Machine Learning (2024) 113:7243–7266
https://doi.org/10.1007/s10994-024-06587-y

Evaluating large language models for user stance detection on X (Twitter)

Margherita Gambini[1,2] · Caterina Senette[1] · Tiziano Fagni[1] · Maurizio Tesconi[1]

Received: 7 March 2024 / Revised: 6 May 2024 / Accepted: 14 June 2024 /
Published online: 6 September 2024

Abstract
Current stance detection methods employ topic-aligned data, resulting in many unexplored topics due to insufficient training samples. Large Language Models (LLMs) pre-trained on a vast amount of web data offer a viable solution when training data is unavailable. This work introduces *Tweets2Stance - T2S*, an unsupervised stance detection framework based on zero-shot classification, i.e. leveraging an LLM pre-trained on Natural Language Inference tasks. T2S detects a five-valued user's stance on social-political statements by analyzing their X (Twitter) timeline. The Ground Truth of a user's stance is obtained from Voting Advice Applications (VAAs). Through comprehensive experiments, a T2S's optimal setting was identified for each election. Linguistic limitations related to the language model are further addressed by integrating state-of-the-art LLMs like GPT-4 and Mixtral into the *T2S* framework. 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. 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, particularly when combined with state-of-the-art LLMs, to generalize across different cultural-political contexts.

Keywords User stance detection · LLMs · Unsupervised · Twitter · Text content · Elections · VAA

Editors: Rita P. Ribeiro, Ana Carolina Lorena, Albert Bifet.

✉ Caterina Senette
c.senette@iit.cnr.it

Margherita Gambini
m.gambini@iit.cnr.it

Tiziano Fagni
t.fagni@iit.cnr.it

Maurizio Tesconi
m.tesconi@iit.cnr.it

[1] Institute of Informatics and Telematics (IIT), CNR, Via G. Moruzzi 1, 56100 Pisa, Italy

[2] Department of Information Engineering, University of Pisa, Via G. Caruso, 16, 56122 Pisa, Italy

1 Introduction

Stance detection is a text-mining technique that identifies a user's attitude towards a specific statement (Biber and Finegan, 1988), differentiating it from sentiment analysis, which merely categorizes text as positive, negative, or neutral without focusing on a particular target. This method is particularly relevant on social media, where it is used to detect agreement or disagreement in debates and assess public opinion on various topics, including politics, ideology, and consumer products (Aldayel and Magdy, 2021; Dias and Becker, 2016; Mohammad et al., 2016; Darwish et al., 2017, 2020; Fagni and Cresci, 2022). Stance detection involves *either* identifying the stance expressed in text *or* determining the user's stance towards a target based on their content and context. Targets for stance detection can be single topics, multi-related topics implying a stance towards related targets, or claim-based, evaluating whether a text or user supports a specific claim (Aldayel and Magdy, 2021).

This research focuses on analyzing and quantifying public opinion on diverse issues, specifically addressing *user stance detection across multiple unrelated subjects*. Analyzing social media texts offers key insights into user stances, but assessing data accuracy and reliability is essential (Cresci et al., 2014) due to potential manipulation by automated accounts (Tardelli et al., 2020). However, existing literature suggests diverse approaches that partially leverage text analysis along with user behavioural analysis, encompassing activities like likes, retweets, and network connections (Gottipati et al., 2013; Aldayel et al., 2019). Moreover, user stance detection on unrelated targets presents computational challenges (Aldayel and Magdy, 2021). Content-based stance detection approaches face limitations, including the inherent difficulty of processing natural language, the necessity for vast annotated tweet corpora and language-specific resources, the absence of unsupervised transfer learning for generalization across unrelated targets, and the need to train distinct classifiers for each target. Cutting-edge research often concentrates on either two (support, against) or three levels of stance including the neutral class[1]) and current unsupervised methods relying on clustering techniques in user networks are inadequate for detecting a user's stance on different unrelated targets. To tackle these challenges and concentrate exclusively on a content-based approach, we extend *Tweets2Stance* (T2S) an unsupervised framework for stance detection (Gambini et al., 2023). T2S examines the content of a user's social media timeline (e.g., X-Twitter) using Zero-Shot Learning (ZSL) techniques (Kojima et al., 2022) to identify their stance toward specific socio-political statements (targets), considering five (completely disagree, disagree, neither disagree nor agree, agree, completely agree) or three levels of agreement (disagree, neither disagree nor agree, agree).

We demonstrate T2S's generalizability with F1 and MAE scores on nine diverse political election datasets, collected from competing party Twitter accounts worldwide from 2019 to 2021. In the current implementation, T2S can internally leverage both specialized zero-shot classification models trained on Natural Language Inference (NLI) tasks and generic Large Language Models (LLMs) pre-trained on massive Web data that can be successfully used in zero-shot settings. While the former are computationally efficient models offering good accuracy with minimal hardware requirements, LLMs achieve higher accuracy at the cost of increased computational demands. In this work, we compare both approaches and show that for content-based stance detection methods, LLMs are

[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.

preferable. This is because they process natural language with sophistication, capturing nuances and context more effectively, and seem to excel at capturing stances relative to a target.

The Ground Truth (GT) of a user's stance comes from Voting Advice Applications (VAAs), online tools that help citizens identify their political leanings by comparing their political preferences with party political stances.

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.

RQ3 – *How well do Large Language Models perform in user stance detection tasks without fine-tuning?* Here we explore the applicability of LLMs such as*GPT-4*, the best proprietary model, and Mixtral, the best open model (according to ChatBot Arena's leaderboard).[2]

Contributions

- To the best of our knowledge, we filled the gap of investigating an *unsupervised content-based-only* model leveraging *Large Language Models* to detect a five-level and three-level stance of a user on multiple and diverse targets (the socio-political statements on different political contexts).
- The proposed technique can be easily customized to be adapted to different social media platforms (as data sources) as it only works on users' timelines containing exclusively their texts. Working solely on text makes it independent of other features (network data, user interactions data, etc.) that are usually specific to a certain social media platform. Moreover, it could be used to detect the stance of any user.
- In this study, we focus on a political scenario. The results can serve as an initial step in predicting users' political orientation. However, T2S can be applied to other contexts such as extremism and radicalization, enabling the inference of user radicalization on specific themes (e.g., immigration and vaccines).
- Finally, we provide the collection of labeled datasets used in our experiments, along with the code for testing and evaluating methods, which can be accessed here.[3] This resource may assist other researchers working on unsupervised stance-detection methods at the user level. To the best of our knowledge, there is no labeled dataset available for detecting user stances on multiple unrelated targets across various political contexts, especially one that utilizes a five-level stance classification.

The remainder of this paper is organized as follows: Section 2 discusses related work. In Section 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, Section 6 concludes and suggests future work.

[2] https://chat.lmsys.org/, accessed on 2023-12-28.

[3] https://github.com/tizfa/Tweets2Stance

2 Related work

In the classical definition, Biber and Finegan (1988), 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.

2.1 Content and behavioural features

Rashed et al. (2021) focused on user-based stance detection using content features alone. They employed Google's Multilingual Universal Sentence Encoder (MUSE) and a pretrained 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) successfully clustered users based on feature similarities such as retweets, common hashtags, and retweeted accounts; Aldayel et al. (2019) achieved an F1 score of 0.72 by leveraging users' online behaviour cues; et al. (2017) Thonet et al. (2017) considered both text and social interactions to uncover topics, user viewpoints, and discourse; Magdy et al. (2016) 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).[4] Lastly, Fraisier et al. (2018) used content-based and social-based proximities in a multi-layer graph, achieving an F1 score of 0.95.

2.2 Supervised and unsupervised learning

Stance detection techniques using supervised learning rely on large annotated datasets (Mohammad et al., 2016). User-based stance detection has received less attention in these competitions, but notable studies include (Aldayel et al., 2019) and Magdy et al. (2016). 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 Fraisier et al. (2018) applied a proximity-based two-level stance detector to different datasets related to political events and gun control. More recently (Ghosh et al., 2019; Küçük and Can, 2020), the trend in language processing for stance-detection tasks relies on language representation models (e.g., BERT (Devlin et al., 2019)) pre-trained on large unannotated corpora and *fine-tuned* on labeled and domain-specific datasets (Devlin et al., 2019; Yin et al., 2019). The work of Devlin et al. (2019) demonstrated how BERT led to considerable performance improvements for NLP tasks such as sentiment analysis. Ghosh et al. (2019) 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.

[4] See footnote 1.

2.3 Stance detection and large language models

Recently, researchers have begun exploring the potential of Large Language Models (LLMs) in stance detection tasks, yielding contrasting results. Zhang et al. (2022) utilized ChatGPT for text-based stance detection, matching or surpassing state-of-the-art results on SemEval-2016 (Mohammad et al., 2016) and PStance (Li et al., 2021) datasets. Aiyappa et al. (2023) found ChatGPT improved performance but warned of potential reliability issues due to data contamination risks. Cruickshank et al. (2023) explored the efficacy of various LLMs with different prompt schemes and increasing contextual information on stance detection, finding that while LLMs show promise, they currently do not outperform traditional supervised learning models, highlighting the need for further research to improve LLMs' stance detection abilities.

In general, it remains uncertain whether LLMs, particularly when incorporating prompt engineering and without utilizing fine-tuning on labeled data, can effectively carry out the task of stance classification. Our work aims to contribute to the literature with twofold results. On the one hand, we present a contribution in terms of unsupervised solutions that are still underrepresented in the literature. Existing unsupervised methods, such as Darwish et al. (2020), Trabelsi et al. (2018), Fraisier et al. (2018), and Fagni and Cresci (2022), rely on standard linguistic features like n-grams, keyword counts, and content embeddings. Recognizing this lack and the increasing use of pre-trained models in stance detection, we propose *Tweets2Stance* (*T2S*), an unsupervised framework based on zero-shot classification, hence leveraging an LLM: it effectively detects the stance of multiple unrelated targets without the need for separate models per target; unlike transfer-learning approaches that require training, our framework is entirely unsupervised. On the other hand, we assess the potential of using "raw" LLMs in our framework, i.e. by replacing each module of T2S with an advanced LLM pre-trained on a large corpus of web data.

Comparing the T2S framework to state-of-the-art user-based stance detection methods presents several challenges. Firstly, the method by Rashed et al. (2021) filtered tweets by selecting mentions of specific targets, which is incompatible with our work as our topic lacks a well-defined person or organization. Other methods rely on timelines of users connected through specific keywords, while T2S aims to infer the stance of 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 labeled 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 X-Twitter timeline). Some illustrative examples are in Appendix C.

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 GT is the label G_s^u, which represents the known agreement/disagreement level of User u regarding sentence s. The GT is solely used for evaluating our proposed framework and optimizing its parameters; no training step is involved. In this study, users are assumed to be X (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 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 nine political elections held between 2019 and 2021 in *VoteCompass*,[5] including the 2019 Great Britain Election *WhoGetsMyVoteUK*.[6] The statements and corresponding Ground Truths (GTs) for each election and Party can be found in the provided repository.[7] For our analysis, we collected the X (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).

4 Methodology

In this Section we present the proposed Tweets2Stance (T2S) framework (Fig. 1) to detect the stance A_s^u of a X (Twitter) User u regarding a sentence s, exploiting its X (Twitter) timeline $TL_u = [tw_1, ..., tw_n]$. In the first part (Sect. 4.1), we describe T2S leveraging a Zero-Shot Classifier (ZSC). In the second part (Sect. 4.2), we detail the usage of state-of-the-art Large Language Models (LLM) in T2S.

4.1 Tweets2Stance - T2S

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. Yin et al. (2019) defines ZSC as a Language Model (e.g., BART or DeBERTa) pre-trained on Natural Language Inference tasks that, given a text and a set of labels (e.g., topics), assigns a classification probability score to each label. The higher the score assigned to a label, the higher the likelihood that the input text pertains to that specific label. 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. Figure 1 colour-codes the four parameters of the *T2S* framework to be tuned: i) the language model (*LM*) used for zero-shot classification (ZSC) in the *Topic*

[5] https://www.votecompass.com/

[6] https://www.whogetsmyvoteuk.com/#!/

[7] https://github.com/marghe943/Tweets2Stance_generalization

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

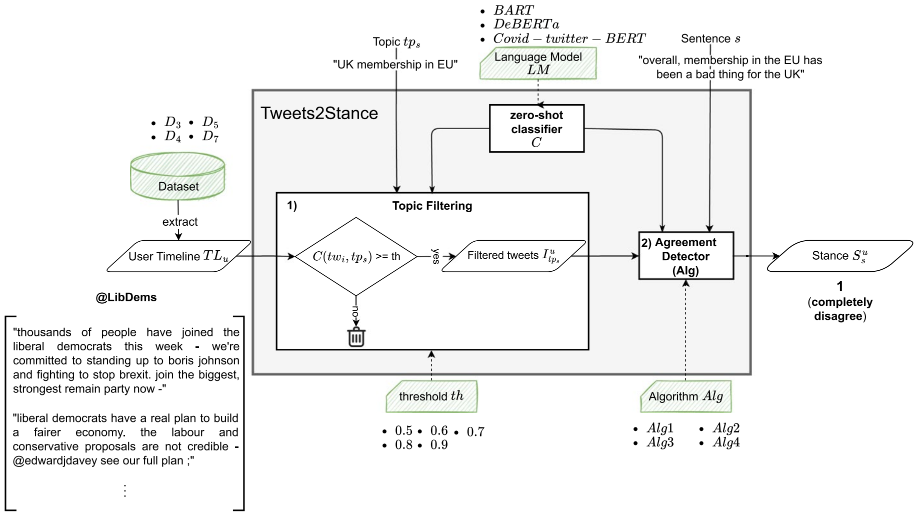
Election	No. of parties	No. of statements	D3	D4	D5	D7
Alberta Provincia Election (AB19)	5	18	5119 (1024)	5701 (1140)	6755 (1351)	8502 (1700)
Australian Federal Election (AU19)	3	17	2538 (846)	3130 (1043)	3368 (1123)	4582 (1527)
Canadian Federal Election (CA19)	6	16	7460 (1243)	9284 (1547)	10750 (1792)	12903 (2151)
Great Britain Election (GB19)	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)

Filtering and *Agreement Detector* modules to gauge topic agreement and sentence relevance, respectively; ii) the dataset D_i from which extracting the timeline TL_u; iii) the algorithm Alg to use in the *Agreement Detector* module; iv) 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 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 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.

4.1.1 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



"thousands of people have joined the liberal democrats this week - we're committed to standing up to boris johnson and fighting to stop brexit. join the biggest, strongest remain party now -"

"liberal democrats have a real plan to build a fairer economy. the labour and conservative proposals are not credible - @edwardjdavey see our full plan ;"

⋮

Fig. 1 Our Tweets2Stance framework to compute the agreement/disagreement level A^u_s of User u in regard to sentence s *leveraging a Zero-Shot Classifier*. 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 (in green) are the parameters that we'll vary during our experiments, as explained in Sect. 4.3

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.

4.1.2 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:

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

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

where $M1(\mathfrak{s})$ ranges from 1 to 5, corresponding to the five agreement/disagreement labels defined in Sect. 3. Similarly, $M2(\mathfrak{s})$ ranges from 1 to 4, representing an intermediate agreement/disagreement scale. Specifically, $M2(\mathfrak{s}) = \{1, 2\}$ has the same meaning as in Sect. 3, while $M2(\mathfrak{s}) = 3$ indicates agreement and $M2(\mathfrak{s}) = 4$ represents complete agreement. The rationale behind this intermediate mapping is explained in Algorithm 4 (subsection 4.1.2). The proposed algorithms ordered by complexity are the following:

Algorithm 1 **[Alg1]** The label A^p_s is computed as

$$A^p_s = \begin{cases} M1\left(\frac{\sum_{i=1}^{|I^p_{tp_s}|} \mathfrak{s}_i \cdot t_i}{\sum_{i=1}^{|I^p_{tp_s}|} \mathfrak{s}_i} \right) & \text{if } | I^p_{tp_s} | \neq 0 \\ 3 & \text{otherwise} \end{cases} \tag{6}$$

where $\mathfrak{s}_i \in S^p_{tp_s}$ and $t_i \in T^p_{tp_s}$.

Algorithm 2 **[Alg2]** First, it maps each tweet $tw_i \in I^p_{tp_s}$ into the label $l_i \in \{1, 2, 3, 4, 5\}$ using its sentence score $\mathfrak{s}_i \in S^p_s$

$$l_i = M1(\mathfrak{s}_i) \tag{7}$$

then, A^p_s is

$$A^p_s = \begin{cases} \left\lfloor \frac{\sum_{i=1}^{|I^p_{tp_s}|} l_i}{|I^p_{tp_s}|} \right\rceil & \text{if } | I^p_{tp_s} | \neq 0 \\ 3 & \text{otherwise} \end{cases} \tag{8}$$

The step of assigning l_i to each tweet $tw_i \in I^p_{tp_s}$ (Eq. 7) aims to achieve a fairer A^p_s. 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^p_s 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

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

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 $| I_{tp_s}^p | \neq 0$. However, we explored the scenario where nl is *only* considered when $| I_{tp_s}^p | = 0$. In such cases, the user might not have taken a position on the sentence s yet, and determining A_s^p based on a single tweet may lack significance. Hence, $Alg4$ extends $Alg3$ with the following modifications:

$$l_i = M2(\mathfrak{s}_i) \tag{10}$$

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

$$a_s^p = \begin{cases} 3 & \text{if } | I_{tp_s}^p | < m \\ \text{majority voting (case 10a)} & \\ \text{rounded standard mean (case 10b)} & \end{cases} \tag{11}$$

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.

$$A_s^p = \begin{cases} a_s^p & \text{if} a_s^p = 1 \lor a_s^p = 2 \\ a_s^p + 1 & \text{if} a_s^p = 3 \lor a_s^p = 4 \end{cases} \tag{12}$$

4.2 Enhancing T2S efficiency: leveraging state-of-the-art LLMs

Figure 2 describes how our T2S framework can leverage either a ZSC or an LLM as the Topic Detector component. The Topic Detector employing zero-shot classification (ZSC) is detailed in Sect. 4.1, with the application of an LLM is explained subsequently. Enhancing Transformer-based language models by increasing their size and refining their pre-training has led to *state-of-the-art LLMs* (such as GPT-4 and Mixtral) achieving higher proficiency in processing and adhering to instructions, demonstrating superior performance in understanding and completing tasks with increased precision and efficiency (Kasneci et al., 2023). Hence, we also assess our T2S framework's capabilities by implementing either one of the two modules –*Topic Filtering* and *Agreement Detector* – and then both, with an advanced LLM, obtaining the following three configurations:

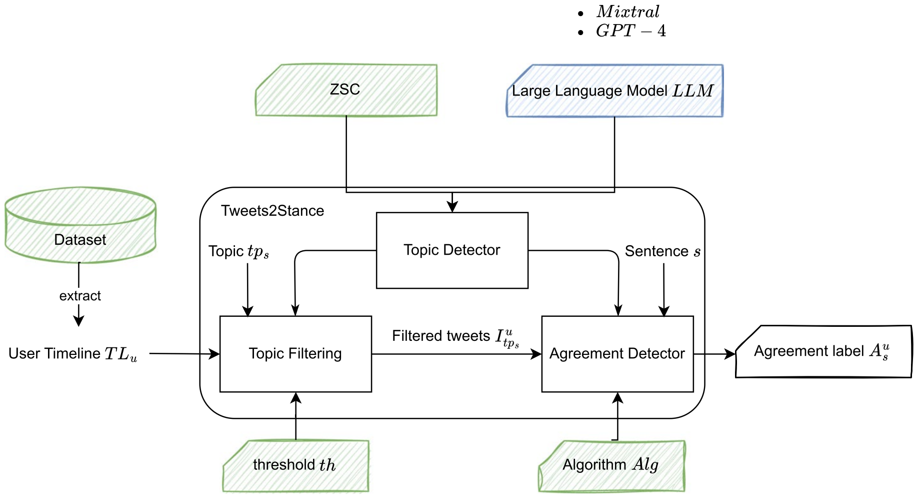


Fig. 2 T2S's architecture leveraging either a ZSC model or a generic LLM in the zero-shot setting. The parameters required for T2S using a ZSC model are marked in green. Implementing the Topic Filtering and Agreement Detector modules via an LLM involves adequately prompting the LLM with the content (tweets, topics, and sentences) we possess

1. *LLM Topic Filtering - T2S Agreement Detector* First and foremost, the LLM is used to get the filtered tweets $I^p_{tp_s}$. For each tweet, we provide the LLM with a list of topics related to sentences from the political election involving Party p, requesting it to identify the relevant topics, if any. Section 4.3.3 describes the used *prompt_filtering*. Then, the T2S Agreement Detector configured with the optimal settings (see later on Table 4 with the highest F1 score for five and three labels) is used on those filtered tweets to compute the five-level stance A^p_s. Note that the sentence scores needed for *Alg4* (the best algorithm in the optimal setting) are the ones computed using the ZSC of T2S.
2. *T2S Topic Filtering - LLM Agreement Detector* First and foremost, the T2S Topic Filtering is used to get the filtered tweets $I^p_{tp_s}$ using the ZSC topic scores and the threshold th from the T2S's optimal setting. Then, the LLM is used to compute the five-level stance A^p_s by prompting it with precise instructions and the set of in-topic tweets, as shown in Sect. 4.3.3. Briefly, it detects and outputs the stance found in the provided tweets. Differently from the T2S's Agreement Detector, if the set of in-topic tweets is empty, all tweets for that political election and Party p are used.
3. *LLM Topic Filtering - LLM Agreement Detector* The LLM replaces both T2S's modules. Hence, it is leveraged to both get the filtered tweets $I^p_{tp_s}$ for the current Party p and sentence s, and then to compute the five-level stance A^p_s given the set of in-topic tweets. Differently from the T2S's Agreement Detector, if the set of in-topic tweets is empty, all tweets for that political election and Party p are used.

Each experiment involving the LLM as the Agreement Detector is conducted for the three-level stance too ('*disagree*', '*neither disagree nor agree*', '*agree*'), prompting the LLM with a three-level stance instead of five. Last but not least, note that LLMs like GPT-4 and Mixtral are proficient in handling multilingual text and do not require translation into English (Jiang et al., 2024).

4.3 Experiment settings

This work began before GPT-3's introduction and the subsequent emergence of Large Language Models. As a consequence, we offer two versions of T2S: one utilizing ZSC and another that employs an LLM. To evaluate the T2S's performance we have to choose the baselines to which to compare T2S, and the evaluation metrics. Furthermore, the T2S version with ZSC requires choosing 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). Conversely, the T2S version employing an LLM requires configuring the parameters of the cutting-edge LLMs to be tested (GPT-4 and Mixtral) and selecting appropriate prompts for the experiments.

4.3.1 Baselines and evaluation metrics

To validate T2S's abilities, we compare 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.

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 (Chai and Draxler, 2014) is the most appropriate choice. Additionally, the F1 weighted score (Sebastiani, 2002) is employed due to the integer nature of the detected labels and the imbalanced distribution of the GT values among the agreement/disagreement labels.

4.3.2 Parameters for T2S with ZSC

We choose 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. 4.1.2), and three language models for the ZSC. The chosen filtering threshold range is set higher than 0.5 to ensure better agreement between a text and a topic. The language models that we adopt are[8]: a) BART-large (Lewis et al., 2020) fine-tuned on the MultiNLI dataset (Williams et al., 2018), 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 use English language models. Non-English tweets are translated using Google Translate.[9] Our attempts to employ Multi-Language Models resulted in worse performances Gambini et al. (2022). BART and DeBERTa are adapted to handle tweets by removing mentions, hashtags, and

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

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

Table 2 Set-up for GPT-4 and Mixtral. The not-mentioned parameters are the default ones of the Azure Microsoft API for GPT-4 and the llama_ccp library for Mixtral

	GPT-4	**Mixtral**
Model	GPT-4-Turbo	Mixtral-8x7B Instruct
Version	–	mixtral-8x7b-instruct-v0.1.Q4_K_M
Size (no. of parameters)	$[100B - 200B]$	$46.7B$
Temperature	0.2	0.0
Top_p	1.0	0.95
Max Context Length	128, 000 tokens	32, 768 tokens
Max Generated Sequence Length	512 tokens	512 tokens
GPU	–	NVIDIA RTX 6000 Ada Generation (32GB VRAM)
API	Microsoft Azure OpenAI Service	–
(Python) Library	–	llama_cpp

emojis, while Covid-twitter-BERT, which is already trained on tweets, is evaluated with and without those structures.

4.3.3 State-of-the-art LLMs set-up and prompting strategies

We select two of the most well-known and high-performing LLMs to incorporate into the T2S framework to solve the Stance Detection task: *GPT-4*, the best proprietary model, and Mixtral, the best open source model. Table 2 summarizes the set-up information for both GPT-4 and Mixtral. As for the latter, we choose the 8x7B Instruct version rather than the plain Mixtral 8x7B, as *Instruct* has been optimised through supervised fine-tuning for careful instruction following[10]; besides, the GGML_TYPE_Q4_K quantization method[11] is chosen as it is described as having medium and balanced quality.[12]

The GPT-4's sampling parameters (*temperature* and *top_p*) are determined based on tests conducted by the OpenAI community, as documented in the thread[13] on mastering temperature and top_p in the ChatGPT API. The recommended approach is to set either the temperature or top_p while keeping the other parameter at its default value (1.0) to achieve optimal performance. For our analysis, we select the 'Data Analysis Scripting' setting with a temperature of 0.2. In the case of Mixtral, we opt for a temperature of 0.0 to ensure deterministic results, while leaving the top_p parameter at its default value (0.95).

Prompts

Considering appropriate prompting schemes is the scope of Prompt engineering, a new field arising with the emergence of LLMs, that focuses on optimizing inputs and prompts to maximize model outputs (White et al., 2023). To improve the interaction with the selected LLMs we use the typical Role-Task-Requirements-Instructions (RTRI) prompt

[10] https://mistral.ai/news/mixtral-of-experts/

[11] "type-1" 4-bit quantization in super-blocks containing 8 blocks, each block having 32 weights. Scales and mins are quantized with 6 bits. This ends up using 4.5 bits-per-word.

[12] https://huggingface.co/TheBloke/Mixtral-8x7B-Instruct-v0.1-GGUF

[13] https://community.openai.com/t/cheat-sheet-mastering-temperature-and-top-p-in-chatgpt-api/172683

structure that Open AI supposedly used[14] to organize the interaction clearly and effectively: initially, the LLM is assigned a specific *role* to embody. Following this, the *task* is clearly outlined, providing a detailed description of the objectives to be achieved; subsequently, the *requirements* section specifies the desired characteristics of the output, ensuring clarity on the expected results; finally, the *instructions* segment offers direct guidance on the actions the LLM should undertake to fulfill the prompt successfully. Moreover, the Microsoft Copilot Team highlights the importance of adopting a polite and supportive tone.[15] Utilizing polite expressions, including ’please’ and ’thank you,’ fosters a dialogue characterized by respect, civility, and constructiveness. Conversely, harsh or inflammatory words can provoke unwanted behaviours.

Topic filtering (**Prompt** 1) - This prompt starts by asking the LLM to write again the list of topics we provide and then the first mentioned topic. This strategy is adopted in response to *Mixtral*’s issue with retaining all topics in the list, aiming to enhance memory recall and focus. During topic filtering, the prompts’ length stayed within the LLM’s maximum context length.

Stance detection (**Prompt** 2) - We ensure that the combined length of the prompt and the output sequence do not exceed the maximum context size for the current LLM. If the length of the prompt exceeds this maximum context size, we selectively remove the oldest tweets from the block of tweets until an appropriate size is achieved.

5 Results and discussion

Figure 3 shows the F1 and MAE scores over all nine elections respectively. Table 4 indicates the four general optimal settings across the elections by varying the number of labels and the metric considered.

5.1 RQ1: What are the performances and insights of T2S?

The optimal settings for T2S in nine election datasets were identified through a two-step process focusing on minimizing MAE and maximizing F1, prioritizing MAE. Initially, settings were chosen based on algorithm (*Alg*) and threshold (*th*) adjustments. The performance of T2S (Fig. 3) was found to be superior to baselines, with F1 scores between 0.23 and 0.49 and MAE scores from 0.94 to 1.45. The preferred algorithms were *Alg*3 and *Alg*4, showing that aggregating tweet contributions results in better detection accuracy than averaging sentence scores. However, the optimal filtering thresholds, dataset time periods, and language models for the ZSC varied significantly across 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. As shown by the experiments in response to RQ3 - Sect. 5.3, this issue is mitigated by using GPT-4 model within the T2S framework.

[14] https://www.linkedin.com/pulse/prompt-engineering-educators-making-generative-ai-work-danny-liu

[15] https://www.microsoft.com/en-us/worklab/why-using-a-polite-tone-with-ai-matters

Regarding user expression, there are three challenges for T2S: 1) detecting stance on unmentioned topics from conferences, possibly leading to *neither agree, nor disagree* assignments; 2) potential errors with unknown language model expressions, and 3) dataset time variability affecting topic relevance. Obtaining the user's full timeline, not just limited periods, could help, as discussed in our earlier study (Gambini et al., 2022). For three-level stance detection, F1 scores are notably closer (around 0.6) to top literature scores (Ghosh et al., 2019; Mohammad et al., 2016).

5.2 RQ2: Can T2S generalize over diverse political contexts?

To determine an optimal setting for T2S across nine election datasets, average F1 and MAE performances were calculated, leading to the selection of the top four settings based on these metrics for either three or five stance values (Table 4). The analysis revealed that the dataset's time period D_i and the filtering threshold *th* have a minimal impact on performance. Effective settings often utilize majority voting and neutral label assignment based on the presence of a certain number of relevant tweets. The most successful language models for zero-shot classification (ZSC) are either fine-tuned with numerous hypothesis-premise pairs or pre-trained on tweets. The presence of mentions, hashtags, and emojis showed little effect on outcomes. These four settings nearly matched the best individual election performances, outdoing baselines and lesser settings, though T2S's performance still varies significantly across datasets, with up to 0.2 points in F1 and 0.8 in MAE differences. This variation is linked to the unique ways political parties communicate their platforms on social media.

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 RQ3: How well do LLMs perform in user stance detection tasks without fine-tuning?

Table 3 summarizes the results for each combination of the two T2S's steps (*Topic Filtering* and *Agreement Detection*) implemented either with ZSC or by leveraging an advanced LLM (GPT-4 or Mixtral). The performance of the original T2S is related to the optimal setting over the F1 score (see the first and third rows of Table 4).

Implementing the Agreement Detector with an advanced LLM yields a superior average MAE score compared to the original T2S, indicating closer alignment with Ground Truth stances. However, Mixtral appears to lag in correctly identifying stances when using in-topic tweets filtered with the original T2S. Conversely, GPT-4 demonstrates stronger performance in five-level stance classification; yet, it encounters challenges in three-level stance tasks, often misinterpreting positive stances ('agree', 'completely agree') as negative ('disagree', 'completely disagree'), and vice versa, alongside mislabeling both as neutral stances more frequently. Moreover, replacing the Topic Filtering with an advanced LLM yields comparable or lower performance than the original T2S. The filtering using Mixtral seems more aligned with the T2S's Agreement Detector (using DeBERTa as ZSC). Even in this setting, the average MAE is lower compared to T2S; however, the significantly lower F1 scores on the three-level stance (both derived and computed) may indicate that this configuration mislabeled the positive/negative stance with the neutral one more frequently. Last but not least, replacing both the Topic Filtering and the Agreement Detector

Table 3 Performance of T2S by replacing each module (*Topic Filtering* and *Agreement Detector*) with either GPT-4 or Mixtral. *Derived* implies the 3-labelled stance is derived from the computed 5-labelled stance, whereas *computed* means that the 3-labelled stance is computed from scratch. The *avg F1|MAE* is computed by averaging over the *F1|MAE* of the nine political elections

Method	**avg F1**			**avg MAE**		
	5 labels	**3 labels**		**5 labels**	**3 labels**	
		derived	**computed**		**derived**	**computed**
original T2S	0.29	0.53	–	1.56	0.85	–
T2S Topic Filtering + Agreement Detector w/ Mixtral	0.19	0.47	0.46	1.23	0.69	0.67
T2S Topic Filtering + Agreement Detector w/ GPT-4	0.34	0.50	0.50	1.07	0.63	0.63
Topic Filtering w/ Mixtral + T2S Agreement Detector	0.28	0.47	–	1.54	0.89	–
Topic Filtering w/ GPT-4 + T2S Agreement Detector	0.22	0.37	–	1.39	0.82	–
Topic Filtering and Agreement Detector w/Mixtral	0.24	0.57	0.58	1.13	0.61	0.59
Topic Filtering and Agreement Detector w/GPT-4	**0.41**	**0.62**	**0.62**	**0.94**	**0.53**	**0.54**

with an advanced LLM brought the most interesting results. Mixtral still lacks stance detection capabilities on the five-level stance, even though it may be a good Topic Filterer as shown in the "Topic Filtering w/Mixtral + T2S Agreement Detector" experiment. Nonetheless, it now presents a good stance detection ability in recognizing a three-level stance, as the MAE scores exhibit. Furthermore, GPT-4 can greatly increase/decrease the F1/MAE score of five-level stance detection by 0.12/0.64 points flat and the three-level stance by 0.09/0.32. Integrating advanced Large Language Models (LLMs) such as GPT-4 and Mixtral significantly improves the T2S framework. Proprietary LLMs like GPT-4 enhance the framework's accuracy, while open-source LLMs like Mixtral also prove to be effective, particularly in distinguishing between three levels of stance.

5.4 Potential and limitations

Tweets2Stance detects political orientation in election campaigns, aiding in spotting radicalization on topics like vaccines or immigration. While it extends beyond X(Twitter) to platforms like Facebook, adapting to new contexts poses some challenges. Domain adaptation is essential due to varied topics, and biased pre-trained models can skew results. Limited vocabulary raises issues with domain-specific terms, and overfitting on small datasets reduces generalizability. Moreover, multilingualism adds complexity, requiring multilingual training or translation methods. Experiments have shown that transitioning from zero-shot learning (ZSL) models to Large Language Models (LLMs) significantly enhances the T2S framework's performance. LLMs excel in their intricate understanding of texts, adeptly processing diverse writing styles, including slang and ironic expressions. This intrinsic capability substantially improves T2S's filtering steps and evaluation agreement on topic-related texts, leading to a significant reduction in evaluation errors for individual texts and across a user's entire timeline. A critical aspect to consider is that an LLM's prediction quality depends on its complexity, which can have economic implications (e.g., GPT-4 is a paid service) and computational impacts. While ZSL models offer acceptable accuracy levels with the advantage of being freely available and less computationally costly, LLM models provide better accuracy and superior multilingual capabilities but generally entail higher implementation costs. However, the T2S framework offers flexibility in choosing the best solution-whether a ZSL or an LLM model-based on the needs and context of the application.

6 Conclusion

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-level* user's *stance* about specific social-political statements by analyzing *content-based analysis* of its X (Twitter) timeline *only*. The T2S version leveraging a Zero-Shot Classifier 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 an average MAE of 1.56 and an average F1 of 0.29. 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. To address these limitations, we implemented either the Topic Filtering or Agreement Detection module, and then both, utilizing a state-of-the-art LLM such as GPT-4 or Mixtral: implementing both modules with an LLM enhances T2S's capabilities from an *average* F1|MAE score of 0.29/1.56 to 0.41/0.94. 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-level stance of a user on multiple and diverse targets in various contexts, leveraging the unfiltered social media timeline (filtering applied automatically).

Appendix A Prompts

Prompt 1 Filtering with LLM. <topic_i > must be replaced with the i^{th} topic. <tweet> must be replaced with the tweet's text.

```
Write again the topics I'm providing you:
    `"topics"` : [
    "<topic1 >",
    "<topic2 >",
    ...
]

Then, write the last topic in the list.

Now you are a helpful AI topic labeler, capable of processing a text `"text_to_analyze"` the `"topics"` above. Your job is to semantically assign zero, one or more topics from `"topics"` to a text `"text_to_analyze"`. You must meet ALL these requirements ONLY:
- You must always respond in JSON format containing 2 fields, "text" and "selected_topics". "text" is the analyzed text, "selected_topics" is the set of topics from `"topics"` which are correlated with the `"text_to_analyze"`.
- Your output must include only the JSON data. Do not add any explanation also in case no matching topics can be found.
- You must only assign topics listed on '"topics"', exactly as specified in the list.
- You mustn't assign topics mentioned in the text but not in the `"topics"` list! For example, if the text mentions a topic <topic>which you cannot find in the `"topics"` list, you must not write it in "selected_topics".
- If a topic is not explicitly listed but is implied in the `"text_to_analyze"`, you don't put it inside the "selected_topics" list.
You are capable. You can reason step by step, check that every requirement is met and therefore answer correctly.

`"text_to_analyze"`: "<tweet>"

```

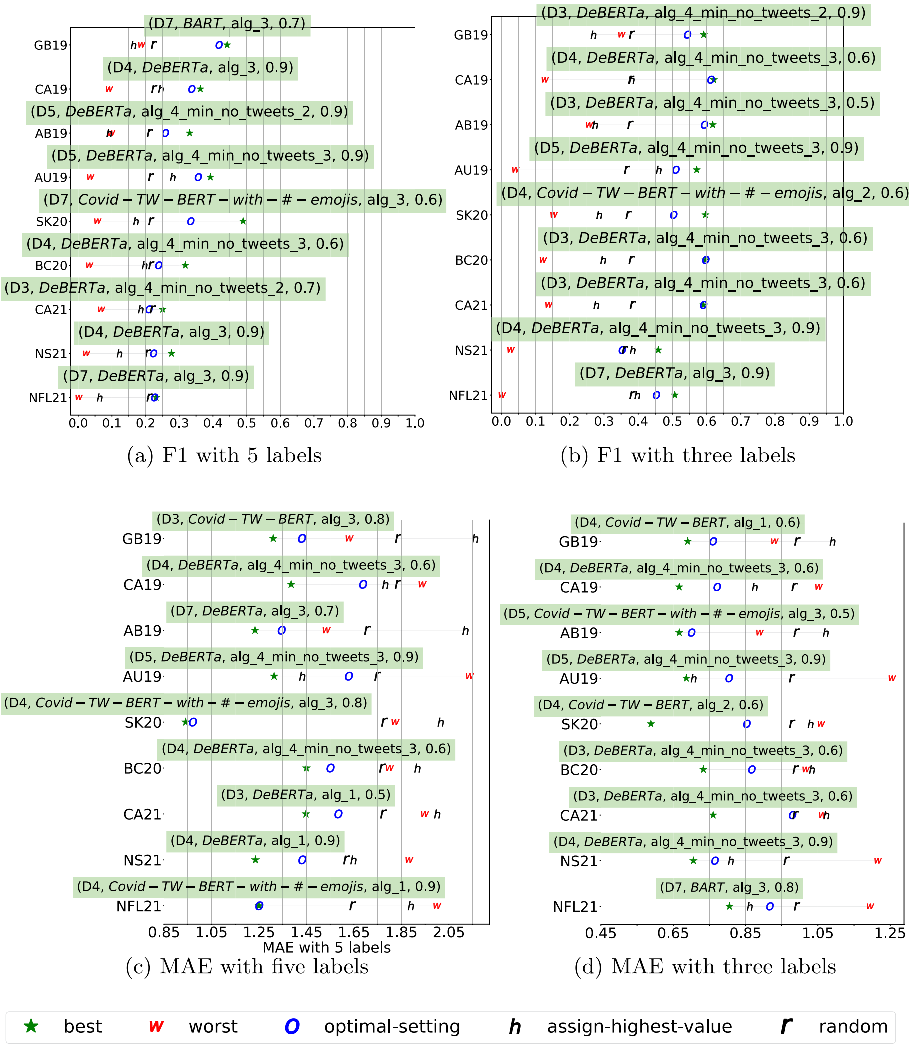
```
Tweets by an unknown Twitter account are submitted in a block delimited by
'=== TWEETS BEGIN ===' and '=== TWEETS END ==='. Each tweet is
separated by the '∼ | ∼' separator. Tweets may include emojis in unicode (e.g,
<Folded_Hands_Emoji>). Also, tweets could be in a language other than English
(e.g., Italian).

=== TWEETS BEGIN ===
< tweet1 >∼ | ∼< tweet2 >∼ | ∼ ... ∼ | ∼< tweetN >
=== TWEETS END ===

Your job is to assess the stance of the Twitter account in response to the statement
"the federal government should impose a carbon tax on provinces without a tax of
their own" leveraging what the account wrote (consider all tweets you've just read).
The stance can assume only these <five|three> values: <'completely disagree',
'disagree', 'neither disagree nor agree', 'agree', and 'completely agree'|'disagree',
'neither disagree nor agree', 'agree'>. The 'neither disagree nor agree' case refers
to cases where there are tweets about the topic but the stance is neutral.

The stance for the statement must be in JSON format like this:

{
    "stance": <the stance>,
    "statement": <the statement>
}

```

Prompt 2 Stance Detection with LLM. The notation $<A|B>$ signifies that one should select either option A or option B to replace $<A|B>$. <Folded_Hands_Emoji> must be replaced by the real emoji. Every $< tweet1 >$ must be replaced with the text from the i^{th} tweet.

Appendix B

RQ1-RQ2 results

See Fig. 3 and Table 4



(a) F1 with 5 labels

(b) F1 with three labels

(c) MAE with five labels

(d) MAE with three labels

Fig. 3 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

Table 4 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

Appendix C

Illustrative examples of stance detection from tweets

See Table 5

Table 5 Sample of 3 tweets extracted from the @LibDem timeline (14k tweets). For each tweet *T2S* (set-up: *threshold th =0.6; Algorithm =alg*$_4$) calculates a score regarding the topic *"no deal for brexit"* and a score regarding the target sentence *"the UK should leave the EU without a deal"*. These scores are used by the framework to weigh their relevance to the final predicted user stance. In this example, tweets T1 and T3 pass the filter to reach the voting algorithm that predicts the stance of the @LibDem account relative to the target sentence. The framework considers these scores calculated for the user's entire timeline to predict the final stance of the @LibDem Party related to the target sentence. In this specific case, the predicted stance is *"completely disagree"*, which aligns with the GT

#	tweet	topic score	sentence score
T1	"britain deserves better than brexit. that's why the first part of our plan for the future is to stop this brexit mess and invest the £50 billion remain bonus back into our nhs, schools, and care services. back our campaign now"	0.857	0.012
T2	"another labour mp, rebecca long-bailey, says labour may campaign to leave with their own brexit in a people's vote. only the liberal democrats are clear that they want to. back our campaign now "	0.001	–
T3	"liberal democrats are ready to take our message to the country. our plan for an election on december 9th, once no deal is taken off the table, would end this parliamentary deadlock and give the british people the chance to vote for a brighter future in the eu. "	0.823	0.615

Funding Open access funding provided by Consiglio Nazionale Delle Ricerche (CNR) within the CRUI-CARE Agreement. This work is supported by (i) Project: "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" -Prot. IR0000013 - Avviso n. 3264 del 28/12/2021; (ii) Project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

Aiyappa, R., An, J., Kwak, H., & Ahn, Y.-Y. (2023). Can we trust the evaluation on chatgpt? arXiv preprint arXiv:2303.12767.

Aldayel, A., & Magdy, W. (2021). Stance detection on social media: State of the art and trends. *Information Processing & Management, 58*(4), 102597.

Aldayel, A., & Magdy, W. (2019). Your stance is exposed! Analysing possible factors for stance detection on social media. In *Proceedings of the ACM on human–computer interaction 3(CSCW)*, 1–20.

Biber, D., & Finegan, E. (1988). Adverbial stance types in english. *Discourse Processes, 11*(1), 1–34.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions, 7*(1), 1525–1534.

Cresci, S., Petrocchi, M., Spognardi, A., Tesconi, M., & Di Pietro, R. (2014). A criticism to society (as seen by twitter analytics). In *2014 IEEE 34th international conference on distributed computing systems eorkshops (ICDCSW)* (pp. 194–200). IEEE.

Cruickshank, I.J., & L. H. X. (2023). Use of large language models for stance classification. arXiv preprint arXiv:2309.13734.

Darwish, K., Magdy, W., & Zanouda, T. (2017). 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* (pp. 145–148).

Darwish, K., Stefanov, P., Aupetit, M., & Nakov, P. (2020). Unsupervised user stance detection on twitter. In *Proceedings of the international AAAI conference on web and social media* (vol. 14, pp. 141–152).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.(2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the ACL* (vol 1, pp. 4171–4186), Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1423.

Dias, M., & Becker, K. (2016). 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)* (pp. 378–383).

Fagni, T., & Cresci, S. (2022). Fine-grained prediction of political leaning on social media with unsupervised deep learning. *Journal of Artificial Intelligence Research, 73*, 633–672.

Fraisier, O., Cabanac, G., Pitarch, Y., Besançon, R., & Boughanem, M. (2018). Stance classification through proximity-based community detection. In *Proceedings of the 29th on hypertext and social media. HT '18* (pp. 220–228). New York, NY, USA: ACM. https://doi.org/10.1145/3209542.3209549.

Gambini, M., Senette, C., Fagni, T., & Tesconi, M. (2023). From Tweets to Stance: An unsupervised framework for user stance detection on twitter. In: Bifet, A., Lorena, A. C., Ribeiro, R. P., Gama, J., Abreu, P. H. (eds) Discovery Science. DS 2023. Lecture Notes in Computer Science, vol 14276. Springer, Cham. https://doi.org/10.1007/978-3-031-45275-8_7.

Ghosh, S., Singhania, P., Singh, S., Rudra, K., & Ghosh, S. (2019). Stance detection in web and social media: A comparative study. In *International conference of the cross-language evaluation forum for European languages* (pp. 75–87). Springer.

Gottipati, S., Qiu, M., Yang, L., Zhu, F., & Jiang, J. (2013). Predicting user's political party using ideological stances. In *International conference on social informatics* (pp. 177–191). Springer.

Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. D. L., Hanna, E. B., Bressand, F. et al. (2024). Mixtral of experts. arXiv preprint arXiv:2401.04088.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., & Hüllermeier, E. (2023). Chatgpt for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems, 35*, 22199–22213.

Küçük, D., & Can, F. (2020). Stance detection: A survey. *ACM Computing Surveys (CSUR), 53*(1), 1–37.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Tlemoyer, L. (2020). RT: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the ACL* (pp. 7871–7880). ACL, Online. https://doi.org/10.18653/v1/2020.acl-main.703.

Li, Y., Sosea, T., Sawant, A., Nair, A. J., Inkpen, D., & Caragea, C. (2021). 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).

Magdy, W., Darwish, K., Abokhodair, N., Rahimi, A., & Baldwin, T. (2016). # isisisnotislam or# deportallmuslims? predicting unspoken views. In *Proceedings of the 8th ACM conference on web science* (pp. 95–106).

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

Rashed, A., Kutlu, M., Darwish, K., Elsayed, T., & Bayrak, C. (2021). 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).

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR), 34*(1), 1–47.

Tardelli, S., Avvenuti, M., Tesconi, M., & Cresci, S. (2020). Characterizing social bots spreading financial disinformation. In *International conference on human-computer interaction* (pp. 376–392). Springer.

Thonet, T., Cabanac, G., Boughanem, M., & Pinel-Sauvagnat, K. (2017). 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).

Trabelsi, A., & Zaïane, O. R. (2018). 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).

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382.

Williams, A., Nangia, N., & Bowman, S. (2018). 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, http://aclweb.org/anthology/N18-1101.

Yin, W., Hay, J., & Roth, D. (2019). 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)* (pp. 3914–3923). Hong Kong, China: ACL. https://doi.org/10.18653/v1/D19-1404.

Zhang, B., Ding, D., & Jing, L. (2022). How would stance detection techniques evolve after the launch of chatgpt? arXiv preprint arXiv:2212.14548

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.