From Tweets to Stance: An Unsupervised Framework for User Stance Detection on Twitter

Margherita Gambini, Caterina Senette(✉), Tiziano Fagni, and Maurizio Tesconi

Institute of Informatics and Telematics (IIT) - CNR,
Via Giuseppe Moruzzi, 1, 56124 Pisa, Italy
{m.gambini,c.senette,f.fagni,m.tesconi}@iit.cnr.it

Abstract. Current stance inference methods use topic-aligned training data, leaving many unexplored topics due to the lack of training data. Zero-shot approaches utilizing advanced pre-trained Natural Language Inference (NLI) models offer a viable solution when training data is unavailable. This work introduces the *Tweets2Stance - T2S* framework, an unsupervised stance detection framework based on Zero-Shot Learning. It detects a five-valued user's stance on social-political statements by analyzing their Twitter timeline. The ground-of-truth user's stance is obtained from Voting Advice Applications (VAAs), online tools that compare political preferences with party political stances. The *T2S* framework's generalization potential is demonstrated by measuring its performance (F1 and MAE scores) across nine datasets. These datasets were built by collecting tweets from competing parties' Twitter accounts in nine political elections held in different countries from 2019 to 2021. Through comprehensive experiments, an optimal setting was identified for each election. The results, in terms of F1 and MAE scores, outperformed all baselines and approached the best scores for each election. This showcases the ability of T2S to generalize across different cultural-political contexts.

Keywords: user stance detection · transfer learning · unsupervised · Twitter · text content · elections · vaa

1 Introduction

Stance detection (SD) is a text-mining approach that infers the expression of a user's point of view and perception toward a given statement [3]. Unlike sentiment analysis, which categorizes a text as positive, negative, or neutral regardless of a specific target, stance detection focuses on classifying a text based on the user's attitude toward a predetermined target. It is commonly applied in two areas: inferring user agreement/disagreement in social media debates across various contexts (such as political, ideological, and social), and assessing public opinion on products and services [8,15].

ALDayel et al. [2] proposed a recent taxonomy of stance detection tasks. Firstly, the level at which stance is computed must be determined, whether it

A. Bifet et al. (Eds.): DS 2023, LNAI 14276, pp. 96–110, 2023.
https://doi.org/10.1007/978-3-031-45275-8_7

involves detecting the stance expressed in a piece of text or inferring the stance of a user towards a specific target based on their posted content and context. Secondly, the targets for detecting the stance need to be identified. These targets can be single (e.g., a specific topic), multi-related (where expressing a stance towards one target implies a stance towards similar targets), or claim-based (determining whether a text or user confirms a claim).

In this study, we focus on the investigation and measurement of public opinion on several issues as *user stance detection on multiple unrelated targets*. The analysis of texts extracted from social media users' posts provides valuable information for making such inferences. However, existing literature proposes mixed approaches that partially exploit text analysis in conjunction with user behaviour analysis, such as likes, retweets, and the network of contacts [1,10]. Additionally, user stance detection on unrelated targets poses computational challenges [2]. Limitations in content-based stance detection approaches include the inherent difficulty of processing natural language, the need for large annotated corpora of tweets and language-specific resources, the lack of unsupervised transfer learning to generalize across unrelated targets, and the requirement of training separate classifiers for each target. State-of-the-art research often focuses on two (support, against) or three levels of stance (including the neutral class[1]), and existing unsupervised methods based on clustering techniques in user networks are not suitable for inferring a user's stance for different unrelated targets.

Therefore, in an attempt to address these issues and focus solely on a content-based approach, we present *Tweets2Stance*(T2S), an unsupervised framework for stance detection. T2S analyzes the content of a user's social media (e.g., Twitter) timeline using Zero-Shot Classification (ZSC) techniques [21] to detect their stance towards specific socio-political statements (targets), considering five levels of agreement (completely disagree, disagree, neither disagree/nor agree, agree, completely agree).

To sum up, this work investigates a completely unsupervised solution to user-stance detection by answering the following research questions:
RQ1 – *What are the performances and insights of a completely unsupervised user-stance detection framework leveraging zero-shot classification capabilities on textual contents only?* Here, we also compare T2S's performance when used to detect either five or three stance classes.
RQ2 – *Is there a general framework that performs well across different political contexts?* Here, we explore the generalizing capabilities of T2S.

Contributions. To the best of our knowledge, we *filled the gap* of investigating an *unsupervised content-based-only* model leveraging an advanced Natural Language Processing technique (that is the *Zero-Shot Classification*) to detect a *five-level stance* of a user on *multiple and diverse targets* (the socio-political statements on different political contexts). Furthermore, the framework can be adapted for various scenarios, extending beyond the specific political context

[1] The *neutral* level indicates that the user or text did not express a stance on that target or does not take a stance at all.

addressed in this study. Additionally, we offer a set of labeled datasets that can assist other researchers in their endeavors involving unsupervised stance-detection techniques at the user level.

The remainder of this paper is organized as follows: Sect. 2 discusses related work. In Sect. 3, we define the user stance-detection task and dataset collection. Section 4 details the Tweets2Stance framework and experiment settings. Section 5 summarizes and discusses the results, highlighting limitations. Finally, Sect. 6 concludes and suggests future work.

2 Related Work

In the classical definition [3], user-level stance detection involves detecting a user's stance on a given topic based on their authored text. In the following paragraphs, we summarize the literature on *user-based* stance detection in social media, considering the features used and the learning approach.

Content and Behavioural Features. Rashed et al. (2021) [17] focused on user-based stance detection using content features alone. They employed Google's Multilingual Universal Sentence Encoder (MUSE) and a pre-trained CNN to extract tweet embeddings. User representation was obtained by averaging these embeddings and projected onto a two-dimensional plane using the Uniform Manifold Approximation and Projection (UMAP) technique. The authors utilized hierarchical density-based clustering (HDBScan) to classify users into pro and anti stances, achieving an F1 score of 0.86 on a dataset of 168k users. Moreover, interaction patterns and historical behaviour on social media, in addition to content features, can be used as well: Darwish et al. (2020) [4] successfully clustered users based on feature similarities such as retweets, common hashtags, and retweeted accounts; Aldayel et al. (2019) [1] achieved an F1 score of 0.72 by leveraging users' online behaviour cues; Thonet et al. (2017) [18] considered both text and social interactions to uncover topics, user viewpoints, and discourse; Magdy et al. (2016) [14] focused on elements such as retweets, replies, mentions, URLs, and hashtags to predict unexpressed stances (a stance that may or may not have transpired *yet*), not to detect them (an existing stance in past data) (See footnote 1). Lastly, Fraiser et al. (2018) [6] used content-based and social-based proximities in a multi-layer graph, achieving an F1 score of 0.95.

Supervised and Unsupervised Learning. Stance detection techniques using supervised learning rely on large annotated datasets [16]. User-based stance detection has received less attention in these competitions, but notable studies include Aldayel et al. [1] and Magdy et al. (2016) [14]. Aldayel et al. (2019) trained a stance detector for each topic using the SemEval2016 dataset with 3,000 users. Magdy et al. (2016) collected timelines of 44,000 users discussing the Paris terrorist attack, while Fraiser et al. (2018) [6] applied a proximity-based two-level stance detector to different datasets related to political events and gun control. More recently [9,11], the trend in language processing for stance-detection tasks

relies on language representation models (e.g., BERT [5]) pre-trained on large un-annotated corpora and *fine-tuned* on labelled and domain-specific datasets [5,21]. The work of Devlin et al. (2018) [5] demonstrated how BERT led to considerable performance improvements for NLP tasks such as sentiment analysis. Ghosh et al. (2019) [9] reported BERT's successful use in stance detection compared to other techniques. Here, the BERT model takes the text as input to generate representations of the words through multiple transformer layers, and then the system is fine-tuned on the task-specific data. Lately, Zang et al. (2023) [22] leveraged ChatGPT for text-based stance detection, achieving state-of-the-art or similar performance on SemEval-2016 [16] and PStance [13] datasets.

To the best of our knowledge, no unsupervised technique for user-based stance detection has yet utilized advanced Transformer-based language models. Existing unsupervised methods, such as Darwish et al. (2019) [4], Trabelsi et al. (2018) [19], and Fraiser et al. (2018) [6], rely on standard linguistic features like n-grams and keyword counts. Recognizing this gap and the increasing use of pre-trained models in stance detection, we propose Tweets2Stance, the first stance detector to work on a five-level stance. We evaluated T2S across diverse political contexts, achieving satisfactory results despite the challenging task.

Comparing the T2S framework to state-of-the-art user-based stance detection methods presents several challenges. Firstly, existing methods (e.g., Rashed et al., 2021 [17]) filtered tweets by removing mentions of specific targets, which is incompatible with our work as our topic lacks a well-defined person or organization. Additionally, these methods rely on timelines of users connected through specific keywords, while T2S aims to infer stance for any random user on any topic without leveraging shared features like retweets or common mentions. Unlike existing methods, updating context for new users in Tweets2Stance does not require recomputing networks and clusters. Moreover, the unavailability of public datasets used by state-of-the-art methods prevents us from evaluating T2S on those datasets. Furthermore, the lack of publicly available labelled datasets for five-level stance further limits the comparison.

3 Task Definition

The task is to detect the stance A_s^u of a Social Media User u with respect to a socio-political statement (or sentence) s making use of the User's textual content timeline (sequence of textual posts) on the considered social media (e.g., the 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 label *neither disagree nor agree* encompasses both a not expressed and neutral stance. We refer to the *agreement/disagreement level (or label)* as the stance level (or label). The desired ground-of-truth (GoT) is the label G_s^u, which represents the known agreement/disagreement level of User u regarding sentence s. The GoT is solely used for evaluating our proposed framework and optimizing its parameters; no training step is involved. In this

Table 1. Details of the nine elections under study with the total number of tweets. D_i contains i months of tweets. Values between round brackets are the average number of tweets per Party.

Election	no. of parties	no. of statements	D3	D4	D5	D7
Alberta Provincia Election (AB19)	5	18	5119 (1024)	5701 (1140)	6755 (1351)	8502 (1700)
Australian Federal Election (AU19)	3	17	2538 (846)	3130 (1043)	3368 (1123)	4582 (1527)
Canadian Federal Election (CA19)	6	16	7460 (1243)	9284 (1547)	10750 (1792)	12903 (2151)
Great Britain Election v	5	20	9135 (1827)	10783 (2157)	12074 (2415)	15145 (3029)
British Columbia (BC20)	3	20	3560 (1187)	3751 (1250)	3969 (1323)	4448 (1483)
Saskatchewan Provincial Election (SK20)	2	17	1070 (535)	1245 (623)	1557 (779)	1982 (991)
New Foundland and Labrador Provincial Election (NFL21)	3	12	930 (310)	986 (322)	1070 (357)	1293 (431)
New Scotia Provincial Election (NS21)	3	17	859 (286)	1027 (342)	1454 (485)	1727 (579)
Canadian Federal Election (CA21)	6	16	6752 (1125)	7756 (1293)	8734 (1456)	10931 (1822)

study, users are assumed to be Twitter accounts of various political parties from different countries, as described in the subsequent section.

3.1 Data Collection

A Voting Advice Application (VAA) is an established online tool that helps citizens determine their political leaning by comparing their stance on socio-political statements (e.g., "Brexit was an error") with the positions of political parties. To analyze the Parties' stances, we collected data from eight political elections held between 2019 and 2021 *VoteCompass*[2], including the 2019 Great Britain Election *WhoGetsMyVoteUK*[3]. The statements and corresponding Ground-Of-Truths (GoTs) for each election and Party can be found in the provided repository[4].

For our analysis, we collected the Twitter timelines of the competing Parties using the Full-archive search Twitter API. Since some Parties had significantly fewer tweets compared to others, we removed certain Parties from the analysis and focused on those listed in Table 1. D_i represents the collection of tweets posted within i months before the election day (further details in the Methodology section).

[2] https://www.votecompass.com/.
[3] https://www.whogetsmyvoteuk.com/#!/.
[4] https://github.com/marghe943/Tweets2Stance_generalization.

4 Methodology

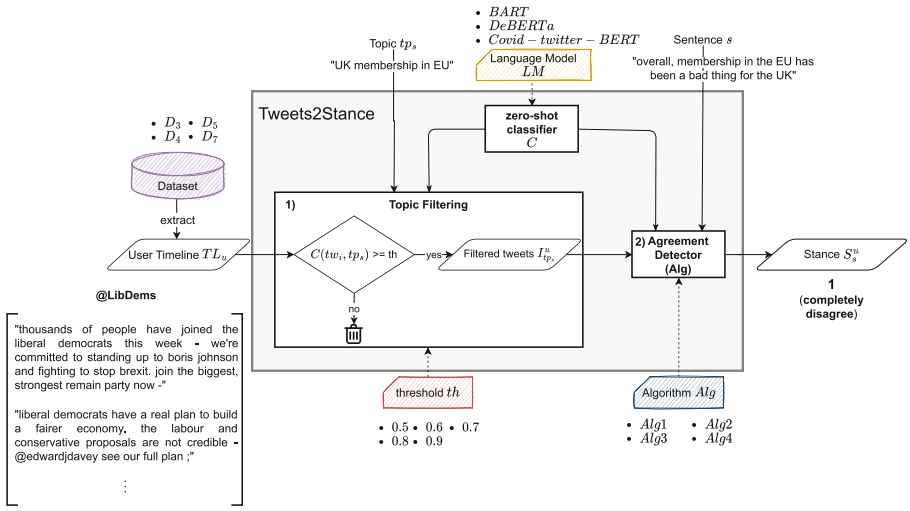


Fig. 1. Our Tweets2Stance framework to compute the agreement/disagreement level A^u_s of User u in regard to sentence s. The inputs are the Twitter timeline TL_u extracted from a certain time-period dataset D_i, the sentence s, the topic tp associated with s, a language model LM, a threshold th and an algorithm Alg. The highlighted components are the parameters that we'll vary during our experiments, as explained in Sect. 4.3.

This section presents the proposed Tweets2Stance (T2S) framework (Fig. 1) to detect the stance A^u_s of a Twitter User u in regard to a sentence s, exploiting its Twitter timeline $TL_u = [tw_1, ..., tw_n]$.

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 $I^u_{tp_s}$ through Topic Filtering, the Agreement Detector module employs the same ZSC to detect the user's agreement/disagreement level.

Figure 1 colour-codes the four parameters of the $T2S$ framework to be tuned:

1. the language model (LM) used for Zero-Shot Classification (ZSC) in the *Topic Filtering* and *Agreement Detector* modules to gauge topic agreement and sentence relevance, respectively,
2. the dataset D_i from which extracting the timeline TL_u,

3. the algorithm *Alg* to use in the *Agreement Detector* module,
4. the threshold *th* to get the in-topic tweets $I^u_{tp_s}$ in the *Topic Filtering* module.

The next subsections provide detailed descriptions of the *Topic Filtering* and *Agreement Detector* modules. We will focus on a specific political scenario where the Twitter accounts of interest are those of the political Parties mentioned in Sect. 3.1, and the User u corresponds to the Party p. 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.

4.1 Topic Filtering

The *Topic Filtering* module extracts the in-topic tweets $I^p_{tp_s}$ from the 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. The module utilizes the ZSC C to retrieve the in-topic tweets $I^p_{tp_s}$ and their corresponding topic scores $T^p_{tp_s}$.

$$I^p_{tp_s} = \{tw_1, ..., tw_m | C(tw_i, tp_s) >= th\} \tag{1}$$

$$T^p_{tp_s} = \{C(tw_i, tp_s) | tw_i \in I^p_{tp_s}\} \tag{2}$$

$C(tw_i, tp_s) \in [0, 1]$ indicates the degree to which tweet tw_i is associated with topic tp_s. The filtering threshold value th was varied to determine the best and optimal parameter set.

4.2 Agreement Detector

The *Agreement Detector* module (Fig. 1 - Module 2) computes the final five-valued label A^p_s through an algorithm $Alg(T^p_{tp_s}, S^p_s)$, defining

$$S^p_s = \{C(tw_i, s) | tw_i \in I^p_{tp_s}\} \tag{3}$$

as the C scores of tweets $I^p_{tp_s}$ 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:

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

$$M2(s) = \begin{cases} 1 & \text{if } s \in [0, 0.25) \\ 2 & \text{if } s \in [0.25, 0.5) \\ 3 & \text{if } s \in [0.5, 0.75) \\ 4 & \text{if } s \in [0.75, 1] \end{cases} \tag{5}$$

where $M1(s)$ ranges from 1 to 5, corresponding to the five agreement/disagreement labels defined in Sect. 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 Sect. 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 (Subsect. 4.2).

The proposed algorithms ordered by complexity are the followings:

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

$$A_s^p = \begin{cases} M1(\frac{\sum_{i=1}^{|I_{tp_s}^p|} s_i \cdot t_i}{\sum_{i=1}^{|I_{tp_s}^p|} s_i}) & \text{if } \mid I_{tp_s}^p \mid \neq 0 \\ 3 & \text{otherwise} \end{cases} \tag{6}$$

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 $s_i \in S_s^p$

$$l_i = M1(s_i) \tag{7}$$

then, A_s^p is

$$A_s^p = \begin{cases} \left\lfloor \frac{\sum_{i=1}^{|I_{tp_s}^p|} l_i}{|I_{tp_s}^p|} \right\rceil & \text{if } \mid I_{tp_s}^p \mid \neq 0 \\ 3 & \text{otherwise} \end{cases} \tag{8}$$

The step of assigning l_i to each tweet $tw_i \in I_{tp_s}^p$ (Eq. 7) aims to achieve a fairer A_s^p. Tweet normalization aids in aggregating the contribution of each tweet (l_i) through standard mean, employing macro aggregation. Macro-metric aggregation is preferred in multi-class classification setups when class imbalance is suspected. In the current context, the values of l_i are unbalanced with respect to sentence s. Typically, if Party p agrees with a sentence, there will be numerous tweets in agreement (many $l_i = 4$ or $l_i = 5$), and few 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 A_s^p is computed with a slight modification. Introducing V_l as the number of voters for the integer label $l \in \{1, 2, 3, 4, 5\}$

$$V_l = |\{l_i : l_i = l\}_{i=1}^{|I_{tp_s}^p|}| \tag{9}$$

where l_i are the labels computed from Eq. 7. Let's define $v = max(V_l)$, then

$$A_s^p = \begin{cases} l & \text{if } |\{l : V_l = v\}| = 1 \quad (10a) \\ \left\lfloor \frac{\sum_{i=1}^{|I_{tp_s}^p|} l_i}{|I_{tp_s}^p|} \right\rceil & \text{if } |\{l : V_l = v\}| > 1 \quad (10b) \\ 3 & \text{otherwise} \quad (10c) \end{cases}$$

where $\lfloor ... \rceil$ is the rounding function. Majority voting (case 10a) potentially contributes more to assigning correct labels than the plain standard mean (case 10b taken from Eq. 8 of $Alg2$) as it effectively accounts for class imbalance.

Algorithm 4 [Alg4] The previous algorithms consider the neutral label $nl = 3$ (*neither disagree, nor agree*) even when $\mid I^p_{tp_s} \mid \neq 0$. However, we explored the scenario where nl is *only* considered when $\mid I^p_{tp_s} \mid = 0$. In such cases, the user might not have taken a position on the sentence s yet, and determining A^p_s based on a single tweet may lack significance. Hence, $Alg4$ extends $Alg3$ with the following modifications:

$$l_i = M2(s_i) \tag{11}$$

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

$$a^p_s = \begin{cases} 3 & \text{if } \mid I^p_{tp_s} \mid < m \\ \text{majority voting (case 10a)} & \\ \text{rounded standard mean (case 10b)} & \end{cases} \tag{12}$$

Here, m is the minimum number of tweets required to activate either the majority voting algorithm or the standard mean. The output labels $\{3, 4\}$ from $M2(s)$ correspond to the final labels *agree* and *completely agree*, and they are mapped to the integer labels 4 and 5 as defined in Sect. 3.

$$A^p_s = \begin{cases} a^p_s & \text{if } a^p_s = 1 \vee a^p_s = 2 \\ a^p_s + 1 & \text{if } a^p_s = 3 \vee a^p_s = 4 \end{cases} \tag{13}$$

4.3 Experiment Settings

To validate the T2S's performance we had to choose i) the set of values for each of the four parameters to tune (the dataset size D_i, the language model LM for ZSC, the algorithm Alg, and the topic-filtering threshold th - Fig. 1), ii) the baselines to which compare T2S, and iii) the evaluation metrics.

T2S Parameters. We chose three to seven months of tweets (D_i), a filtering threshold from 0.5 to 0.9, four algorithms for the *Agreement Detector* module (Sect. 4.2), and three language models for the ZSC. The chosen filtering threshold range was set higher than 0.5 to ensure better agreement between a text and a topic. The language models we adopted are[5]: a) BART-large [12] fine-tuned on the MultiNLI dataset [20], b) DeBERTa-v3-base-mnli-fever (*DeBERTa*), and c) covid-twitter-bert-b1-fever-anli (*Covid-twitter-BERT*). Since the majority of collected tweets are in English, we used English language models. Non-English tweets were translated using Google Translate[6]. Our attempts to employ Multi-Language Models resulted in worse performances [7]. BART and DeBERTa were adapted to handle tweets by removing mentions, hashtags, and emojis, while Covid-twitter-BERT, which is already trained on tweets, was evaluated with and without those structures.

[5] From huggingface.co: a) facebook/bart-large-mnli, b) MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli, c) digitalepidemiologylab/covid-twitter-bert-v2-mnli.

[6] https://github.com/lushan88a/google_trans_new.

Baselines. To validate T2S's abilities, we compared its performance with two bare baselines: (i) **Random**: the final agreement/disagreement label A_s^p is set to a random integer picked from a discrete uniform distribution of $int \in [1, 5]$; (ii) **Assign-highest-value**: A_s^p is always assigned the highest label (*completely agree*) since our datasets are skewed towards the *agree* and *completely agree* values.

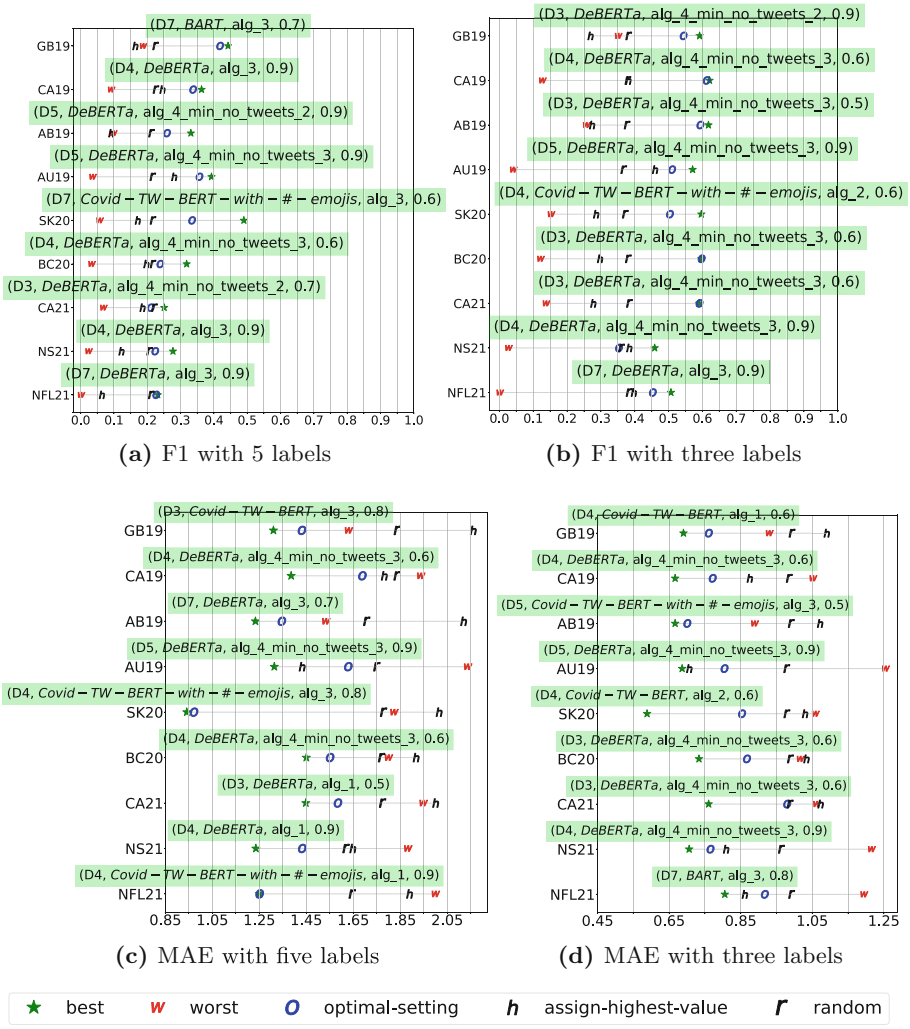
Evaluation Metrics. In assessing the performance of the detection model for this stance detection task, traditional error metrics such as MSE, MAE, R2 Score, Residual Plots, and Macro Averaged Mean Absolute Error are commonly used. However, a custom error metric is needed to account for the varying importance of errors among the stance classes. For example, misclassifying as *agree* instead of *completely disagree* is considered a more acceptable error than misclassifying as *neither disagree, nor agree* instead of *agree*, even though both errors have a magnitude of one. In the absence of such a metric, MAE is the most appropriate choice. Additionally, the F1 weighted score is employed due to the integer nature of the detected labels and the imbalanced distribution of the Ground-of-Truth values among the agreement/disagreement labels.

5 Results and Discussion

Figure 2 shows the F1 and MAE scores over all nine elections respectively. Table 2 indicates the four general optimal settings across the elections by varying the number of labels and the metric considered.

Table 2. The four optimal settings over *no. of labels* and *metric*.

no. of labels	metric	D_i	model	alg	th	avg F1	avg MAE
5	F1	D_3	DeBERTa	alg_4 min no. of tweets: 3	0.9	0.29	1.56
5	MAE	D_4	Covid-twitter-BERT with # and emojis	alg_3	0.9	0.20	1.43
3	F1	D_3	DeBERTa	alg_4 min no. of tweets: 3	0.6	0.53	0.85
3	MAE	D_5	DeBERTa	alg_3	0.9	0.49	0.82



(a) F1 with 5 labels **(b)** F1 with three labels

(c) MAE with five labels **(d)** MAE with three labels

Fig. 2. F1 and MAE scores for all nine elections across baselines (assign-highest-value and random), best and worst setting for each election, and general optimal setting. The green boxes display the best setting for each election. (Color figure online)

5.1 RQ1: What are the Performances and Insights of T2S?

The best setting for each of the nine elections was chosen in two steps: firstly, by varying the algorithm *Alg* and the threshold *th* according to Fig. 1, we selected the D_i and LM with the minimum (maximum) MAE (F1), giving priority to MAE. Then we proceeded to choose the filtering threshold (*th*) and the algorithm (*Alg*) in a similar manner, while keeping the dataset size and language model fixed. The performance results in Fig. 2 demonstrate that T2S is a strong user

stance detection model, surpassing random and assign-highest-value baselines. The best setting for T2S varies across the nine elections, with F1 scores ranging from 0.23 to 0.49 and MAE scores (for five-labelled stance) ranging from 0.94 to 1.45. The selected algorithm alternates between $Alg3$ and $Alg4$, indicating that aggregating the tweet contributions (l_i) yields higher detection precision than directly averaging the sentence scores (s_i). However, the chosen filtering threshold, dataset time period, and language model for ZSC differ significantly across the nine datasets.

These differences can be attributed to two intertwined factors: i) the *diverse topic knowledge* of different language models and ii) the *manner* and *timing* of a user's (political party's) *expression on social media*, which influences T2S stance detections. The choice of the language model is crucial, as models not trained or fine-tuned on the topics in the dataset struggle to assign accurate scores to texts containing those topics. This issue could potentially be addressed by using more advanced models like GPT3 or ChatGPT, which have demonstrated state-of-the-art performance on text stance detection [22]. As for how a user expresses themselves on social media, there are three issues: first, if T2S attempts to detect a user's stance on a socio-political statement they haven't tweeted about but have discussed in a conference, T2S may incorrectly assign the *neither agree, nor disagree* label. Second, if a user tweets about a statement using expressions (e.g., acronyms) that T2S's language model hasn't been trained on, T2S is likely to detect an incorrect stance value. Conversely, if another user tweets about the same statement using more common words, T2S is more likely to detect the correct stance. Lastly, the significant variation in the dataset time periods (D_i) suggests that a user may discuss a certain topic either close to or far from the election date. Therefore, obtaining the user's entire timeline, rather than limiting data collection to specific time periods, could be beneficial. In a previous study, we extensively discussed how the writing style of Italian political parties influences T2S's performance [7]. Similar considerations can be made for the results of the three-labelled stance detection. Noticeably, the F1 scores vary less and are closer (around 0.6) to the best F1 score (0.95) of supervised and semi-supervised text-based techniques in the literature [9,16].

To sum up, although T2S's performance is still distant from state-of-the-art user-based stance detection, we believe it represents a valuable starting point for *addressing* the research *gap* in *unsupervised content-based models* leveraging an advanced Natural Language Processing technique (ZSC) to detect a *five-level stance* of the user on *multiple and diverse targets* (the socio-political statements on different political contexts).

5.2 RQ2: Can T2S Generalize over Diverse Political Contexts?

Figure 2 demonstrates that T2S effectively captures the complex five-level stance across diverse political contexts. However, the optimal settings vary for each election. To identify a potential optimal setting, we calculated the average F1 or MAE performance across all nine election datasets. We selected the four best settings based on the metric (F1 or MAE) and the number of stance values (five

or three). Analyzing the selected optimal settings (Table 2), we observed that the dataset's time period (D_i) and the filtering threshold value (th) have less influence. Effective algorithms involve majority voting and assign the neutral label based on the presence of a minimum number of in-topic tweets. The best-performing language models for ZSC are either fine-tuned on a larger number of hypothesis-premise pairs or pre-trained on tweets. The inclusion or exclusion of leading mentions, hashtags, and emojis does not significantly affect the results.

Overall, the four optimal settings closely approach the best setting for each election, surpassing the performance of baselines and worst settings, with few exceptions. Despite a fixed setting, T2S exhibits considerable performance variation among the nine election datasets, with a maximum variance of approximately 0.2 points for F1 and 0.8 points for MAE. This variability is attributed to how a user (in our case, a political party) expresses its election program on social media platforms such as Twitter.

In summary, although sacrificing some performance, a general framework setting can achieve satisfactory results across different political contexts, consistently outperforming random and assign-highest-value baselines.

5.3 Potential and Limitations

The Tweets2Stance framework was tested on political parties during election campaigns to detect a user's political orientation. It has potential applications in identifying radicalization and extremism, particularly on topics like vaccines or immigration. The framework can also be applied to social media platforms other than Twitter, such as Facebook. However, T2S has limitations when used in unknown scenarios and different topics, such as the need for domain adaptation, as pre-trained models may not perform well when applied to different domains. Data bias is another issue, as pre-trained models may be biased toward certain topics or demographics, leading to inaccurate stance detections and reinforcing biases. Limited vocabulary is a challenge, as pre-trained models may not understand or classify texts with domain-specific words or phrases. Overfitting can occur when fine-tuning on small datasets, resulting in poor performance on new data. Multilingualism is also a limitation, as pre-trained models trained on one language may not work well for another, requiring multilingual training or alternative methods like automatic translation. Finally, T2S faces a major limitation in transfer-learning as it cannot detect stances when users are not discussing a specific socio-political topic. In these cases, T2S detects a middle stance, which may indicate either neutrality or insufficient data for accurate analysis.

6 Conclusions

The main purpose of this work was to devise and probe the specific and generalizing capabilities of *Tweets2Stance*, an *unsupervised stance detection* framework

based on Zero-Shot Learning that detects a *five-labelled* user's *stance* about specific social-political statements by analyzing *content-based analysis* of its Twitter timeline *only*. T2S outperformed the baselines (random and assign-highest-stance-value) on all nine election datasets and demonstrated its ability to generalize across diverse political contexts with a minimum MAE of 0.95 and a maximum F1 of 0.6. However, the scarcity of relevant posts to socio-political statements and the language model's limitations (domain adaptation, data bias, and limited vocabulary) pose constraints on the T2S framework's capabilities.

T2S fills the SOTA gap of unsupervised stance detection models of multiple unrelated targets using content features and innovative language models. While SOTA user-based methods achieve higher F1 scores, they focus on simpler targets (e.g., pro or anti-Trump) with limited stance levels (from two to three); besides, they use a straightforward filtering approach (e.g., excluding tweets mentioning a specific person or organization) or focus on interconnected users through keywords, URLs, and hashtags. In contrast, the T2S framework detects the five-labelled stance of a user on multiple and diverse targets in various contexts, leveraging the unfiltered social media timeline (filtering applied automatically). Lastly, future research could overcome T2S's limitations by employing an advanced language model like GPT-4 or conversational AI like ChatGPT as the ZSC for Topic Filtering and Stance Detector steps, since they showed robust *text* stance detection capabilities.

References

1. Aldayel, A., Magdy, W.: Your stance is exposed! analysing possible factors for stance detection on social media. Proc. ACM Hum.-Comput. Interact. **3**(CSCW), 1–20 (2019)
2. Aldayel, A., Magdy, W.: Stance detection on social media: state of the art and trends. Inf. Process. Manag. **58**(4), 102597 (2021)
3. Biber, D., Finegan, E.: Adverbial stance types in English. Discourse Process. **11**(1), 1–34 (1988)
4. Darwish, K., Stefanov, P., Aupetit, M., Nakov, P.: Unsupervised user stance detection on twitter. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 14, pp. 141–152 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the ACL, Minneapolis, Minnesota, vol. 1, pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423
6. Fraisier, O., Cabanac, G., Pitarch, Y., Besançon, R., Boughanem, M.: Stance classification through proximity-based community detection. In: Proceedings of the 29th on Hypertext and Social Media, HT 2018, pp. 220–228. ACM, New York (2018). https://doi.org/10.1145/3209542.3209549
7. Gambini, M., Fagni, T., Senette, C., Tesconi, M.: Tweets2Stance: users stance detection exploiting zero-shot learning algorithms on tweets. arXiv preprint arXiv:2204.10710 (2022)
8. Garimella, K., De Francisci Morales, G., Gionis, A., Mathioudakis, M.: Reducing controversy by connecting opposing views. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 81–90 (2017)

9. Ghosh, S., Singhania, P., Singh, S., Rudra, K., Ghosh, S.: Stance detection in web and social media: a comparative study. In: Crestani, F., et al. (eds.) CLEF 2019. LNCS, vol. 11696, pp. 75–87. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28577-7_4
10. Gottipati, S., Qiu, M., Yang, L., Zhu, F., Jiang, J.: Predicting user's political party using ideological stances. In: Jatowt, A., et al. (eds.) SocInfo 2013. LNCS, vol. 8238, pp. 177–191. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03260-3_16
11. Küçük, D., Can, F.: Stance detection: a survey. ACM Comput. Surv. (CSUR) **53**(1), 1–37 (2020)
12. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. ACL, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.703
13. Li, Y., Sosea, T., Sawant, A., Nair, A.J., Inkpen, D., Caragea, C.: P-stance: a large dataset for stance detection in political domain. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 2355–2365 (2021)
14. Magdy, W., Darwish, K., Abokhodair, N., Rahimi, A., Baldwin, T.: # isisisnotislam or# deportallmuslims? Predicting unspoken views. In: Proceedings of the 8th ACM Conference on Web Science, pp. 95–106 (2016)
15. Moghaddam, S., Ester, M.: Aspect-based opinion mining from product reviews. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 1184 (2012)
16. Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: SemEval-2016 task 6: detecting stance in tweets. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 31–41 (2016)
17. Rashed, A., Kutlu, M., Darwish, K., Elsayed, T., Bayrak, C.: Embeddings-based clustering for target specific stances: the case of a polarized turkey. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 15, pp. 537–548 (2021)
18. Thonet, T., Cabanac, G., Boughanem, M., Pinel-Sauvagnat, K.: Users are known by the company they keep: topic models for viewpoint discovery in social networks. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 87–96 (2017)
19. Trabelsi, A., Zaiane, O.: Unsupervised model for topic viewpoint discovery in online debates leveraging author interactions. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 12 (2018)
20. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the ACL, vol. 1, pp. 1112–1122. ACL (2018). http://aclweb.org/anthology/N18-1101
21. Yin, W., Hay, J., Roth, D.: Benchmarking zero-shot text classification: datasets, evaluation and entailment approach. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3914–3923. ACL (2019). https://doi.org/10.18653/v1/D19-1404
22. Zhang, B., Ding, D., Jing, L.: How would stance detection techniques evolve after the launch of ChatGPT? arXiv preprint arXiv:2212.14548 (2022)