Inferring Political Leaning on X (Twitter): A Zero-Shot Approach in an Italian Scenario

Caterina Senette[1,*], Margherita Gambini[1], Tiziano Fagni[1], Victoria Popa[1,2] and Maurizio Tesconi[1]

[1]*Institute of Informatics and Telematics (IIT) - CNR, Via Giuseppe Moruzzi, 1 56124 Pisa – Italy*
[2]*Università di Pisa, Dipartimento di Computer Science, Largo Bruno Pontecorvo, 3, 56127 Pisa*

Abstract

In recent years, there has been growing attention on predicting the political orientation of active social media users, aiding in political forecasts, modeling opinion dynamics, and understanding user polarization. Existing methods, primarily for X (Twitter) users, use content-based or a blend of content, network, and communication analysis. The latest research highlights that a user's political stance mainly hinges on their views on key political and social issues, prompting a shift towards detecting user stances through their content shared on social networks. This work investigates the use of an unsupervised stance-detection framework Tweets2Stance (T2S) based on zero-shot classification (ZSC) models [1] to predict users' stances toward a set of social-political statements using content-based analysis of their X (Twitter) timelines in an Italian scenario. The ground-truth user stances are drawn from Voting Advice Applications (VAAs), tools aiding citizens in identifying their political leanings by comparing their preferences with party stances. Leveraging the agreement levels of six parties on 20 statements from VAAs, the study aims to predict Party p's stance on each statement s using X (Twitter) Party account data. T2S, employing zero-shot learning, proves effective across various contexts beyond politics, showcasing a minimum MAE of 1.13 despite a general maximum F1 value of 0.4, demonstrating significant progress given the task complexity.

Keywords

user stance detection, Zero-shot learning, unsupervised ML, political leaning, X (Twitter), VAA

1. Introduction

During the last few years, there has been a growing attention towards social media for what is explicitly shared among users (content, thoughts, and behavior), as well as for what is hidden and latent. Among this latent information, the user's stance, i.e. the expression of a user's point of view and perception toward a given statement [2], is particularly interesting; in fact, stance detection on social media is an emerging opinion mining paradigm that well applies in different social and political contexts, and for which many researchers are working to propose solutions ranging from natural language processing, web science, and social computing [3, 4, 5, 6, 7, 8, 9, 10]. Some work [3, 11] dealt with stance-detection at the user-level; however,

ITASEC 2024: The Italian Conference on CyberSecurity, April 08–12, 2024, Salerno, Italy

[*] Corresponding author.

caterina.senette@iit.cnr.it (C. Senette); margherita.gambini@iit.cnr.it (M. Gambini); tiziano.fagni@iit.cnr.it (T. Fagni); victoria.popa@iit.cnr.it (V. Popa); maurizio.tesconi@iit.cnr.it (M. Tesconi)
0000-0002-4411-7134 (C. Senette); 0000–0003–0640–2724 (M. Gambini); 0000–0003–1921–7456 (T. Fagni); 0000–0001–8228–7807 (M. Tesconi)

to the best of our knowledge, a completely unsupervised technique exploiting user's textual content only has never been explored. Hence, the work herein described investigates the use of an unsupervised stance-detection framework based on zero-shot learning models and previously introduced by us [12, 1] named *Tweets2Stance* (*T2S*), to detect the stance of a X (Twitter) account using its timeline in an Italian scenario. The idea for this framework stems out from observing how Voting Advice Applications (VAAs) work. Voting Advice Applications, originally developed in the 1980s as paper-and-pencil civic education questionnaires [13], are online tools that aid citizens, mainly before elections, to identify their political leaning by comparing their policy preferences with the political stances of parties or candidates running for office. VAAs are widespread in many countries and have a crucial role in online election campaigns worldwide. Basically, the user marks its position on a range of policy statements. The application compares the individual's answers to the positions of each Party or candidate and generates a rank-ordered list or a graph indicating which Party or candidate is located closest to the user's policy preferences. One of the crucial elements of the VAAs is the questionnaire: the selection of the statements, their balance among the political poles, and their phrasing have an impact both on the way in which users respond, as well as on the overall users' engagement on the poll itself. For these reasons, the VAA's issued statements should cover the spectrum of the most important topics of an election campaign and adequately show crucial differences among all the competitors in the political scenario for which the VAA is designed [14]. This careful definition of the questionnaire, i.e. taking into high consideration the main topics under discussion at a certain time, suggested us the possibility of using the official position of Italian parties about specific political statements (during a certain political election period) as the ground-truth to determine that stance from the timeline of the X (Twitter) Party accounts in a completely unsupervised way [1]; notice that only tweets written during the pre-election period are considered.

Objectives Starting from the knowledge of the agreement level of six parties on 20 different statements (VAA's statements), the objective of the study is to predict the stance of a Party p toward each statement s exploiting what the X (Twitter) Party account wrote on X (Twitter). Differently from previous works in the literature [3], our classification model is built for different topics and we come up with a fine-grained stance-detector solution working along five classes that could be generalized to various spheres, not just the political one.

2. Related Work

Stance-detection is an emerging opinion-mining paradigm that well applies in several social and political scenarios. The state of the art resumed in a highly valuable survey [3] highlights the importance of categorization since stance-detection can be classified according to the target (single, multi-related, or claim-based), according to the task type (detection or prediction) or distinguishing between stance at user level or the statement level. At the statement level [17, 18],

[1] the Italian Parties' official positions about 20 political statements were kindly provided by the Observatory on Political Parties and Representation [15] based on the VAA *NavigatoreElettorale* for the European Elections 2019 [16]

whose objective is to predict the stance described in a piece of text, previous research works are mainly based on Natural Language Processing (NLP) methods and classification tasks with three classes (support/against/none). Instead, at the user level, the objective is to predict the stance of a user toward a given topic and generally, prediction solutions incorporate different users' attributes along with the text of their posts. Our work falls under the category of stance detection tasks at the user level, specifically focusing on target-specific stances—a common approach in social media stance detection. This involves predicting stances on specific topics, often using separate classification models for each topic. Notable approaches [19, 4, 5, 20] utilize post text along with various user attributes, typically employing binary classification (support and against). Lynn et al. [11] explored using user-level features alone versus document-level features in predicting tweet stances without the tweets highlighting the importance of integrating user features into predictive systems. Other target-specific strategies in literature were conducted at the statement level [6, 7, 8, 21]. In [6] the approach was conducted at the statement level through unsupervised methods, and classification was made along three positions (favour, against, neither). In [7] is introduced a stance-detection shared task, where teams inferred three-level tweet stances using natural language systems: for, against, or neutral towards the given target. Divided into supervised (Task A) and unsupervised (Task B) sub-tasks, they received 19 and 9 team submissions respectively. The highest F-score reached was 67.82 for Task A and 56.28 for Task B. As mentioned above, target-specific approaches could consider single or multiple targets. Usually, the concept of multi-target classification has been used to analyse the relation between two political candidates by using domain knowledge about these targets. In that case, the same model can be applied to different targets on the hypothesis that the same piece of text that contains the stance in favour to a target, it also implicitly contains the stance against the other [22, 9, 23]. Our method handles a broad multi-target classification task, where each statement represents a specific target. Unlike previous methods, it operates without the need for pre-selected texts or distinct models for each target

2.1. Machine Learning (ML) approaches

Among ML features for stance detection, the literature distinguishes between linguistic features, revealing stance based on text-linguistic features [24, 7], and users' vocabulary, which is based on their choice of words [10, 25]. Since textual cues could refer both to textual features, sentiment, and semantics, we limit our attention to textual features. In this context, the most used ML approaches are based on *supervised* techniques [19, 5, 23, 18, 26]. Some works attempted to enrich dataset entities applying *unconstrained supervised* methods such as transfer learning, weak-supervision, and distant supervision methods for stance detection [6, 4]. Other innovative approaches are those that propose *unsupervised learning* strategies [10, 27, 28] exploiting clustering techniques and embeddings representations of users' tweet[29]. The limitations across these studies include: (i) time-intensive data collection and analysis, particularly with network-based approaches; (ii) challenges in accessing or retrieving necessary data due to stringent social media data protection policies; (iii) most models are limited to two or three stance classes at most; (iv) reliance on supervised or semi-supervised models, which require large datasets and have limited generalizability tied to training sets[30]. For all these reasons, the recent challenge for user-level and target-specific stance detection is to move towards unsupervised systems exploiting textual content only. To this aim, a ZSL technique exploiting advanced pre-trained Natural Language Inference (NLI) models [24, 31] can be a viable solution as our T2S framework proved.

3. Task Definition

The task is to predict the stance A_s^u of a Social Media User u with respect to a social-political statement (or sentence) s making use of the User's textual content timeline on the considered social media (e.g., the X (Twitter) timeline). The stance A_s^u represents a five-level categorical label: *completely agree* (5), *agree* (4), *neither disagree nor agree* (3), *disagree* (2), *completely disagree* (1). The integer mappings used by the Tweets2Stance framework are shown in parentheses.

The desired ground-truth is the label G_s^u, which is the known agreement/ disagreement level of User u in regard to sentence s. Remind that the ground-truth is only used to evaluate our proposed $Tweets2Stance$ framework and find its optimal parameters; no training step ever occurs. In this work, we assume that users are the X (Twitter) accounts of six Italian Parties, as the following section will detail.

4. Data collection and Pre-processing

The political scenario under analysis refers to the European and Municipal elections in Italy on 26th May 2019, when Italian citizens were called for the election of the Italian representatives to the European Parliament. The number of Members of the European Parliament (751 deputies in total) for each country is approximately proportional to the population. In 2019, Italy had to elect 76 deputies. Contextually, Italian voters had also to participate in the municipal election of mayors, municipal and district councillors (in about 3800 Italian municipalities), with a planned run-off on 9th June 2019. In that context, we focused our attention on the six major parties in Italy: three center-right parties including Forza Italia (FI), Fratelli d'Italia (FDI), and Lega, two left-wing parties including Partito Democratico (PD) and +Europa (+Eu)[2], and the Movimento 5 Stelle (M5S) representing a sort of third pole at that time. The Italian parliament included other minor parties, especially on the left- wing, representing less than 5% of the Italian population each. We did not consider these parties in the current study. As previously said, we started from the assumption that knowing the parties' answers on the VAA's statements, it is possible to predict the stance of a Party p in regard to each statement s exploiting what the Party wrote on X (Twitter). The definition of the 20 statements (Table 3 in Appendix A) that express the political positions of the six referenced parties towards selected themes under discussion in Italy and in Europe in 2019, was entrusted to a group of political experts [15, 16] who provided us with the ground-truth G_s^p for each Party p and statement s on which the current work is based. At first, we collected timelines of the official X (Twitter) account of each party using the official X (Twitter) API[3]. Considering the speed with which political discussion nowadays takes place, especially on social media, the observation period was adequately chosen in order to maximize the number of tweets avoiding noise and off-topic content. Furthermore, to intercept any

[2] +Europa was recently born in 2018 and it is characterized for a pro-European and liberal orientation.

[3] https://developer.X(Twitter).com/en/docs

valuable information or discussion trends over time we have extended the analysis considering four temporal ranges and built the associated datasets[4] as described in Table 1.

Table 1
The four studied datasets with the total number of tweets before the pre-processing step. D_j contains j months of tweets.

	D_3	D_4	D_5	D_7
Period	[2019-03-01, 2019-05-25]	[2019-02-01, 2019-05-25]	[2019-01-01, 2019-05-25]	[2018-11-01, 2019-05-25]
#tweets	20'266	25'979	34'736	44'370

As a preliminary step, since the text collected from tweets contains a lot of noise and irrelevant information, we pre-processed the tweets in order to remove anything which doesn't have predicting significance, such as: item URLs, "$RT@user$:" prefix of retweets, mentions at the beginning of a reply tweet, tweets with $\{1, 2, 3\}$ words and empty tweets, hashtags and emojis (replaced with empty string). Lastly, since we wanted to test our prediction approach on English tweets as well, we further translated the Italian tweets using the *google_trans_new*[5] Python package.

5. Framework Design

This section briefly describes the proposed Tweets2Stance (T2S) framework (Fig. 1) to detect the stance A_s^u of a X (Twitter) User u in regard to a sentence s, exploiting its X (Twitter) timeline $TL_u = [tw_1, ..., tw_n]$. More details of the framework are provided in a previous work where we have extensively introduced it [1].

A User might either not talk about a specific political argument (here expressed with sentence s), or debate on an issue not risen by our pre-defined set of statements. For these reasons, our framework executes a preliminary *Topic Filtering* step, exploiting a Zero-Shot Classifier (ZSC) to get only those tweets talking about the topic tp of the sentence s. A ZSC is a language-model-based method that, given a text and a set of labels (e.g., topics), assigns a classification probability score to each label [21]. The higher the score assigned to a label, the higher the likelihood that the input text pertains to that specific label. ZSC does not require further fine-tuning on the target dataset. After obtaining the in-topic tweets $I_{tp_s}^u$through Topic Filtering, the Agreement Detector module employs the same ZSC to detect the user's agreement/disagreement level. In Fig. 1 we use colour-codes to identify the four parameters of the *T2S* framework that we'll vary during our experiments, as explained in Section 6.

Topic Filtering The *Topic Filtering* module extracts the in-topic tweets $I_{tp_s}^p$ from the X (Twitter) Timeline TL_p of Party p, using the topic tp_s associated with sentence s (e.g., the topic for the sentence "*overall, membership in the EU has been a bad thing for the UK*" can be "*UK membership in EU*"). The topic definitions for all considered sentences can be found in the linked repository.

[4] The four raw datasets can be found at https://github.com/marghe943/Tweets2Stance_dataset
[5] https://pypi.org/project/google-trans-new/

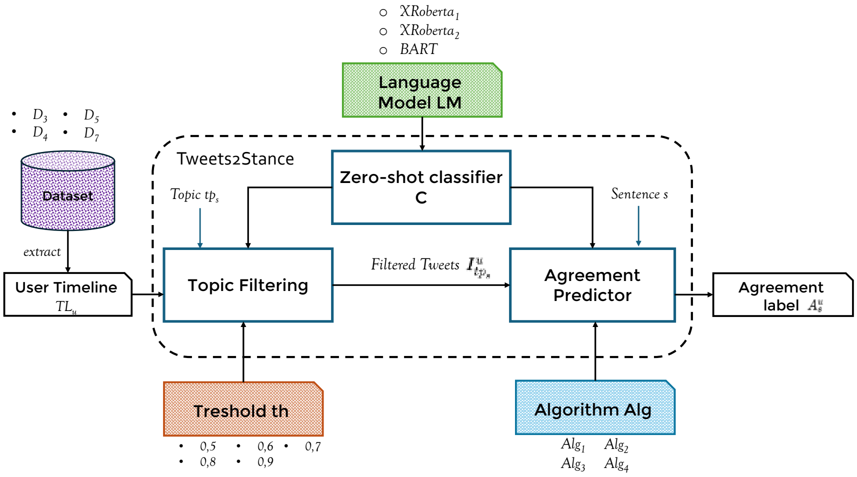

Figure 1: Our Tweets2Stance framework to compute the agreement/disagreement level A_s^u of User u in regard to sentence s. The inputs are the X (Twitter) timeline TL_u extracted from a certain time-period dataset D_i, the sentence s, the topic tp associated with s, a language model LM, a threshold th and an algorithm Alg. The highlighted components are the parameters that we'll vary during our experiments, as explained in Section 6.

The module utilizes the ZSC C to retrieve the in-topic tweets $I_{tp_s}^p$ and their corresponding topic scores $T_{tp_s}^p$.

Agreement Detector The *Agreement Detector* module (Fig. 1 - Module 2) computes the final five-valued label A_s^p through an algorithm $Alg(T_{tp_s}^p, S_s^p)$, defining

$$S_s^p = \{C(tw_i, s) | tw_i \in I_{tp_s}^p\} \tag{1}$$

as the C scores of tweets $I_{tp_s}^p$ with respect to sentence s, each one indicating the relevance and agreement of tweet tw_i with sentence s.

Each employed algorithm Alg exploits one of the following mapping functions:

$$M1(s) = \begin{cases} 1 & \text{if } s \in [0, 0.2) \\ 2 & \text{if } s \in [0.2, 0.4) \\ 3 & \text{if } s \in [0.4, 0.6) \\ 4 & \text{if } s \in [0.6, 0.8) \\ 5 & \text{if } s \in [0.8, 1] \end{cases} \tag{2}$$

$$M2(s) = \begin{cases} 1 & \text{if } s \in [0, 0.25) \\ 2 & \text{if } s \in [0.25, 0.5) \\ 3 & \text{if } s \in [0.5, 0.75) \\ 4 & \text{if } s \in [0.75, 1] \end{cases} \tag{3}$$

where $M1(s)$ ranges from 1 to 5, corresponding to the five agreement/disagreement labels defined in Section 3. Similarly, $M2(s)$ ranges from 1 to 4, representing an intermediate agreement/disagreement scale. Specifically, $M2(s) = \{1, 2\}$ has the same meaning as in Section 3, while $M2(s) = 3$ indicates agreement and $M2(s) = 4$ represents complete agreement. The rationale behind this intermediate mapping is explained in Algorithm 4 [1].

We defined four algorithms with different complexity levels, details of each one are provided in the Appendix B and the already mentioned work [1].

6. Experimental Setup

6.1. Baselines

It is a good practice to compare the proposed methods with a bunch of baselines. To the best of our knowledge, no baseline method has been devised for the typology of our stance detection task yet: unlike our approach, the state-of-the-art unsupervised user-stance detection method proposed by Darwish et al. [10] cannot operate without context information from other users and it is not suitable for a multi-class ordinal classification like our case. Therefore, the following baselines to compute A_s^p for Party p and sentence s were used:

Random A_s^p is set to a random integer picked from a discrete uniform distribution of $int \in [1, 5]$. The *numpy* random method[6] was used with random seed set to 42. .

Predict 3 A_s^p is set to 3 (*neither disagree, nor agree*).

Sentence Bert The newest Transformer-based language models like BERT can be used as feature extractors [32], providing contextual word and sentence embeddings. The Sentence-Bert architecture of the *Sentence Transformers* Python library[7] was used with the English *all-mpnet-base-v2* model on translated tweets, and with the multi-lingual model *distiluse-base-multilingual-cased-v1* on the Italian tweets.

6.2. Experiments in detail

As already explained in section 5, our *T2S* method has got four parameters to tune: the language model *LM* to be used for zero-shot classification, the dataset *Di* from which extract the X (Twitter) timeline TL_p, the algorithm Alg for the *Agreement* step, and the threshold value *th* for the $Topic$ *Filtering* step. Considering the values of those parameters in Fig. 1, we carried out each experiment having in mind the four research questions summarized in Table 2 and ordered by specificity.

6.3. Evaluation

In evaluating the stance detection model, traditional metrics like MSE, MAE, R2 Score, and Residual Plots are common. However, a bespoke metric is needed to address varying error importance across stance classes. For instance, misclassifying *agree* instead of *completely disagree* carries a different weight than *neither disagree, nor agree* instead of *agree*. In the absence of such a metric, MAE is chosen. Lastly, since the predicted value is an integer among $\{1, 2, 3, 4, 5\}$, a classification evaluation metric was considered as well: the weighted *F1* score was picked, since it summarizes both Precision and Recall [33]. The *sklearn.metrics* Python package was used to compute both *MAE*[8] and *F1_weighted*[9]

[6] https://numpy.org/doc/stable/reference/random/generated/numpy.random.randint.html
[7] https://www.sbert.net/
[8] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html
[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Table 2
Description of all carried out experiments

Experiment Name	Research Question
Best language model *LM*	Which is the best language model *LM* for zero-shot classification? Which is the best model to deal with Italian tweets? All in all, is an English model better?
Best dataset *D*	Fixed the language model *LM*, which is the best dataset to work on, considering all proposed algorithms? Hence, which is the best time period to listen to before a Political Election?
Best algorithm *Alg*	Fixed the language model *LM* and dataset *Di*, which is the best algorithm to work on, considering all evaluated thresholds *th*? Are all our proposed algorithms better than the baselines (subsection 6.1)? Are the more complex algorithms better or not?
Best threshold *th*	Fixed the language model *LM*, the dataset $D - Nmonths$ and the algorithm *Alg*, which is the best filtering threshold *th*, hence the optimal set-up?
Party Analysis	Fixed the optimal setup for our framework, which are the Parties on which *T2S* behaves well or poorly?

7. Results and Discussion

7.1. Best Language Model LM

First, we explored which is the best language model for ZSC on Italian tweets: a model pre-trained on a mix of languages including Italian or one fine-tuned on Italian text? Also, would results improve with an English model on translated tweets? Furthermore, would the results benefit from using an English language model on translated tweets instead? We answered these questions by looking at Fig. 2: each cell (LM_i, D_j) indicates the minimum MAE (maximum F1) obtained with our *T2S* method for a certain language model LM_i and dataset D_j by varying the algorithm *Alg* and the threshold *th* according to Fig. 1.

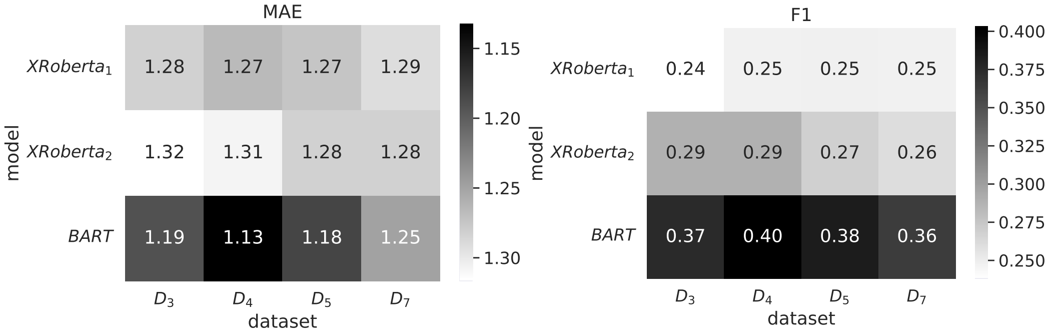

Figure 2: Best MAE and F1 values of our *T2S* method for each couple (LM_i, D_j) of language models and datasets. Darker colors indicate optimal values for both metrics.

Among the cross-lingual models $XRoberta_1$ and $XRoberta_2$, the best one seemed to be $XRoberta_1$: it had an overall better MAE, while F1 results were close to $XRoberta_2$'s; we considered MAE as the first metric to judge the performances since it tells how much we are close to the correct answer. Apparently, fine-tuning on an Italian translation of a subset of the MNLI dataset ($XRoberta_2$) doesn't contribute a lot to text classification in our *T2S* framework. All in all, the best choice is translating the pre-processed tweets in English and using an English model like *BART*: it reached significantly higher values on both MAE and F1. Supposedly, using a model pre-trained and fine-tuned on a single language gives better results for our prediction task: learning on a single language allows us to focus on more details and features of the language.

7.2. Best Dataset D

The choice of the dataset's time period (D_i) as one of the parameters to tune is motivated by the use of T2S for stance detection during political elections, where the proximity to the elections may impact the likelihood of users discussing socio-political topics. Fixed the language model $LM = BART$, the dataset D_4 was immediately detected as the best one, since it had the best MAE and F1 (Fig. 2). Presumably, the X (Twitter) political discussion four months before the Italian elections was enough to grasp the Parties' stances. We evaluated the mean MAE and mean F1 for each cell (LM_i, D_j) of Fig. 2 as well, but the results confirmed *BART* and D_4 as the best language model and dataset.

7.3. Best Algorithm Alg

Once the language model $LM = BART$ and dataset D_4 were chosen, we tested our algorithms Alg against the baselines *random*, $predict_3$, and $sentence_bert$, examining the best Alg across all thresholds th. Fig. 3 describes how much each algorithm performed across different thresholds. These results include the performances of the three baselines as well. Altogether, the optimal algorithm can be identified in $Alg3$: F1 seemed to contradict it and bend over $Alg4$ instead, but the gain over the prediction error is far more important. This result suggests that assigning the neutral label (*neither disagree, nor agree*) only when there's a minimum number of tweets m does not boost the performance of our *T2S* method. Also, we executed $Alg4$ with $m = \{2, 3\}$, finding out that the results didn't vary a lot from each other; therefore, we showed $Alg4_{m=3}$ in Fig. 3.

7.4. Best Threshold th and Party Analysis

Fixed the language model $LM = BART$, the dataset D_4 and the algorithm $Alg3$, threshold $th = 0.6$ was immediately detected as the optimal one, since it had the best MAE and a good F1 (Fig. 3). Therefore, the best setup su_{opt} of our *T2S* framework was (LM, D_j, Alg, th) = ($BART$, D_4, $Alg3$, 0.6). To explore the specific performance of our *T2S* method over the Parties, we used the optimal setup su_{opt} but by varying the threshold th. Fig. 4 shows the results. Each point indicates the MAE (F1) on the 20 sentences' agreement level A_s^p for a certain Party p. Each Party behaves differently, thus it is likely that *T2S* highly depends on the Party's timeline in terms of how much it generally writes, how much it writes in-topic, and how much it writes using figures of speech or hashtags and emojis (which we removed). Looking at both the MAE and F1, we observed a regular trend for thresholds $th = \{0.8, 0.9\}$ for five parties out of six: the outlier Party

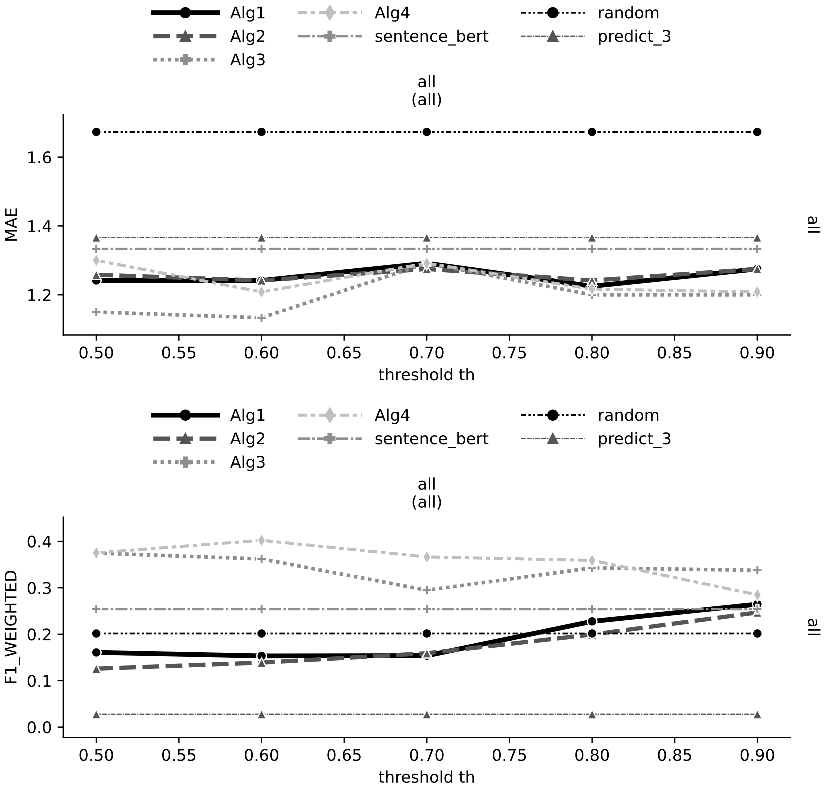

Figure 3: MAE and F1 of our four proposed algorithms *Alg*s and the three baselines by varying the threshold *th*. It is shown *Alg*4 with $m = 3$ (see B).

*Mov*5*Stelle* was more predictable for those thresholds. That may happen because the user's timeline deals with a certain statement in a clearer way; for example, looking at *Mov*5*Stelle* and *forza_italia*'s tweets filtered for the sentence *S*19 and $th = 0.9$, we saw that *Mov*5*Stelle* wrote clearer and explicit tweets supporting the argument (it completely agrees), while from *forza_italia*'s timeline it's not immediately clear that it disagrees; *forza_italia* tweeted about tax reduction, fewer fees on families, and job creation, in that case, our *T2S* framework marked it 'completely agree' since the party didn't explicitly disagree with income support for the poorest as beneficial for the Italian economy.

8. Conclusions and Future Work

In this work, we investigate the use of an unsupervised stance-detection framework Tweets2Stance (T2S) based on zero-shot classification [1] to predict users' stances toward a set of social-political statements using content-based analysis of their X (Twitter) timelines in an Italian scenario. In particular, we dealt with the stance of 20 political statements for the six major parties in Italy. Results showed that, although the general maximum F1 value was 0.4, *T2S* could correctly predict the stance with a general minimum MAE of 1.13, which is a great

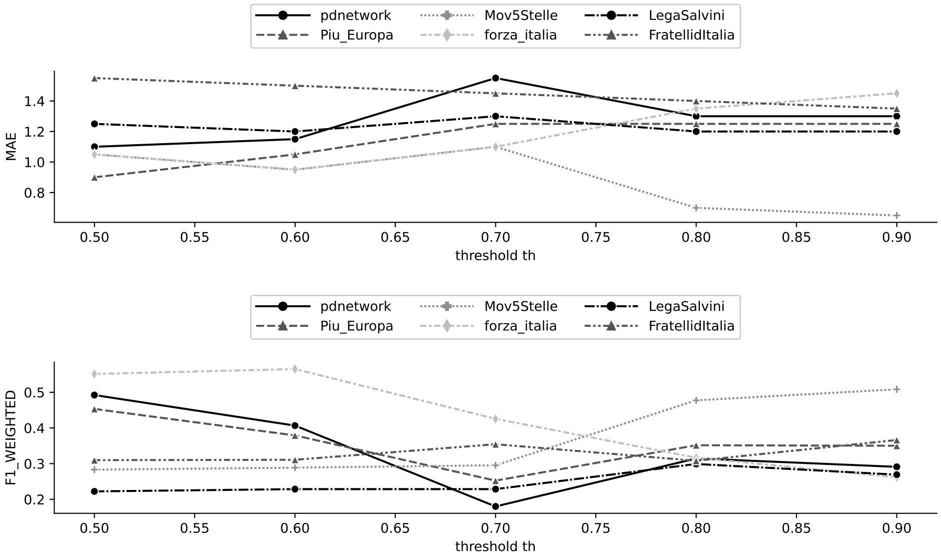

Figure 4: MAE and F1 computed for each Party over the stance predictions of the 20 VAA statements. The optimal su_{opt} is used, but the threshold *th* varies.

achievement considering that MAE tells how close we are to the correct answer, and that we worked with a final five-valued label. Also, as we hypothesized, the *T2S*'s performance highly depends on how the X (Twitter) account of the Party (hence the social media user) writes, e.g. the employed figures of speech, the words used, and so on. As mentioned when introducing the work, the approach is potentially generalizable to several topics. If applied to political discourse, it could represent the first step of a pipeline whose output is the user's political leaning. In the near future, we will investigate how *T2S*'s agreement levels output can be used to derive the political leaning of a social media user, for example by trying to emulate a VAA algorithm. Besides, we hope to apply it to detect extremist accounts on social media; however, a domain expert may be needed to define precise social statements to use. Future research could address T2S limitations by using advanced models like GPT-4 or conversational AI such as ChatGPT for robust stance detection.

Acknowledgments

We thank Project SERICS (PE00000014) - NRRP MUR program funded by the EU- NGEU, and Project SoBigData-PlusPlus Grant Agreement number: 871042 CUP B54I1900639000.

References

[1] M. Gambini, C. Senette, T. Fagni, M. Tesconi, From tweets to stance: An unsupervised framework for user stance detection on twitter, in: International Conference on Discovery Science, Springer, 2023, pp. 96–110.

[2] D. Biber, E. Finegan, Adverbial stance types in english, Discourse processes 11 (1988) 1–34.

[3] A. ALDayel, W. Magdy, Stance detection on social media: State of the art and trends, Information Processing & Management 58 (2021) 102597.

[4] M. Dias, K. Becker, Inf-ufrgs-opinion-mining at semeval-2016 task 6: Automatic generation of a training corpus for unsupervised identification of stance in tweets, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 378–383.

[5] Y. Igarashi, H. Komatsu, S. Kobayashi, N. Okazaki, K. Inui, Tohoku at semeval-2016 task 6: Feature-based model versus convolutional neural network for stance detection, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 401–407.

[6] I. Augenstein, T. Rocktäschel, A. Vlachos, K. Bontcheva, Stance detection with bidirectional conditional encoding, arXiv preprint arXiv:1606.05464 (2016).

[7] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, C. Cherry, Semeval-2016 task 6: Detecting stance in tweets, in: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), 2016, pp. 31–41.

[8] S. Hamidian, M. T. Diab, Rumor detection and classification for twitter data, arXiv preprint arXiv:1912.08926 (2019).

[9] K. Darwish, W. Magdy, T. Zanouda, Improved stance prediction in a user similarity feature space, in: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, 2017, pp. 145–148.

[10] K. Darwish, P. Stefanov, M. Aupetit, P. Nakov, Unsupervised user stance detection on twitter, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 14, 2020, pp. 141–152.

[11] V. Lynn, S. Giorgi, N. Balasubramanian, H. A. Schwartz, Tweet classification without the tweet: An empirical examination of user versus document attributes, in: Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science, 2019, pp. 18–28.

[12] M. Gambini, T. Fagni, C. Senette, M. Tesconi, Tweets2stance: users stance detection exploiting zero-shot learning algorithms on tweets, arXiv preprint arXiv:2204.10710 (2022).

[13] L. Cedroni, Voting Advice Applications in Europe: The state of the art, Scriptaweb, 2010.

[14] T. Louwerse, M. Rosema, The design effects of voting advice applications: Comparing methods of calculating matches, Acta politica 49 (2014) 286–312.

[15] OPPR, Opi - observatory on political parties and representation, ???? URL: http://opi.sp.unipi.it/opi-political-parties/.

[16] O. on Political Parties, R. (OPPR), Navigatoreelettorale europee 2019, 2019. URL: http://opi.sp.unipi.it/opi-political-parties/oppr-projects/.

[17] A. Murakami, R. Raymond, Support or oppose? classifying positions in online debates from reply activities and opinion expressions, in: Coling 2010: Posters, 2010, pp. 869–875.

[18] M. A. Walker, P. Anand, R. Abbott, J. E. F. Tree, C. Martell, J. King, That is your evidence?: Classifying stance in online political debate, Decision Support Systems 53 (2012) 719–729.

[19] S. Gottipati, M. Qiu, L. Yang, F. Zhu, J. Jiang, Predicting user's political party using ideological stances, in: International Conference on Social Informatics, Springer, 2013, pp. 177–191.

[20] A. Aldayel, W. Magdy, Your stance is exposed! analysing possible factors for stance detection on social media, Proceedings of the ACM on Human-Computer Interaction 3 (2019) 1–20.

[21] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3914–3923. URL: https://aclanthology.org/D19-1404. doi:10.18653/v1/D19-1404.

[22] P. Sobhani, D. Inkpen, X. Zhu, A dataset for multi-target stance detection, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 551–557.

[23] M. Lai, V. Patti, G. Ruffo, P. Rosso, Stance evolution and twitter interactions in an italian political debate, in: International Conference on Applications of Natural Language to Information Systems, Springer, 2018, pp. 15–27.

[24] S. Ghosh, P. Singhania, S. Singh, K. Rudra, S. Ghosh, Stance detection in web and social media: a comparative study, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2019, pp. 75–87.

[25] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, H.-W. Hon, Unified language model pre-training for natural language understanding and generation, Advances in Neural Information Processing Systems 32 (2019).

[26] B. Zhang, M. Yang, X. Li, Y. Ye, X. Xu, K. Dai, Enhancing cross-target stance detection with transferable semantic-emotion knowledge, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3188–3197.

[27] A. Joshi, P. Bhattacharyya, M. Carman, Political issue extraction model: A novel hierarchical topic model that uses tweets by political and non-political authors, in: Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2016, pp. 82–90.

[28] T. Fagni, S. Cresci, Fine-Grained Prediction of Political Leaning on Social Media with Unsupervised Deep Learning, Journal of Artificial Intelligence Research 73 (2022) 633–672.

[29] A. Rashed, M. Kutlu, K. Darwish, T. Elsayed, C. Bayrak, Embeddings-based clustering for target specific stances: The case of a polarized turkey, arXiv preprint arXiv:2005.09649 (2020).

[30] R. Cohen, D. Ruths, Classifying political orientation on twitter: It's not easy!, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 7, 2013.

[31] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, arXiv preprint arXiv:1909.00161 (2019).

[32] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410. doi:10.18653/v1/D19-1410.

[33] F. Sebastiani, Machine learning in automated text categorization, ACM computing surveys (CSUR) 34 (2002) 1–47.

A. Statements

Table 3: Defined topic for each of the 20 sentence (Italian version).

nr.	Sentence	Topic
1	nel complesso, essere membri dell'UE è uno svantaggio	svantaggi dell'Unione Europea
2	l'Italia dovrebbe uscire dall'Euro	uscire dall'euro
3	dovrebbe esistere un esercito comune europeo	esercito europeo comune
4	le multinazionali come Google e Youtube dovrebbero pagare i diritti d'autore e le tasse secondo le regole di ciascun paese europeo	tasse per le multinazionali in relazione alle regole di ciascun Paese Europeo
5	l'integrazione economica europea si è spinta troppo oltre: gli Stati membri dovrebbero riguadagnare maggiore autonomia	autonomia economica dei membri dell'Unione Europea
6	l'Unione Europea dovrebbe riformare la propria politica dell'immigrazione: l'Italia dovrebbe ricevere più supporto dagli altri Stati membri	gestione dell'immigrazione nell'Unione Europea
7	l'Italia dovrebbe intensificare le sue relazioni economiche con la Cina	relazioni economiche dell'Italia con la Cina
8	l'uso ricreativo della cannabis dovrebbe essere legale	uso ricreativo della cannabis
9	l'Islam è una minaccia per i valori dell'Italia	minaccia dell'Islam nei confronti dei valori italiani
10	alle donne deve essere garantita autonomia di scelta sull'aborto	autonomia di scelta sull'aborto
11	ogni forma di auto-difesa all'interno della proprietà privata dovrebbe essere legittima	legittima difesa nella propria abitazione con armi
12	le attività della magistratura devono essere indipendenti dalle pressioni della politica	indipendenza della magistratura dalla politica
13	i bambini, nati in Italia da cittadini stranieri, dovrebbero ricevere la cittadinanza italiana automaticamente	cittadinanza italiana per bambini nati in Italia da famiglie straniere
14	la ricchezza dovrebbe essere redistribuita dai cittadini più abbienti ai cittadini più poveri	redistribuzione della ricchezza verso i piu poveri
15	le imprese dovrebbe poter licenziare i dipendenti più facilmente	possibilita delle imprese di licenziare facilmente i propri dipendenti

Continued on next page

Table 3 – *Continued from previous page*

nr.	Sentence	Topic
16	la Sanità dovrebbe essere più aperta agli operatori privati	apertura della Sanità ad operatori privati
17	proteggere l'ambiente è più importante della crescita economica	importanza della protezione dell'ambiente
18	tagliare la spesa pubblica è un buon modo per risolvere la crisi economica	tagli alla spesa pubblica come soluzione per la crisi economica
19	il sostegno al reddito alle fasce più povere della popolazione è positivo per l'economia italiana	migliorare l'economia aiutando le fasce a basso reddito
20	l'introduzione di una aliquota unica sui redditi ("flat tax") sarebbe di beneficio all'economia italiana	conseguenze della flat tax per l'economia italiana

B. Algorithms ordered by complexity

Algorithm 1 [Alg1] The label A_s^p is computed as

$$A_s^p = \begin{cases} M1\left(\frac{\sum_{i=1}^{|I_{tp_s}^p|} s_i \cdot t_i}{\sum_{i=1}^{|I_{tp_s}^p|} s_i}\right) & \text{if } |\, I_{tp_s}^p \,| \neq 0 \\ 3 & \text{otherwise} \end{cases} \tag{4}$$

where $s_i \in S_{tp_s}^p$ and $t_i \in T_{tp_s}^p$.

Algorithm 2 [Alg2] First, it maps each tweet $tw_i \in I_{tp_s}^p$ into the label $l_i \in \{1, 2, 3, 4, 5\}$ using its sentence score $s_i \in S_s^p$

$$l_i = M1(s_i) \tag{5}$$

then, A_s^p is

$$A_s^p = \begin{cases} \left\lfloor \frac{\sum_{i=1}^{|I_{tp_s}^p|} l_i}{|I_{tp_s}^p|} \right\rceil & \text{if } |\, I_{tp_s}^p \,| \neq 0 \\ 3 & \text{otherwise} \end{cases} \tag{6}$$

The step of assigning l_i to each tweet $tw_i \in I_{tp_s}^p$ (Eq. 5), hopefully returns a more fair A_s^p. In fact, the tweet normalization may help in aggregating the contribution of each tweet (l_i) using the standard mean, which means applying the macro aggregation. In a multi-class classification setup, macro-metric aggregation is preferable if it is suspected that there may be class imbalance; in fact, the values l_i are not balanced with respect to the current sentence s: likely, if a Party p agrees with a sentence, there will be lot of tweets in agreement with it (many $l_i = 4$ or $l_i = 5$) and a few (errors) or no tweets in disagreement (few labels $l_i = 1$, or $l_i = 2$, or $l_i = 3$), and vice-versa.

Algorithm 3 [Alg3] Like $Alg2$, but slightly modifying how A_s^p is computed (Eq. 6). Let's further define V_l as the number of voters for the integer label $l \in \{1, 2, 3, 4, 5\}$

$$V_l = |\{l_i \, : \, l_i = l\}_{i=1}^{|I_{tp_s}^p|}| \tag{7}$$

where l_i are the labels computed from Eq. 5. Let's define $v = max(V_l)$, then

$$A_s^p = \begin{cases} l & \text{if } |\{l \, : \, V_l = v\}| = 1 & \text{(8a)} \\ \left\lfloor \dfrac{\sum_{i=1}^{|I_{tp_s}^p|} l_i}{|I_{tp_s}^p|} \right\rceil & \text{if } |\{l \, : \, V_l = v\}| > 1 & \text{(8b)} \\ 3 & \text{otherwise} & \text{(8c)} \end{cases}$$

where $\lfloor ... \rceil$ is the round function. The majority voting (case 8a) may have a bigger contribution in assigning correct labels than the plain standard mean (case 8b taken from Eq. 6 of $Alg2$), since it better accounts for class imbalance.

Algorithm 4 [Alg4] The previous algorithms take into consideration the neutral label $nl = 3$ (*neither disagree, nor agree*) also when $| \, I_{tp_s}^p \, | \neq 0$. However, we wondered how the results would change if nl was *only* considered when $| \, I_{tp_s}^p \, | = 0$. The neutral label may also be assigned in the presence of a low number of in-topic $I_{tp_s}^p$: in this particular situation, the user may have not taken a position about the current sentence s yet; also, choosing A_s^p looking at just one tweet may not be significant. Therefore, $Alg4$ stems from $Alg3$ having

$$l_i = M2(s_i) \tag{9}$$

where $l_i \in \{1, 2, 3, 4\}$; we define

$$a_s^p = \begin{cases} 3 & \text{if } | \, I_{tp_s}^p \, | < m \\ \text{majority voting (case 8a)} & \\ \text{rounded standard mean (case 8b)} & \end{cases} \tag{10}$$

where m is the minimum number of tweets for which the majority voting algorithm or the standard mean is executed. Since the $\{3, 4\}$ labels in output from $M2(s)$ represent the *agree* and *completely agree* final labels, they must be mapped again to the real final integer labels 4 and 5 respectively (as coded in Table **??**)

$$A_s^p = \begin{cases} a_s^p & \text{if } a_s^p = 1 \lor a_s^p = 2 \\ a_s^p + 1 & \text{if } a_s^p = 3 \lor a_s^p = 4 \end{cases} \tag{11}$$