification of stance can be complex even for humans (Walker et al., 2012), and our strategy was to address it partly as an opinion mining problem. Stance can be exposed as positive or negative opinions, but not necessarily about the target of the stance problem. For instance, when the opinion target is a politician, the target may be his/her agenda for health or education, members of the same party, or opponents. In addition, stance detection faces challenges common to sentiment analysis in general, such as use of vocabulary and slang specific of the media, orthography errors, sarcasm, etc.

We developed a weakly supervised method for detecting stance in tweets. Our method requires some side-related targets and key stance n-grams, which are used, together with sentiment detection solutions, to automatically label tweets with regard to a stance. The automatically labeled tweets are then used to train a classifier to detect stance in tweets, resulting in a stance prediction model.

The remaining of the paper describes the obtained results, the proposed solution and the experiments developed.

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

Proceedings of SemEval-2016, pages 378–383,
San Diego, California, June 16-17, 2016.

		Predicted			
		Against	**Favor**	**None**	**Recall**
Actual	**Against**	206	44	49	68.90
	Favor	94	42	12	28.38
	None	192	24	44	16.92
	Precision	41.87	38.18	41.90	
	F-score	52.09	32.56	24.11	

Table 1: Trump Corpus: Confusion Matrix and Metrics

2 Results

Task 6-b evaluated the proposed solutions according to the mean F-score of *Against* and *Favor* stances, considering the test dataset. Among 9 participants, the solution proposed by our team was the second runner-up, with a mean F-score of 42.43. Table 1 displays the confusion matrix related to the test dataset, together with the respective precision, recall and F-score metrics. The precision obtained for all classes are relatively similar, ranging from 38.18% (*Favor*) to 41.9% (*None*). However, it is clear that our solution tends to classify instances as *Against*, which explains why only the recall of this stance displays a good value (68.9%).

These results can be explained by the fact that our approach is more effective in identifying *Against* tweets when they are related to Trump or related targets, but fails to recognize endorsements to Trump when they are expressed as criticisms to his opponents. In addition, our system relies on the identification of sentiment, and the performance for identification of negative sentiment was clearly superior, compared to the detection of positive and neutral tweets, as discussed in more detail in Section 4.2. The particularly low recall obtained for the *None* class, which assumes lack of sentiment, might have influenced the other results.

3 The Process

Our system was developed on the premise that the stance towards a politician could be detected by analysing opinions of people tweeting explicitly about him, his party, his supporters or opponents, as well as subjects that he endorses or criticizes. As a weakly supervised solution, our system receives as input n-grams representing opinion targets (e.g. variations on the name of Trump, or his opponents), and terms frequently used to denote stance (e.g. hashtags, derogatory terms). Tweets expressing opinions according to these characteristics could be used to automatically compose a large training corpus for a supervised approach, such that a stance prediction model could be produced. This eliminates the burden of manual annotation.

For that purpose, we developed a process for automatically detecting stance in tweets that converts a stance detection problem into a polarity detection problem, according to the following steps:

1. Stance Features Identification;
2. Rule-based Automatic Annotation of a Training Corpus;
3. Creation of a Stance Prediction Model Using Supervised Learning;
4. Stance Prediction of Unlabeled Tweets.

3.1 Stance Features Identification

The first step in the process was to identify n-grams that typically would indicate a stance in the domain corpus. We developed a program to extract n-grams (uni-gram, bi-grams and tri-grams) from the domain corpus, and rank them by frequency. Then, we manually inspected the most frequent n-grams (top 200), selecting the ones that directly or indirectly were related to the stance target. We divided these n-grams into two categories: *side-related targets* and *stance keywords*. Side-related targets are expressions used as the target of an opinion. Variations of "Donald Trump" (e.g. "Donald", "Trump") and his party are examples of the Favor side, whereas variations of the name of his opponents (e.g. Hilary Clinton) and subjects that compose this political platform (e.g. immigrants) are instances of the Against side. Table 2 shows all the side-related targets selected for Trump.

Stance keyword n-grams consist of expressions that enable to assign a stance even when the opinion target is implicit. For example, the unigram "Apprentice", name of the TV show presented by Trump, was used in ironic tweets denoting an against stance about him, whereas the hashtag #stophillary would represent a favor stance towards Trump. Table 3 displays the stance all keyword n-grams selected for Trump.

3.2 Automatic labeling of a Training Corpus

The goal of this step is to automatically label tweets, so as to compose a training dataset to develop a

Side	Side Related Targets
FAVOR	realdonaldtrump, donaldtrump, donald trump, donald, trump, republican, republicans
AGAINST	hillaryclinton, hillary clinton, hilary clinton, hillary, hilary, clinton, clintons, hill, democrats, democrat, bill clinton, obama, mexicans, mexican, latino, latinos, elchapo, chapo, immigrant, immigrants, immigration, mexico

Table 2: Side Related Targets

Side	Key Stance N-grams
FAVOR	stop hillary, stophillary
AGAINST	love wins, lovewins, apprentice, dontvotefortrump, mr trump, racist

Table 3: Key Stance N-grams

stance prediction model using supervised learning. We devised a set of rules that represent our premise about opinions characterizing the stance, displayed in Table 4. Only clearly expressed stances are considered for the training dataset (FAVOR, AGAINST and NONE), otherwise they are disregarded.

Rules 1 and 2 assume the use of stance keywords to denote positive/negative opinions, whereas rules 3-6 consider the combination of a side-related target and the positive/negative polarity of the sentiment contained in the text. Rule 7 assumes that it is unlikely that tweets without sentiments represent a stance (NONE).

We developed a program to generate for each tweet of the corpus the following features:

- Presence of at least one favor stance keyword n-gram;
- Presence of at least one against stance keyword n-gram;
- Presence of at least one favor-related target n-gram;
- Presence of at least one against-related target n-gram;
- Tweet polarity;

The program scans all tweets in the domain corpus tweets, and generates a new dataset with these features. It verifies the presence of key stance n-grams and side related targets, and evaluates tweet polarity by submitting the tweet text to three sentiment analysis APIs. Once the stance features are generated for each tweet, the rules are applied. Tweets are included in the training corpus if rules 1, 2, 3, 4, 5, 6 or 7 hold, or discarded otherwise. The resulting training dataset is composed by the original tweet texts, and the label automatically assigned to the tweet.

Rule	Stance
1 - KEY-FAVOR Presence of a favor keyword n-gram with no against keyword n-gram	FAVOR
2 - KEY-AGAINST Presence of an against keyword n-gram with no favor keyword n-gram	AGAINST
3 - FAVOR-POSITIVE Presence of a favor-related side target with no against-related side target and positive tweet polarity	FAVOR
4 - FAVOR-NEGATIVE Presence of a favor-related side target with no against-related side target and negative tweet polarity	AGAINST
5 - AGAINST-POSITIVE Presence of a against side related target with no favor-related side target and positive tweet polarity	AGAINST
6 - AGAINST-NEGATIVE Presence of an against-related side target with no favor-related side target and negative tweet polarity	FAVOR
7 - NEUTRAL Neutral tweet polarity	NONE
Other cases	DISCARD TWEET

Table 4: Rules used for automatic labeling

With the goal of increasing the precision of tweet polarity identification, we combined the results of three off-the-shelf sentiment analysis APIs, namely HP Haven On Demand[2], IBM Alchemy[3] and Vivekn[4]. Each API returns the polarity property as a label (i.e. negative, positive or neutral) and the respective score property. The developed program first verifies if Haven's score and Alchemy's score are equal to zero, a condition that defines a tweet as neutral. Otherwise, the program combines the APIs by adding the three scores. The result is a negative tweet if the calculated value is negative, and positive otherwise. This particular combination was based on experiments on the use of these APIs, which are described in more details in Section 4.2.

Table 5 displays the distribution of tweets per rule considering both Trump and Hillary datasets. With regard to Trump corpus, although there were 78,156 tweets in the domain corpus supplied for the task,

[2]https://www.havenondemand.com/

[3]http://www.alchemyapi.com/

[4]http://sentiment.vivekn.com/docs/api/

Rule	Hillary	Trump
1 - KEY-FAVOR	10	1
2 - KEY-AGAINST	96	1,383
3 - FAVOR-POSITIVE	63	2,295
4 - FAVOR-NEGATIVE	127	7,369
5 - AGAINST-POSITIVE	6	0
6 - AGAINST-NEGATIVE	25	4
7 - NEUTRAL	41	2,201
Other cases	271	6,709
Total	**639**	**19,952**

Table 5: Domain Corpus Tweets X Rule

it was possible to apply the rules only to 19,952 tweets due to quota restrictions for the free usage of Alchemy. Most tweet labeled as *Against* were detected using Rules 4 and 2, whereas most *Favor* labels were assigned due to Rule 3. It is possible to see that most tweets in Trump's corpus were focused on Donald Trump himself and target representing support to him, and therefore, the bias related to opinions related to opposition targets, as represented by rules 5 and 6, could not be explored. The training sets resulting from application of the rules were unbalanced, with a dominance of *Against* instances. Hillary training set was composed of 229 *Against* (62%), 98 *Favor* (27%) and 41 *None* instances (11%). Trump training set was composed of 8,752 *Against* (66%), 2,300 *Favor* (17%) and 2,201 *None* instances (17%).

3.3 Creation of a Stance Prediction Model Using Supervised Learning

The goal of this step is to develop a stance predictive model by submitting the training dataset automatically labeled to a classification algorithm.

The training corpus generated in previous step is composed of the original tweet text and the stance label. We pre-processed the texts and submitted them to a classification algorithm for a three class problem: Favor, Against and None. We adopted the Weka platform (version 3.7.11) (Hall et al., 2009), and an SVM classification algorithm (SMO) available in this environment with the default parameters.

The following actions were performed during preprocessing: a) convert all tweet texts to lower case; b) replace all mentions for tweet profiles by the `"a_mention"` unigram, except those containing the text of a side-related target (e.g. "realdonaldtrump"); c) submit the text to the *Stringtowordvector* function available in Weka for textual feature extraction and creation of the training dataset. We chose the following parameters for feature extraction: a) extraction of alphabetic unigrams and bigrams; b) removal of stopwords; c) no limitation on the number of n-grams extracted; and d) binary representation of features (i.e. presence or absence).

		Predicted			
		Against	**Favor**	**None**	**Recall**
Actual	**Against**	290	46	25	80.33
	Favor	45	55	12	49.11
	None	113	23	30	18.07
	Precision	64.73	44.35	44.78	
	F-score	71.69	46.61	25.75	

Table 6: Hillary Corpus: Confusion Matrix and Metrics

3.4 Stance Prediction of Unlabeled Tweets

The last step of the process is to predict the stance for any tweet, using the model trained in the previous step. For the Semeval task, we applied the predictive model to the provided test set, composed of 707 instances, obtaining the results displayed in Table 1.

As an experiment, we also tested the approach on the Hillary dataset provided in Task 6-a, but limited to the tweets that were discarded during the creation of the training set. The results are displayed in Table 6, which are significantly better, compared to Trump's. In both results, it is clear that performance for the *None* class is the weakest point of our solution. Very few neutral tweets are recognized as such (recall of 16.92% and 18.07% for the Trump and Hillary datasets, respectively), compromising the precision of both *Favor* and *Against* stances.

4 Experiments

We made two main experiments as the basis for the proposed solution: a) verification of the impact of proposed rules with regard to the performance of the predictive model, using the Hillary labeled dataset, and b) verification of the precision of the off-the-shelf sentiment detection APIs used. These are described in the remaining of this section.

Rules	Class	Automatic labeling			Predictive Model			Combined		
		Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
(a) 1.2.7	Against	85.41	79.61	82.41	52.45	95.35	67.67	58.05	90.86	70.84
	Favor	40.00	33.33	36.36	0.00	0.00	0.00	40.00	3.57	6.55
	None	36.59	46.88	41.10	21.74	3.73	6.37	31.25	12.05	17.39
	Weighted Avg.	71.08	68.71	69.66	33.43	51.02	37.22	47.92	55.09	45.69
(b) 3.4.5.6.7	Against	78.20	66.24	71.72	58.30	70.59	63.86	65.26	68.70	66.94
	Favor	48.86	65.15	55.84	17.39	26.09	20.87	35.03	49.11	40.89
	None	36.59	38.46	37.50	50.82	24.41	32.98	45.10	27.71	**34.33**
	Weighted Avg.	64.62	61.83	62.63	50.79	49.60	48.21	54.72	54.62	53.90
(c) All rules	Against	81.22	77.18	79.15	47.49	86.67	61.36	64.73	80.33	**71.69**
	Favor	47.96	66.20	55.62	30.77	19.51	23.88	44.35	49.11	**46.61**
	None	36.59	26.79	30.93	57.69	13.64	22.06	44.78	18.07	25.75
	Weighted Avg.	68.01	67.39	67.27	49.10	46.87	39.74	55.98	58.68	**55.36**

Table 7: Experiments with Hillary dataset

4.1 Experiments on the Rules to Detect Stance

We experimentally developed our method using the Hillary labeled corpus provided for Task 6-a, given that both stance targets are politicians in campaign. To perform this test and assess the performance of our method, we automatically labeled the Hillary training corpus, and applied the predictive model only to the instances that were not filtered by any of the rules. Table 7 displays the results obtained according to the rules used to prepare the training dataset. We compared three different scenarios:

(a) Stance Keywords rules (rules 1 and 2) combined with the neutral rule (rule 7);
(b) Side-related target and polarity rules (3 to 7 rules);
(c) All rules;

We measured the precision, recall and F-measure of the instances that were included in the training dataset (columns *Automatic labeling*), predicted instances according to the trained model (columns *Predictive Model*), and the whole set of instances labeled either using the rules or the predictive model (columns *Combined*). Scores are detailed per class and weighted average.

With regard to automatic labeling, the best weighted F-score was obtained by the set of stance keyword rules (a), i.e. 69.66, despite the poor result for the *Favor* class. However, our objective was to improve detection of *Against* and *Favor* stances, even at the expense of a less favorable result for the *None* class. Thus the set of all rules (c), which presents the second better F-score, is more interesting because it yields good scores for both *Favor* and *Against* stances.

Considering the predictive model, even if the training set is less accurate using only the rules involving sentiment about a side-related target (b), the respective model yields more accurate predictions (i.e. weighted F-measure of 48.21%). However, considering the *Against* and *Favor* classes, again the set of all rules (c) produced a slightly better result.

Finally, when considering the combination of the instances labeled by the rules and the ones by the predictive model, the best results are displayed for the set of all rules (c), considering both the weighted F-measure, and the scores for the *Against* and *Favor* classes. Thus, the set of all rules presents more balanced results throughout the 2 phases of the process, yielding the best final result.

It should be noticed that the results summarized in Tables 1 and 6 for Trump and Hillary corpora, respectively, were produced differently. The test set provided for Task-b was labeled using the predictive model only, whereas the Hillary results were produced using the combined approach of rules and predictive model. Considering that Trump corpus was much bigger, compared to Hillary‘s, we assumed for the task that the results of the predictive model would be more accurate. By applying the combined approach over Trump test set instead, the result would be slightly better (43.8%, using the same evaluation criterion adopted for the task).

4.2 The Sentiment Analysis APIs

We started the development of our solution using only Haven On Demand API but, after some tests,

we noticed issues on the precision of the polarity detection step, which is key in our process. Indeed, it can be seen in Table 5 that an elevated number of tweets is filtered by sentiment rules.

API	Class	Precision	Recall	F-Score
Alchemy	Negative	64.39	78.42	70.71
	Neutral	17.82	26.34	**21.26**
	Positive	39.23	15.89	22.62
Haven	Negative	46.42	51.28	48.73
	Neutral	16.84	17.20	17.02
	Positive	55.97	46.73	50.93
Vivekn	Negative	57.99	56.62	57.30
	Neutral	13.39	17.20	15.06
	Positive	36.20	31.46	33.67
Combination	Negative	64.84	78.42	**70.99**
	Neutral	31.46	15.05	20.36
	Positive	61.56	61.37	**61.47**

Table 8: APIs Comparison

Using a polarity annotated dataset also in the political domain[5] (Mohammad, 2016), we compared the performance of the 3 chosen APIs, and their combination. Results are displayed in Table 8. It can be seen that the best results for negative and neutral texts are yielded by Alchemy API, whereas Haven on Demand provides a better result for positive tweets. The major benefit obtained by the combination of the three APIs was a significant better performance with regard positive texts, given that the results for negative and neutral texts are quite similar to the use of Alchemy alone. These results reveal that all solutions perform better in the detection of negative sentiment.

The performance in sentiment identification seems to have a straightforward relationship with the results obtained in stance identification. According to Table 5, the rules that combine sentiment with Favor-related side are the most representative ones, not to mention that the only rule for identifying lack of stance is related to neutral sentiment. It is interesting to note that very similar figures can be found when we compare the F-measure of the *Combination* of solutions in Table 8, and the *Automatic labeling* scores obtained for the sentiment rules (b) in Table 7, particularly Negative/Against (70.99% and 71.72%), and Positive/Favor (61.47% and 55.84%) F-measure scores.

[5]http://www.purl.org/net/PoliticalTweets2012

5 Conclusions and Future Work

The results obtained by the participants of SemEval Task 6 reveal that stance identification is a hard problem, for which available solutions still need to evolve. For the non-supervised task, the best result achieved the mean F-score of 56.28%, which still is not a good performance in itself. The publication of the gold standard for the task, together with the labeled datasets available for Task 6-a, will allow us to improve the process, focusing mainly in strategies for increasing the performance with regard to the Favor and None stances. Among the strategies are the improvement of polarity detection for the positive and neutral classes, increasing the automatic generated training corpus by overcoming the dependency on Alchemy (subject to quota limitations), and the investigation of a revised set of rules. Another approach would be to label instances based on social information like Twitter profiles, and the exploration of conversation threads and the connections among profiles available in the Twitter platform. We also intend to test our process using other domain corpora to explore whether the approach adopted and its underlying premise can be generalized to stance identification on other subjects.

References

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

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.

Saif M. Mohammad. 2016. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In Herb Meiselman, editor, *Emotion Measurement*. Elsevier.

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.