UninaStudents @ SardiStance: Stance Detection in Italian Tweets - Task A

Maurizio Moraca, Gianluca Sabella, Simone Morra
Università degli Studi di Napoli Federico II
[mau.moraca,gia.sabella,simone.morra2]@studenti.unina.it

Abstract

English. This document describes a classification system for the SardiStance task at EVALITA 2020. The task consists in classifying the stance of the author of a series of tweets towards a specific discussion topic. The resulting system was specifically developed by the authors as final project for the Natural Language Processing class of the Master in Computer Science at University of Naples Federico II. The proposed system is based on an SVM classifier with a radial basis function as kernel making use of features like 2 chargrams, unigram hashtag and Afinn weight computed on automatic translated tweets. The results are promising in that the system performances are on average higher than that of the baseline proposed by the task organizers.

Italiano. *Questo documento descrive un sistema di classificazione per il task SardiStance di EVALITA 2020. Il task consiste nel classificare la posizione dell'autore di una serie di tweets nei confronti di uno specifico topic di discussione. Il sistema risultante è stato specificamente sviluppato dagli autori come progetto finale per il corso di Elaborazione del Linguaggio Naturale nell'ambito del corso di laurea magistrale in Informatica presso l'università degli studi di Napoli Federico II. Il sistema qui proposto si basa su un classificatore SVM con una funzione radiale di base come kernel facendo uso di features come 2 char-grams, unigram hashtag e l'Afinn weight calcolato sui tweet tradotti in automatico. I risultati sono promettenti in quanto le performance sono in media superiori rispetto a quelle della baseline proposta dagli organizzatori del task.*

1 Introduction

This work reports on the application of our system for the resolution of the EVALITA 2020's SardiStance task (Basile et al., 2020; Cignarella et al., 2020). Stance detection is a classification task aiming at determining the position (stance) of the author of a given text concerning the topic (target) treated in the text itself. In other words, the challenge deals with automatically guessing if the author of the text is in favour, against or is in a neutral position towards the topic subject of a given post. The utility of such an automatic system can be found in political analysis, marketing and opinion mining. Automatic determination of Stance is a new approach to opinion mining paradigm which finds better application in social and political applications. It is quite different form in which sentiment analysis in many views, but the main difference is the drastic reduction to a three class decision system (in favour, against, neutral) given its main fields of application. The challenge poses many challenges, as the real target might not be expressly cited in the text or could bear a not so clear expression of the author's opinion like in the following example (Lai et al., 2020):

Target: Donald Trump
Tweet: Jeb Bush is the only sane candidate in this republican lineup.

Although one could erroneously think that

this task is similar to sentiment analysis, the following example illustrates how, in some cases, stance detection results are opposed to those reached by sentiment analysis (Lai et al., 2020):

Target: Climate change is a real concern
Tweet: @RegimeChangeBC @ndnstyl It's sad to be the last generation that could change but does nothing. #Auspol
This tweet presents a negative polarity, although the author claims to be in favour of the target. Classification systems for stance detection, then, attempt the individuation of the author position on the target taking into account of features obtained by the text that are almost similar to those used in hate speech detection, irony detection, mood detection, but with some further effort devoted to the specificity of the task.

SardiStance is the first Italian Initiative focused on the automatic classification of stance in tweets. It includes two different tasks: A) Stance Detection at a textual level, where tasl participants are asked to resolve the guess basing only on the tweet textual content, and B) Stance Detection with the addition of contextual information about the tweet, such as the number of retweets, the number of favours or the date of posting; contextual information about the author, location, user's biography); we proposed runs only for task A). As required by the task proposal, task A requires a three-class classification process where the system has to predict whether the items in the set are in FAVOUR, AGAINST or NEUTRAL exploiting the text of the tweet.

2 Description of the System

The system is based on a SVM classifier with a radial basis function (rbf) kernel. Most of the features selected were inspired by (Lai et al., 2020) and correspond to the following ones:

- n-grams, bag of n consecutive words in binary representation (presence/absence) where n corresponds to 1, 2 or 3.
- char-grams, bag of n consecutive characters in binary representation (presence/absence) where n corresponds to 2, 3, 4 or 5.
- unigram hashtag, bag of hashtags in binary representation (presence/absence).
- unigram emoji, bag of emojis in binary representation (presence/absence)
- unigram mentions, bag of mentions in binary representation (presence/absence).
- num uppercase words, number of uppercase words in a tweet.
- punctuation marks, frequency of each punctuation mark (. , ; ! ?) and their total frequency.
- Afinn weight[1] (Nielsen, 2011), based on a sentiment analysis lexicon made up of 3500 English words manually annotated with a polarity value within the range [-5, +5]. The value of this feature is computed for each tweet as the sum of the polarities associated to the words constituting the tweet translated to English via Google Translate.
- Hu&Liu weight[2], based on a sentiment analysis lexicon composed of two separated lists of English words, where the first one contains 2,006 words with a positive connotation, and the second one contains 4,783 words with a negative connotation. In this work, a value of +1 is given to words which overlap with the positive ones in the lexicon and a value of -1 to the ones overlapping with the negative list. The total polarity of each tweet is computed as the sum of the weights given to the words in a tweet.
- NRC vector[3] (Bravo-Marquez et al., 2019), based on a lexicon consisting in a list of English words, each of which is associated to the most representative emotion. The emotion which are comprised are anger, fear, expectancy, trust, surprise, sadness, joy, and disgust. Furthermore, to each sample, a score indicating the emotion intensity is also associated. This score has a value within the range [0, 1].
- DPL vector[4] (Castellucci et al., 2016), based on a lexicon of 75,021 pairs of

[1] https://github.com/fnielsen/afinn/tree/master/afinn/data
[2] https://github.com/woodrad/Twitter-Sentiment-Mining/tree/master/Hu%20and%20Liu%20Sentiment%20Lexicon
[3] http://saifmohammad.com/WebPages/AffectIntensity.htm
[4] http://sag.art.uniroma2.it/demo-software/distributional-polarity-lexicon/

lemma::pos_tag associated to scores indicating the level of positivity, negativity, and neutrality of the lemma, as it follows

(1) buono::a 0.76691014 0.12262548 0.11046442

For each tweet of the dataset, each word was lemmatised and, for each resulting lemma, a morpho-syntactic category was associated. For this kind of analysis LinguA (Dell'Orletta, 2009; Attardi and Dell'Orletta, 2009; Attardi et al., 2009) was used. The DPL vector feature consists of a triplet of scores representing positivity, negativity, and neutrality levels in the tweet. To obtain this value, the scores of each pair lemma::pos_tag in a tweet were summed.

In order to select the best features combination, a wrapper-based feature selection algorithm was used to test all the possible features combinations. The best one resulting from the collected performance on the validation set was chosen, that is the one combining 2 char-grams, unigram hashtag and Afinn weight. The evaluation metrics are discussed in the next section (Section 3). Since a SVM classifier with an RBF kernel was used, it was important to tune the C and γ parameters.

To set the complexity of a generic SVM model, C is used: this parameter controls the acceptable distance of the decision boundary in the n-dimensional features space from the support vectors. A higher C complexity value increases the model's complexity, thus reducing the acceptable distance but also increasing the risk of overfitting; a lower C value leads to more general models that may have reduced discrimination capability. The γ parameter is specific for the RBF kernel. This parameter controls the influence single points have in the features space and controls the *smoothness* of the model, with lower values of γ leading to smoother models and vice-versa. SVMs are very sensitive to parameters tuning so specific optimisation strategies must be adopted. In this case, a grid search was performed using the following ranges of values:

- C [0.1, 0.2, . . . , 1.0, 10, 100, 1000]
- Gamma [0.001, 0.0009, 0.0008, . . . , 0.0001]

The best settings obtained on the validation set data correspond to $C = 10$ e $\gamma = 0.001$.

3 Results

In this section the performances of our system obtained during the test phase on the validation and test set are described. The validation set was obtained extracting a sample of tweets from the training set via the Stratified Sampling algorithm selecting the 20% of the training set. The evaluation metrics used are the mean value of the F1 score for the classes Against and Favour, Precision, Recall and F1 score for each class, and Accuracy. In table 3, the results obtained from the validation set are shown. From these results, the mean F1 score is obtained, corresponding to 0.5200. In table 3, the results obtained from the test set are presented.

	Precision	**Recall**	**F1 Score**
Against	0.5500	0.8300	0.6600
Favor	0.4400	0.3200	0.3100
None	0.3800	0.1300	0.0900

Table 1: Validation Set Performance

	Precision	**Recall**	**F1 Score**
Against	0.7300	0.8491	0.7850
Favor	0.4348	0.3571	0.3922
None	0.3488	0.1744	0.2326

Table 2: Test Set Performance

Team	**F1-score**		
	Against	**Favour**	**None**
UNITOR_1	0.7866	**0.5840**	0.3910
UNITOR_2	0.7881	0.5721	**0.3979**
UNITOR_3	**0.7939**	0.5647	0.3672
UNITOR_4	0.7689	0.5522	0.3702
UninaStudents	**0.7850**	0.3922	0.2326
Baseline	0.7158	0.4409	0.2764

Table 3: Results compared with the baseline and the winning system

In table 3, on the other hand, the results are compared with the baseline proposed by the task organizers and the winning systems whose runs were submitted by the UNITOR team (Giorgioni et al., 2020) for task A. Specifically, the

baseline used a SVM classifier based on token uni-gram features, whereas UNITOR used UmBERTo[5], adding sentiment, hate and irony tags to the dataset sentences and using additional data to train their systems. As it may be noted, the *against* class result for our system is higher than the baseline and not so different from the first two runs of UNITOR. Further investigations are, conversely, needed as far as the other two classes are concerned.

4 Discussion

Our results are conditioned by the use of a training set originally in English and translated into Italian for our purposes, and, in particular, for the derivation of the Afinn weight features. As expected, the translation, made via Google translate is, in some cases poor and approximate, and can give rise to a significant level of ambiguity, however we decided to afford this risk, translating directly the tweets, instead of the lexicon, as we thought that in this last case the ambiguity could have been even greater, we just hoped that automatic translation is by far more uncertain because of polysemy, lack of flexive morphological information, and similar problems, as automatic translation skills are trained to solve at least at a first level of aproximation. In this view the use of an imperfect translation, however, is able to capture part of the semantic context in the texts, allowing us not to recur to lemmatization and further processes on the lexicon before translation. We choose to use a classic approach based on an SVM classifier in order to make our results explainable, given the scholar context in which this experience is grown. This possibility would have been impossible if we had used Deep Neural Networks, whose processes are not "readable" from an external point of view. Furthermore, the size of the data-set distributed for this challenge does not consent an affordable training with these systems. In this view, a comparison of results obtained in other stance detection challenges, similar to that proposed here in Evalita (Mohammad et al., 2016; Taulé et al., 2017; Lai et al., 2017), give strength to our choice concerning the use of SVM that often outperform DNNs. As Master students, we approached these NLP topics for the first time. Therefore, we are aware that our results are not at the state of the art in the field. However, a comparison with average performances in similar tasks for languages different from English indicates performances that are not significantly different.

[5]https://huggingface.co/Musixmatch/umberto-commoncrawl-cased-v1

Acknowledgements

We thank our teachers Francesco Cutugno and Maria Di Maro for letting us approach with NLP and EVALITA 2020 (Basile et al., 2020) and for supporting us in our work. We also thank them for giving us the opportunity to take part to the competition and for encouraging us to do our best.

References

Attardi, G. and Dell'Orletta, F. (2009). Reverse revision and linear tree combination for dependency parsing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 261–264.

Attardi, G., Dell'Orletta, F., Simi, M., and Turian, J. (2009). Accurate dependency parsing with a stacked multilayer perceptron. *Proceedings of EVALITA*, 9:1–8.

Basile, V., Croce, D., Di Maro, M., and Passaro, L. C. (2020). Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In Basile, V., Croce, D., Di Maro, M., and Passaro, L. C., editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Bravo-Marquez, F., Frank, E., Pfahringer, B., and Mohammad, S. M. (2019). Affectivetweets: a weka package for analyzing affect in tweets. *Journal of Machine Learning Research*, 20(92):1–6.

Castellucci, G., Croce, D., and Basili, R. (2016). A language independent method for generating large scale polarity lexicons. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 38–45.

Cignarella, A. T., Lai, M., Bosco, C., Patti, V., and Rosso, P. (2020). SardiStance@EVALITA2020: Overview of the Task on Stance Detection in Italian Tweets. In Basile, V., Croce, D., Di Maro, M., and Passaro, L. C., editors, *Proceedings of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2020)*. CEUR-WS.org.

Dell'Orletta, F. (2009). Ensemble system for part-of-speech tagging. *Proceedings of EVALITA*, 9:1–8.

Giorgioni, S., Politi, M., Salman, S., Croce, D., and Basili, R. (2020). UNITOR@Sardistance2020: Combining Transformer-based architectures and Transfer Learning for robust Stance Detection. In Basile, V., Croce, D., Di Maro, M., and Passaro, L. C., editors, *Proceedings of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2020)*. CEUR-WS.org.

Lai, M., Cignarella, A. T., Farías, D. I. H., Bosco, C., Patti, V., and Rosso, P. (2020). Multilingual stance detection in social media political debates. *Computer Speech & Language*, page 101075.

Lai, M., Cignarella, Alessandra Teresa, H. F. D. I., et al. (2017). itacos at ibereval2017: Detecting stance in catalan and spanish tweets. In *IberEval 2017*, volume 1881, pages 185–192. CEUR-WS. org.

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. (2016). Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Nielsen, F. Å. (2011). A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In Rowe, M., Stankovic, M., Dadzie, A.-S., and Hardey, M., editors, *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, volume 718 of *CEUR Workshop Proceedings*, pages 93–98.

Taulé, M., Martí, M. A., Rangel, F. M., Rosso, P., Bosco, C., Patti, V., et al. (2017). Overview of the task on stance and gender detection in tweets on catalan independence at ibereval 2017. In *2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2017*, volume 1881, pages 157–177. CEUR-WS.