TextWiller @ SardiStance, HaSpeede2: Text or Con-text? A Smart Use of Social Network Data in Predicting Polarization

Federico Ferraccioli[a], **Andrea Sciandra**[b], **Mattia Da Pont**[c], **Paolo Girardi**[a], **Dario Solari**[d], **Domenico Madonna**[a], **Livio Finos**[a]

a. Università degli Studi di Padova

b. Università degli Studi di Modena e Reggio Emilia

c. WMRI

d. BeeViva

ferraccioli@stat.unipd.it, andrea.sciandra@unimore.it, mattia.dapont@wmr.it, paolo.girardi@unipd.it, dario.solari@gmail.com, domenico.madonna@studenti.unipd.it, livio.finos@unipd.it

Abstract

In this contribution we describe the system (i.e. a statistical model) used to participate in Evalita conference 2020, SardiStance (Tasks A and B) and Haspeede2 (Tasks A and B). We first developed a classifier by extracting features from the texts and the social network of users. Then, we fit the data through an extreme gradient boosting, with cross-validation tuning of the hyper-parameters. A key factor for a good performance in SardiStance Task B was the features extraction by using Multidimensional Scaling of the distance matrix (minimum path, undirected graph) applied on each network. The second system exploits the same features above, but it trains and performs predictions in two-steps. The performances proved to be lower than those of the single-step model.

1 Introduction

In this paper we describe and show the results of the approach we developed to participate in the SardiStance task (Cignarella et al., 2020) for the polarity detection (i.e. Task A and B, both with constrained data) within the EVALITA campaign (Basile et al., 2020). The goal of this task was a Stance Detection in Italian tweets about the Sardines movement. The Task A is a three-class classification task where the system has to predict whether a tweet is in *Favour*, *Against* or *Neutral/none* towards the given target, exploiting only textual information, i.e. the text of the tweet. The Task B is the same as the first one, except a wider range of contextual information are available, that is: the number of retweets, the number of favours, the type of posting source (e.g. iOS or Android), and date of posting. Furthermore, the networks of the users based on Friends, Quote, Reply and Retweet were provided. We developed two systems (i.e. models) extracting features from the text (both for Task A and B) and from the social network of the users (only for Task B) and then exploited extreme gradient boosting (Chen et al., 2020) to train the model on the data. A cross-validation hyper-parameter tuning was used to define the optimal set of parameters.

We use a very similar strategy for HaSpeede2 (Sanguinetti et al., 2020) where the goal is the prediction of Hate Speech (i.e. Task A) and Stereotype (i.e. Task B). In this case, however, the sample contains documents from three different topics. We believe that these may be characterized by different vocabularies and kind of speech. We take this in account in the prediction model as explained in 3.3.

2 Features extraction and E.D.A.

2.1 Text-based Features extraction

The text preprocessing was done in R (R Core Team, 2019) software with the package TextWiller (Solari et al., 2019) (function *normalizzaTesti* with default parameters). We describe the preocess used to define the features for both for SardiStance and HaSpeede2.

The first set of features is defined by the

columns of the DocumentTermMatrix which is a matrix having documents on the rows and a column for each term. The cells contain the number of given words in the document. We defined the matrix on the basis of the normalized texts and removing terms (i.e. columns) with a sparsity larger than .9. These procedures generated a 317 terms vocabulary for SardiStance and 170 terms for HaSpeede2.

In Figure 1 we plot the term frequencies of the "In favour" and "Against" stances. The terms close to the bisector are the ones with a similar frequency in the two classes (such as "caro", "alto", "acqua"), so probably these terms don't carry much useful information to our cause. More often we found interesting terms far from the bisector, like "bolognanonsilega", "antifascismo", "abuso" or "branco" and we expected these terms to carry more weight in the classification model.

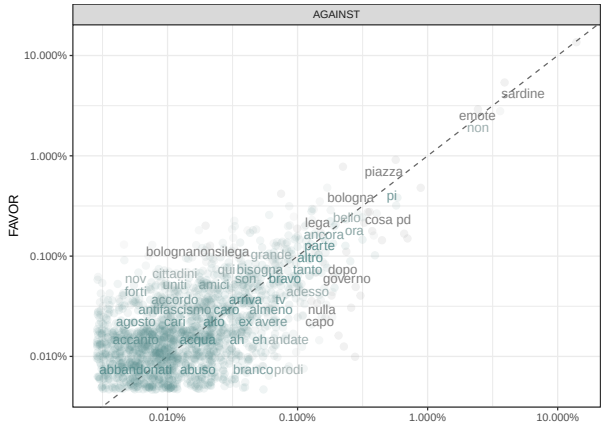

Figure 1: Scatterplot of "Favour" and "Against" term frequencies.

Further text features considered were: the number of characters and the number of words, the counts of "?" and "!" for each document. Moreover, a sentiment value was computed for each document by *sentiment* function of the R package TextWiller (Solari et al., 2019).

Figure 2 shows the association between True Stances and Sentiment. This variable will be used as a feature in Task A and B models.

Previous analyses, such as sentiment attribution through a lexicon, refer to a bag-of-words (BoW) approach. One of the most notable disadvantages of BoW is that it generally fails to capture words semantics by ignoring words order. A common solution to this problem involves the use of Word Embedding (WE). WE techniques are

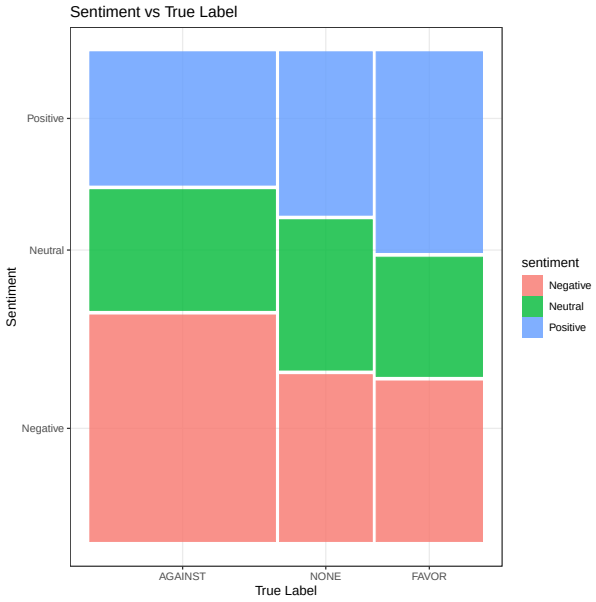

Figure 2: The Mosaic plot of True stances and Sentiment shows a clear association between the two variables.

based on neural networks and generate dense vectors for word representation, by defining a context window, i.e. a string of words before and after a focal word, that will be used to train a word embedding model. In WE, words are represented as coordinates on a latent multidimensional space derived from an underlying deep learning model that considers the contiguous words. So, for both tasks we also used a WE technique to produce context-based features. In particular, we used the *word2vec* model (Mikolov et al., 2013), a widely used natural language processing technique to extract word associations from a large corpus of text. *word2vec* is a neural network prediction model containing continuous bag-of-words (CBoW) model and Skip-gram (SG) model. The CBoW model predicts a target word from its context words, while the SG model predicts the context words given a target word. Since WE needs a huge corpus of textual data for training and given the limited amount of tweets, we augmented the data with the corpus PAISÀ (Lyding et al., 2013), a large collection of Italian web texts. We trained the model with embedded dimension set to 50 and a 5 words context window. The results for each word are then combined via averaging to obtain the final features.

2.2 Network-based Features extraction

A key point to explain the good performance in the SardiStance Task B (i.e. second best score, F-avg = 0.7309) is the efficient extraction of features from the four Networks available, that is: Friends, Retweet, Reply, and Quote. For each network, a distance matrix among subjects was computed. The distance used is the shortest path, forcing the graph to be undirected. The Distance Matrix was then projected into a euclidean space trough a Multidimensional Scaling (MDS). Since we expected the users to be strongly polarized in clusters within the network, we also expected the largest dimension to discriminate among the stances. Therefore, we retained the first and second dimension for each of the four networks. This expectation was confirmed by Exploratory Data Analysis. As an example, in Figure 3 we show the scatter plot of the first two dimensions for the Friend Network. The First Dimension clearly discriminates the three stances (in particular *Favour* vs *Against*).

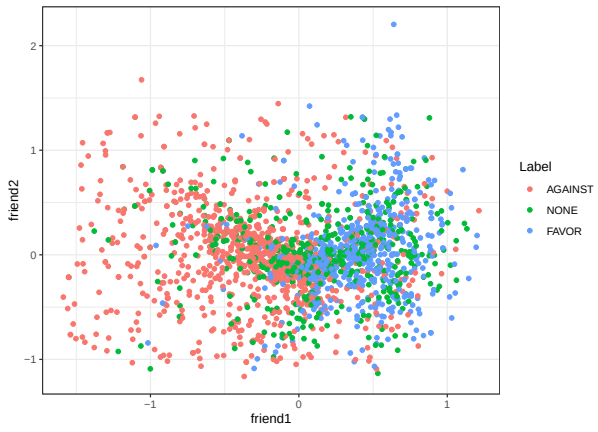

Figure 3: Scatter plot of the First and Second dimension extracted by the MDS from the distance matrix of the Friend Network (minimum path distance). There is a clear separation between between the stances *Favour* and *Against* along the first axis.

3 Developed Systems

Due to the – relatively – small sample size of the train set (composed from 2,132 tweets in Italian, the BenderRule), we decided not to use any neural network. Instead, we preferred a Gradient Boost approach (Friedman, 1999). Since this method has been developed within the statistical learning community, we used the word "model" as a synonymous of "system". We adopted the R implementation of the XGBoost (eXtreme Gradient Boosting) (Chen et al., 2020). A cross-validation parameter tuning was used to define the optimal set of parameters.

3.1 System One

As features for Task A, we used information taken from the text, that is, words/emoticons, special characters, scores of word embedding (50 dimensions), sentiment, length of the message and number of words.

For Task B we used the same features used for Task A together with the first and the second dimension extracted from the MDS computed for each network (as explained in 2.2).

3.2 System Two

Since System Two uses the same features of System One for Task A and B, the focus here is on the employed metric: the average between $F1_{Against}$ and $F1_{Favour}$. With the aim to cast the model into the metric, we fitted two separated models (i.e. one for *Favour* and one for *Against*) in the first step and then we combine the two predictions in a second step. To be more precise, the two models used in the first step predict if a document is in Favour or not (first model) and if is Against or not (second model). The two prediction are combined in a final score by a simple subtraction: *(Predicted1==Favour) - (Predicted2==Against)* which makes a -1,0,1 final score.

3.3 System for HaSpeeDe2

The corpus of documents for HaSpeeDe2 is a sample of tweets from three different topics, namely Immigrants, Muslims and Roma communities. Since the vocabulary may change among topic, we want our models to account for this specificity. We leverage on this with models that use the estimated topic. The topic is estimated by a xgboost model (trained by cross-validation). Table 1 and Table 2 report the confusion matrix and performances indices of the trained model (cross-validated).

	Reference		
Prediction	Immigrants	Rom	Terrorism
Immigrants	408	24	55
Rom	24	780	16
Terrorism	41	8	192

Table 1: Confusion matrix for the xgboost model.

Index	Immigrants	Rom	Terrorism
Sensitivity	0.86	0.96	0.73
Specificity	0.93	0.95	0.96
F1	0.85	0.96	0.76

Table 2: Sensitivity, Specificity and F1 for each topic for the xgboost model.

System One is based on an xgboost with binomial response (for both tasks). The fitting is done separately, after splitting of the sample based on the topic classification provided by the model described above in this subsection. The model is trained with the same cross-validated strategy used to train System One for the SardiStance Task.

System Two is based on an xgboost with binomial response (for both tasks). The estimate is computed on the whole sample (i.e. without splitting of System One), but the topic classification is used as feature.

For both systems the basic set of features are the same used in the SardiStance - Task A.

4 Results and discussion

4.1 Results for HaSpeeDe2

The results of the two systems are disappointing. The final ranks are always at the very bottom of the rankings. This may be partially due to a sub-optimal parameters optimization (we discovered a mistake in the parameter setting), but this is certainly not the only reason. We will take this result as an opportunity to revise the approach.

4.2 Results for SardiStance

System Two performed poorly in the final score for both Tasks. Our intuition was that the benefit of a separate optimization of $F_{Against}$ and F_{Favour} was overcome by the gain in doing a joint training (i.e. System One). We will address further efforts to better understand this result.

The results for System One are given in Table 3 for Task A and Table 4 for Task B, respectively.

The rank of System One in Task A is 13, that is just below the benchmark. The System was weak in the correct estimation of Against stance ($F1_{Against} = 0.776$), while it estimated fairly well Favour stance ($F1_{Favour} = 0.3791$).

The best performance of System One is on Task B ($F1_{Against} = 0.8505$, $F1_{Favour} = 0.6114$) where it scored 2^{nd} position.

	Reference		
Prediction	AGAINST	NONE	FAVOUR
AGAINST	613	118	108
NONE	32	22	12
FAVOUR	97	32	76

Table 3: Confusion Matrix for Task A (System One). $F1_{Against} = 0.776$, $F1_{Favour} = 0.3791$, Final: $(F1_{Against} + F1_{Favour})/2 = 0.5773$

	Reference		
Prediction	AGAINST	NONE	FAVOUR
AGAINST	623	71	29
NONE	54	44	27
FAVOUR	65	57	140

Table 4: Confusion Matrix for Task B (System One). $F1_{Against} = 0.8505$, $F1_{Favour} = 0.6114$, Final: $(F1_{Against} + F1_{Favour})/2 = 0.7309$

To support the intuition that network-based features play a crucial role in this model, we explore the Importance of the Features. Results are given in Table 4.2 (Top 10).

	Feature	Importance
1	NW_Retweet1	0.13
2	NW_Friend1	0.12
3	NW_Quote2	0.04
4	Created_at	0.02
5	WE24	0.02
6	Statuses_count	0.02
7	NW_retweet2	0.02
8	WE14	0.02
9	We10	0.01
10	WE25	0.01

Table 5: Top 10 Features' Importance. Legend: NW = MDS dimension of the network; WE = Word-Embedding dimension.

The top three far more important features were dimensions extracted by the MDS approach explained in section 2.2.

5 Conclusion

For SardiStance, the System One proposed here performed well in the Task B, while it has a much poorer result in Task A. It exploits a simple method to handle the network-based information, while further refinement should be made on the exploitation of text-based information. In this way

we want to stress the importance of data mashup, as the system we deployed showed better results for Task B which contains, in addition to texts, information of a different nature derived from network structures.

It is to be expected that more networks should carry similar information. A future direction of research should be the joint analysis of the Networks. There is a sparkling community working on multilayer Networks (De Domenico et al., 2013) (Durante et al., 2017) that may inspire more effective use of this joint information.

References

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. CEUR-WS.org.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Li, 2020. *xgboost: Extreme Gradient Boosting*. R package version 1.0.0.2.

Alessandra Teresa Cignarella, Mirko Lai, Cristina Bosco, Viviana Patti, and Paolo Rosso. 2020. SardiStance@EVALITA2020: Overview of the Task on Stance Detection in Italian Tweets. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2020)*. CEUR-WS.org.

Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A. Porter, Sergio Gómez, and Alex Arenas. 2013. Mathematical Formulation of Multilayer Networks. *Physical Review X*, 3(4):041022, October.

Daniele Durante, David B. Dunson, and Joshua T. Vogelstein. 2017. Nonparametric bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112(520):1516–1530.

Jerome H. Friedman. 1999. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38:367–378.

Verena Lyding, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell'Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. 2013. PAISÀ corpus of italian web text. Eurac Research CLARIN Centre.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

R Core Team, 2019. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Manuela Sanguinetti, Gloria Comandini, Elisa Di Nuovo, Simona Frenda, Marco Stranisci, Cristina Bosco, Tommaso Caselli, Viviana Patti, and Irene Russo. 2020. Overview of the evalita 2020 second hate speech detection task (haspeede 2). In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

Dario Solari, Andrea Sciandra, and Livio Finos. 2019. Textwiller: Collection of functions for text mining, specially devoted to the italian language. *Journal of Open Source Software*, 4(41):1256.