This is a repository copy of *Nowcasting the stance of social media users in a sudden vote: The case of the Greek referendum*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/144803/

Version: Accepted Version

Proceedings Paper:

Reuse

Takedown

Nowcasting the Stance of Social Media Users in a Sudden Vote: The Case of the Greek Referendum

Adam Tsakalidis
University of Warwick
The Alan Turing Institute
atsakalidis@turing.ac.uk

Nikolaos Aletras
University of Sheffield
n.aletras@sheffield.ac.uk

Alexandra I. Cristea
Durham University
University of Warwick
alexandra.i.cristea@durham.ac.uk

Maria Liakata
University of Warwick
The Alan Turing Institute
m.liakata@warwick.ac.uk

ABSTRACT

Modelling user voting intention in social media is an important research area, with applications in analysing electorate behaviour, online political campaigning and advertising. Previous approaches mainly focus on predicting national general elections, which are regularly scheduled and where data of past results and opinion polls are available. However, there is no evidence of how such models would perform during a sudden vote under time-constrained circumstances. That poses a more challenging task compared to traditional elections, due to its spontaneous nature. In this paper, we focus on the 2015 Greek bailout referendum, aiming to nowcast on a daily basis the voting intention of 2,197 Twitter users. We propose a semi-supervised multiple convolution kernel learning approach, leveraging temporally sensitive text and network information. Our evaluation under a real-time simulation framework demonstrates the effectiveness and robustness of our approach against competitive baselines, achieving a significant 20% increase in F-score compared to solely text-based models.

CCS CONCEPTS

• **Information systems** → **Web mining**; **Social networks**; • **Applied computing** → **Voting / election technologies**; • **Social and professional topics** → *User characteristics*;

KEYWORDS

social media; Greek referendum; natural language processing; multiple kernel learning; convolution kernels; Twitter; polarisation

1 INTRODUCTION

Predicting user voting stance and final results in elections using social media content is an important area of research in social media analysis [21, 39] with applications in online political campaigning and advertising [11, 25]. It also provides political scientists with tools for qualitative analysis of electoral behaviour on a large scale [2]. Previous approaches mainly focus on predicting national general elections, which are regularly scheduled and where data of past results and opinion polls are available [33, 60]. However, there is no evidence of how such models would work during a *sudden* and *major* political event under time-constrained circumstances. That forms a more challenging task compared to general elections, due to its spontaneous nature [35]. Building robust methods for voting

intention of social media users under such circumstances is important for political campaign strategists and decision makers.

Our work focuses on nowcasting the voting intention of Twitter users in the 2015 Greek bailout referendum that was announced in June, 27th 2015 and was held eight days later. We define a time-sensitive binary classification task where the aim is to classify a user's voting intention (YES/NO) at different time points during the entire pre-electoral period.

For this purpose, we collect a large stream of tweets in Greek and manually annotate a set of users for testing. We also collect a set of users for training via distant supervision. We predict the voting intention of the test users during the eight-day period until the day of the referendum with a multiple convolution kernel learning model. The latter allows us to leverage both temporally sensitive textual and network information. Collecting all the available tweets written in Greek[1], enables us to study user language use and network dynamics in a complete way. We demonstrate the effectiveness and robustness of our approach, achieving a significant 20% increase in F-score against competitive text-based baselines. We also show the importance of combining text and network information for inferring users' voting intention.

Our paper makes the following contributions:

- We present the first systematic study on nowcasting the voting intention of Twitter users during a sudden and major political event.
- We demonstrate that network and language information are complementary, by combining them with multiple convolutional kernels.
- We highlight the importance of the temporal modelling of text for capturing the voting intention of Twitter users.
- We provide qualitative insights on the political discourse and user behaviour during this major political crisis.

2 RELATED WORK

Most previous work on predicting electoral results focuses on forecasting the final outcome. Early approaches based on word counts [63] fail to generalize well [21, 29, 39]. Lampos et al. [33] presented a bilinear model based on text and user information, using opinion polls as the target variable. Tsakalidis et al. [60] similarly predicted the election results in different countries using Twitter and

---

[1] As per Twitter Streaming API limitations: https://developer.twitter.com/en/docs/basics/rate-limiting

polls while others used sentiment analysis methods and past results [8, 10, 44, 53]. More recently, Swamy et al. [56] presented a method to forecast the results of the latest US presidential election from user predictions on Twitter. The key difference between our task and this strand of previous work lies in its spontaneous and time-sensitive nature. Incorporating opinion polls or past results is not feasible, due to the time-constrained referendum period and the lack of previous referendum cases, respectively. Previous work on predicting the outcomes of referendums [9, 22, 36] is also different to our task, since they do not attempt to predict a single user's voting intention but rather make use of aggregated data coming from multiple users to predict the voting share of only a few test instances.

On the user-level, most past work has focused on identifying the political leaning (left/right) of a user. Early work by Rao et al. [50] explored the linguistic aspect of the task; follow-up work has also incorporated features based on the user's network [1, 16, 46, 65], leading to improvements in performance. However, most of this work predicts the (static) political ideology of clearly separated groups of users who are either declaring their political affiliation in their profiles, or following specific accounts related to a political party. This has been demonstrated to be problematic when applying such models on users that do not express political opinion [12]. Preoţiuc-Pietro et al. [49] proposed instead a non-binary, seven-point scale for measuring the self-reported political orientation of Twitter users, showcasing that the task is more difficult for users who are not necessarily declaring their political ideology. Our work goes beyond political ideology prediction, by simulating a real-world setting on a dynamically evolving situation for which there is no prior knowledge.

A smaller body of research has focused on tasks that go beyond the classic left/right political leaning prediction. Fang et al. [19] predicted the stance of Twitter users in the 2014 Scottish Independence referendum by analysing topics in related online discussions. In a related task, Zubiaga et al. [66] classified user stance in three independence movements while Stewart et al. [55] analysed user linguistic identity in the Catalan referendum. Albeit relevant, none of these works have actually studied the problem under a real-time evaluation setting or during a sudden event where the time between announcement and voting day is extremely limited (e.g., less than two weeks). Previous work on social media analysis during the Greek bailout referendum [4, 40] has not studied the task of inferring user voting intention, whereas most of the past work in opinion mining in social media in the Greek language has focused primarily on tasks related to sentiment analysis [30, 45, 61]. To the best of our knowledge, this is the first work to (a) infer user voting intention under *sudden* circumstances and a *major* political crisis; and (b) model user information over time under such settings.

3 THE GREEK BAILOUT REFERENDUM

The period of the Greek economic crisis before the bailout referendum (2009-2015) was characterized by extreme political turbulence, when Greece faced six straight years of economic recession and five consecutive years under two bailout programs [62]. Greek governments agreed to implement austerity measures, in order to secure loans and avoid bankruptcy – a fact that caused massive unrest and

demonstrations. During the same period, political parties regardless of their side on the left-right political spectrum were divided into *pro-austerity* and *anti-austerity*, while the traditional two-party system conceived a big blow [6, 52, 58].

The Greek bailout referendum was announced on June, 27th 2015 and was held eight days later. The Greek citizens were asked to respond as to whether they agree or not (YES/NO) with the new bailout deal proposed by the Troika[2] to the Greek Government in order to extend its credit line. The final result was 61.3%-38.7% in favor of the NO vote. For more details on the Greek crisis, refer to Tsebelis [62].

4 TASK DESCRIPTION

Our aim is to classify a Twitter user either as a YES or a NO voter in the Greek Bailout referendum over the eight-day period starting right before its announcement (26/6, day 0) and ending on the last day before it took place (4/7, day 8).

We assume a training set of users $D_t = \{(x_t^{(1)}, y^{(1)}), ..., (x_t^{(n)}, y^{(n)})\}$, where $x_t^{(i)}$ is a representation of user i up to time step $t \in [0, ..., 8]$ and $y^{(i)} \in \{$YES, NO$\}$. Given D_t, we want to learn a function f_t that maps a user j to her or his stance $\hat{y}^{(j)} = f_t(x_t^{(j)})$ at time t. Then, we update our model with new information shared by the users in our training set up to $t+1$, to predict the test users voting intention at $t+1$. Therefore, we mimic a real-time setup, where we nowcast user voting intention, starting from the moment before the announcement of the referendum, until the day of the referendum. Sections 5 and 6 present how we develop the training dataset D_t and the function f_t respectively.

5 DATA

Using the Twitter Streaming API during the period 18/6–16/7, we collected 14.62M tweets in Greek (from 304K users) containing at least one of 283 common Greek stopwords, starting eight days before the announcement of the referendum and stopping 11 days after the referendum date (see Figure 1). This provides us with a rare opportunity to study the interaction patterns among the users in a rather complete and unbiased setting, as opposed to the vast majority of past works, which track event-related keywords only. For example, Antonakaki et al. [4] collected 0.3M tweets using popular referendum-related hashtags during 25/06–05/07 – we have collected 6.4M tweets during the same period. In the rest of this section, we provide details on how we processed the data in order to generate our training set in a semi-supervised way (5.1) and how we annotated the users that were used as our test set in our experiments (5.2).

5.1 Training Set

Manually creating a training set would have required annotating users based on their voting preference on an issue that they had not been aware of prior to the referendum announcement. However, the same does not hold for certain accounts (e.g., major political parties) whose stance on austerity had been known a-priori given their manifestos and previous similar votes in parliament [52]. Such accounts can be used as seeds to form a semi-supervised task, under

---

[2]A decision group formed by the European Commission, the European Central Bank and the International Monetary Fund to deal with the Greek economic crisis.
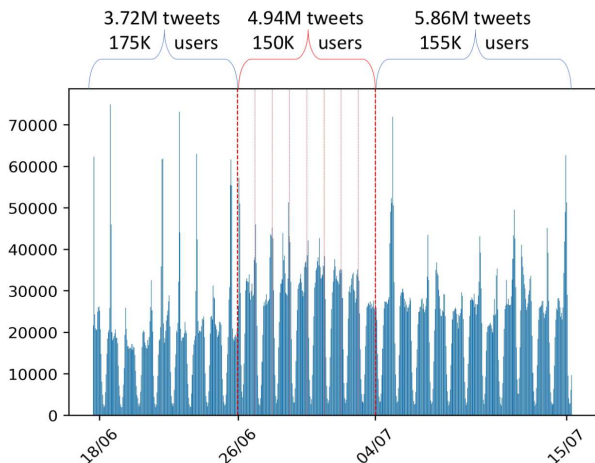
Figure 1: Number of tweets in Greek per hour. The period highlighted in red indicates the nine evaluation time points (see Section 4).

Table 1: Political position, austerity, referendum stance and national election result (January 2015) of the political parties that are used as seeds in our modelling.

Party	Position	Auster.	Referend.	Jan 15 (%)
SYRIZA (ΣΥΡΙΖΑ)	Left	anti	NO	36.34
New Democracy (Νέα Δημοκρατία)	Centre-Right	pro	YES	27.81
Golden Dawn (Χρυσή Αυγή)	Far-right	anti	NO	6.28
The River (Το Ποτάμι)	Centre	pro	YES	6.05
Independent Greeks (Ανεξάρτητοι Έλληνες)	Right	anti	NO	4.75
PASOK (ΠΑΣΟΚ)	Centre-left	pro	YES	4.68
KIDISO (ΚΙΔΗΣΟ)	Centre-left	pro	YES	2.47
ANTARSYA (ΑΝΤΑΡΣΥΑ)	Far-left	anti	NO	0.64
Creation Again (Δημιουργία Ξανά)	Centre-Right	pro	YES	-

Table 2: Number of users (u) and tweets (t) used in our experiments per evaluation day.

day	0	1	2	3	4	5	6	7	8
date	26/06	27/06	28/06	29/06	30/06	01/07	02/07	03/07	04/07
$train_u$	2121	2121	2121	2121	2121	2121	2121	2121	2121
$train_t$	307K	395K	468K	543K	609K	685K	752K	814K	867K
$test_u$	1804	1985	2045	2115	2146	2174	2184	2194	2197
$test_t$	293K	358K	414K	477K	533K	599K	658K	718K	768K

the hypothesis that users who are re-tweeting a political party more often than others, are likely to follow its stance in the referendum, once this is announced. Hence, we compile a set of 267 seed accounts (148 YES, 119 NO) focusing on the pre-announcement period including: (1) *political parties*; (2) *members of parliament (MPs)*; and (3) *political party members*.

- **Political Parties**: We add as seeds the Twitter accounts of nine major and minor parties[3] with a known stance on austerity before the referendum (5 YES, 4 NO, see Table 1). We assume that the pro-austerity parties will back the bailout proposal (YES), while the anti-austerity parties will reject it (NO). The pro-/anti- austerity stance of the parties was known before the referendum, since the pro-austerity parties had already backed previous bailout programs in parliament or had a clear favorable stance towards them, whereas the opposite holds for the anti-austerity parties [52].
- **MPs** The accounts of the (300) MPs of these parties were manually extracted and added as seeds. 153 such accounts were identified (82 YES, 71 NO) labelled according to the austerity stance of their affiliated party.
- **Political Party Members** We finally compiled a set of politically related keywords to look up in Twitter user account names and descriptions (names/abbreviations of the nine parties and keywords such as "candidate"). We identified 257 accounts (133 YES, 124 NO), which were manually inspected by human experts to filter out irrelevant ones (e.g., the word "River" might not refer to the political party) and kept only those that had at least one tweet during the period preceding the announcement of the referendum (44 NO, 61 YES).

To expand the set of seed accounts, we calculate for every user u in our dataset during the pre-announcement period his/her score as:

$$score(u) = PMI(u, \text{YES}) - PMI(u, \text{NO}),$$

---

[3] We excluded KKE (Greek Communist Party) since an active official Twitter account did not exist at the time.

where $PMI(u, lbl)$ is the pointwise mutual information between a certain user and the respective seeding class (YES/NO). A high (low) score implies that the user is endorsing often YES-related (NO-related) accounts, thus he/she is more likely to follow their stance after the referendum is announced. This approach has been successfully applied to other related natural language processing tasks, such as building sentiment analysis lexical resources using a pre-defined list of seed words [43]. Assigning class labels to the users based on their scores, we set up a threshold $tr = n(\max(|scores|))$, with $n \in [0, 1]$. We assign the label YES to a user u if $score(u) > tr$ or NO if $score(u) < -tr$. Setting $n = 0$, would imply that we are assigning the label YES if the user has re-tweeted more YES-supporting accounts (and inversely), which might result into a low quality training set, whereas higher values for n would imply a smaller (but of higher quality) training set. During development, we empirically set $n = 0.5$ to keep users who are fairly closer to one class than the other. From the final set of 5,430 users that have re-tweeted any seed account, 2,121 were kept (along with the seed accounts) as our training set (965 YES, 1,156 NO).

5.2 Test Set

For evaluation purposes, we generate a test set of active users that are likely to participate in political conversations on Twitter. First, we identify all users having tweeted at least 10 times after the referendum announcement (86,000 users). From the 500 most popular hashtags in their tweets, we selected those that were clearly related to the referendum (189) which were then manually annotated with respect to potentially conveying the user's voting intention (e.g., "yesgreece", "no" as opposed to neutral ones, such as "referendum"). Finally, we selected a random sample of 2,700 users (out of 22K) that had used more than three such hashtags, to be manually annotated – without considering any user from the training set. This is standard practice in related work [20, 55] and enables us to evaluate our models on a high quality test set, as opposed to previous related

work which rely on keyword-matching approaches to generate their test set [19, 66].

Two authors of the paper (Greek native speakers) annotated each of the users in the test set, using the tweets after the referendum announcement. Each annotator was allowed to label an account as YES, NO, or N/A, if uncertain. There was an agreement on 2,365 users (Cohen's $\kappa = .75$) that is substantially higher if the N/A labels are not considered ($\kappa = .98$), revealing high quality in the annotations, i.e., in the upper part of the 'substantial' agreement band [5]. We discarded all accounts labelled as N/A by an annotator and used the remaining accounts where the annotators agreed for the final test set, resulting to 2,197 users – similar test set sizes are used in related tasks [18]. The resulting user distribution (NO 77%, YES 23%) is more imbalanced compared to the actual result of the referendum, due to the demographic bias on Twitter [42]. To mimic a real-time scenario, we refrained from balancing our train/test sets, since it would have been rather impossible to know the voting intention distribution of Twitter users a-priori. Overall, we use 18.9% (1.64M/8.66M) of the tweets written in Greek during that period in our experiments (see Table 2).

6 MODELS

6.1 Convolution Kernels

Convolution kernels are composed of sub-kernels operating on the item-level to build an overall kernel for the object-level [14, 23] and can be used with any kernel based model such as Support Vector Machines (SVMs) [27]. Such kernels have been applied in various NLP tasks [14, 31, 37, 64]. Here we build upon the approach of Lukasik and Cohn [37] by combining convolution kernels operating on available (1) *text*; and (2) *network information*.

Let a, b denote two objects (e.g., social network users), represented by two $M \times N$ matrices Z_a and Z_b respectively, where M denotes the number of items representing the object and N the dimensionality of an item vector. For example, an item can be a user's tweet or network information. A kernel K between the two objects (users) a and b over Z_a and Z_b is defined as:

$$K_z(a,b) = \frac{1}{|Z_a||Z_b|} \sum_{i,j} k_z(z_a^i, z_b^j), \qquad (1)$$

where k_z is any standard kernel function such as a linear or a radial basis function (RBF). One can also normalise K_z by dividing its entries $K_z(i,j)$ by $\sqrt{K_z(i,i)K_z(j,j)}$.

The resulting kernel has the ability to capture the similarities across objects on a per-item basis. However, unless restricted to operate on consecutive items (time-wise), it ignores their temporal aspect. Given a set of associated timestamps $T_o = \{t_o^1, ..., t_o^N\}$ for the items of each object o, Lukasik and Cohn [37] proposed to combine the temporal and the item aspects as:

$$K_{zt}(a,b) = \frac{1}{|Z_a||Z_b|} \sum_{i,j} k_z(z_a^i, z_b^j) k_t(t_a^i, t_b^j), \qquad (2)$$

where k_t is any valid kernel function operating on the timestamps of the items. Here, K_{zt} is a matrix capturing the similarities across users by leveraging both the information between pairs of items and their temporal interaction.

6.1.1 Text Kernels.
Let a, b denote two users in a social network, posting messages $W_a = \{w_a^1, ..., w_a^N\}$ and $W_b = \{w_b^1, ..., w_b^M\}$ with associated timestamps $T_a = \{t_a^1, ..., t_a^N\}$ and $T_b = \{t_b^1, ..., t_b^M\}$ respectively. We assume that a message w_i^j of user i at time j is represented by the mean k-dimensional embedding [41] of its constituent terms. This way, we can obtain *text convolution kernels*, K_w and K_{wt} by simply replacing Z and z with W and w respectively in Equations 1 and 2. Following Lukasik and Cohn [37], we opted for a linear kernel operating on text and an RBF on time.

6.1.2 Network Kernels.
Let assume a set of directed weighted graphs $G = \{G_1(N_1, E_1), ..., G_t(N_t, E_t)\}$, where $G_i(N_i, E_i)$ represents the retweeting activity graph of the N_i users at a time point $i \in T = \{1, .., t\}$. Let $L_a \in \mathbf{R}^{N,k}$, $L_b \in \mathbf{R}^{M,k}$ denote the resulting matrices of a k-dimensional, network-based user representation for two users a and b across time. Contrary to the textual vector representation w_i^j that is defined over a fixed space given a pre-defined vocabulary, user network vector representations (e.g., graph embeddings [57]), are computed at each time step on a different network structure. Thus, a standard similarity score between two user representations at timepoints t and $t+1$ cannot be used, since the network vector spaces are different. To accommodate this, at each time point t we calculate the median L_{YES}^t and L_{NO}^t vectors for each class of our training examples and update the respective user vectors as:

$$L_u^{*t} = d(L_{\text{YES}}^t, L_u^t) - d(L_{\text{NO}}^t, L_u^t),$$

using some distance metric d (for simplicity, we opted for the linear distance). If a user has not retweeted, his/her original network representation l_u^t is calculated as the average across all user representations at t. Finally, the *network convolution kernels*, K_n and K_{nt} are computed using Equations 1 and 2 respectively by simply replacing Z with L^* and z with l^*. Similarly to text kernels, we use a linear kernel k_n for the network and an RBF kernel k_t for time.

6.1.3 Kernel Summation.
We can combine the text and network convolution kernels by summing them up: $K_{sum} = K_w + K_{wt} + K_n + K_{nt}$. This implies a simplistic assumption that the contribution of the different information sources with respect to our target is equal. While this might hold for a small number of carefully designed kernels, it lacks the ability to generalise over multiple kernels of potentially noisy representations.

6.2 Convolution Kernel Models

6.2.1 SVMs with Convolution Kernels.
Convolution kernels can be used with any kernel based model. Here, we use them with SVMs. First, a SVM_s operates on a single information source $s = \{w, n\}$, i.e., SVM_w for text and SVM_n for network. Second, a SVM_{st} takes temporal information into account combined with text (SVM_{wt}) and network (SVM_{nt}) information respectively. Finally, we combine the text and the network information using a linear kernel summation (K_{sum}) of their respective kernels (SVM_{sum}).

6.2.2 Multiple Convolution Kernel Learning (MCKL).
Multiple kernel learning methods learn a weight for each kernel instead of assigning equal importance to all of them allowing more flexibility. Such approaches have been extensively used in tasks where different data modalities exist [26, 48, 59]. We build upon the approach of Sonnenburg et al. [54] to build a model based on labelled instances

$x_i \in I$, by combining the different convolution kernels K_s with some weight $w_s > 0$ s.t. $\sum_s w_s = 1$ and apply:

$$f(x) = sign\left(\sum_{i \in I} \alpha_i \sum_s w_s K_s(x, x_i) + b\right).$$

The parameters α_i, the bias term b and the kernel weights are estimated by minimising the expression:

$$\min \quad \gamma - \sum_{i \in I} \alpha_i$$

$$\text{w.r.t.} \quad \gamma \in R, \alpha \in R_+^{|I|}$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C \,\forall i, \quad \sum_{i \in I} \alpha_i y_i = 0$$

$$\frac{1}{2} \sum_{i \in I} \sum_{j \in I} \alpha_i \alpha_j y_i y_j K_s(\mathbf{x}_i, \mathbf{x}_j) \le \gamma \quad \forall s.$$

This way, the four convolution kernels are calculated individually and subsequently combined in a weighted scheme accounting for their contribution in the prediction task. This allows us to combine external and asynchronous information (e.g., news articles), while adding other kernels capturing different aspects of the users (e.g., images) is straight-forward.

7 EXPERIMENTAL SETUP

7.1 Features

7.1.1 Textual Information (TEXT).
We obtain word embeddings by training `word2vec` [41] on a collection of 14.7 non-retweeted tweets obtained by [61], collected in the exact same way as our dataset, over a separate time period. We performed standard pre-processing steps including lowercasing, tokenising, removal of non-alphabetic characters, replacement of URLs, mentions and all-upper-case words with identifiers. We used the CBOW architecture, opting for a 5-token window around the target word, discarding all words appearing less than 5 times and using negative sampling with 5 "noisy" examples. After training, each word is represented as a 50-dimensional vector. Each tweet in our training and test set is represented by averaging each dimension of its constituent words.

7.1.2 Network Information (NETWORK).
We trained LINE [57] embeddings at different timesteps, by training on the graphs $\{G_1(N_1, E_1), ..., G_T(N_T, E_T)\}$, where N_i is the set of users and E_i is the (directed, weighted) set of retweets amongst N_i up to time i. We choose the "retweet" rather than the "user mention" network due to its more polarised nature, as indicated by past work [15][4]. LINE was preferred over alternative models [47, 51] due to its ability to model directed weighted graphs. We construct the network G_t every 12 hours based on the retweets among all users up to time t, and LINE is trained on G_t to create 50-dimensional user representations. We used the second-order proximity, since it performed better than the first-order in early experimentation. We also refrained from concatenating them to keep the dimensionality relatively low.

---

[4]The "following" network cannot be constructed based on the JSON objects returned by Twitter Streaming API; to achieve this requires a very large number of API calls and cannot be constructed accurately in a realistic scenario.

7.2 Models

7.2.1 Convolution Kernel Models.
Our MCKL and our SVM models are fed with the convolution kernels operating on the tweet-level (for TEXT) and each NETWORK representation (derived every 12 hours), based on the tweets and re-tweeting activity respectively of the users up to the current evaluation time point.

7.2.2 Baselines.
We compare our proposed methods against competitive baselines that are commonly used in social media mining tasks trained on feature aggregates [38, 66]. We obtain a TEXT representation of a user at each time step t by averaging embedding values across all his/her tweets until t. Similarly, a user NETWORK representation is computed from the retweeting graph up until t. Finally, we train a regularised Logistic Regression (**LR**) with L_2 regularisation [34], a feed-forward neural network (**FF**) [24], a Random Forest (**RF**) [7] and a **SVM**.

Model Parameters. Parameter selection of our models and the baselines is performed using a 5-fold cross-validation on the training set. We experiment with different regularisation strength (10^{-3}, 10^{-2}, ..., 10^3) for LR, different number of trees (50, 100, ..., 500) for RF, and different kernels (linear, RBF) and parameters C and γ ($10^{-3}, 10^{-2}, ..., 10^3$) for SVMs. For FF, we stack dense layers, each followed by a ReLU activation and a 20% dropout layer, and a final layer with a sigmoid activation function. We train our network using the Adam optimiser [32] with the binary cross-entropy loss function and experiment with different number of hidden layers (1, 2), units per layer (10, 25, 50, 75, 100, 150, 200), batch size (10, 25, 50, 75, 100) and number of epochs (10, 25, 50, 100). For MCKL, we experiment with the same C values as in SVM and apply an L_2 regulariser.

7.3 Evaluation

We train and test our models based on the data collected on a daily basis (every midnight), starting from the day before the announcement of the referendum (day 0) until the day before its due date (day 8). This way, we mimic a real-time setting and gain better evaluation insights. To evaluate our models, we compute the macro-average F-score, which forms a more challenging metric compared to micro-averaging, given the imbalanced distribution of our test set. At each evaluation time point t, we use information about the users in our training set up to t, to classify the test users that have tweeted at least once up to t (note that all of the users in our training set have tweeted before the announcement of the referendum, thus the size of the training set in terms of number of users remains constant). This results into a different number of test instances per day (see Table 2). However, we did not observe any major differences in our evaluation by excluding newly added users. Parameter selection is performed on every evaluation day using a 5-fold cross-validation on the training set.

8 RESULTS

8.1 Nowcasting Voting Intention

Figure 2 presents the macro-average F-scores obtained by the methods compared in all days from the announcement to the day of the referendum. As expected, the closer the evaluation is to the referendum date, the more accurate the models since more information
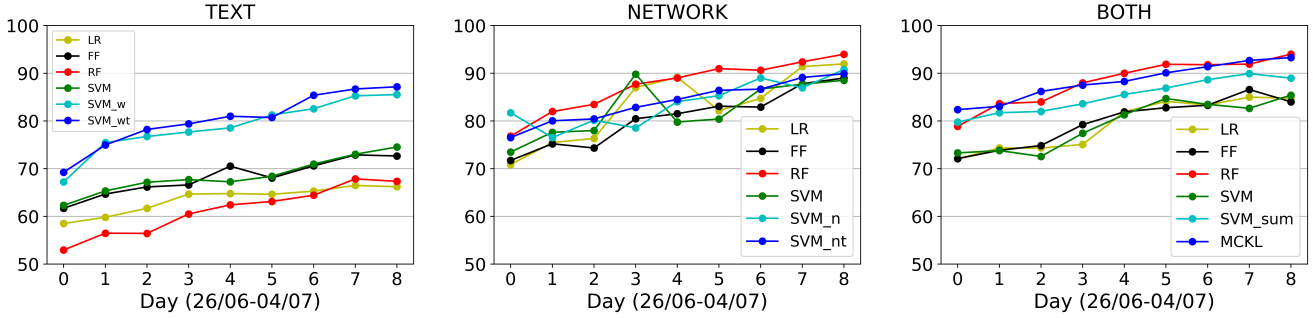
Figure 2: Macro-average F-score across all evaluation days using TEXT, NETWORK and BOTH user representations.

Table 3: Average F-score and standard deviation across all evaluation days using TEXT, NETWORK and BOTH user representations. SVM_s and SVM_{st} denote the SVM with convolution kernels (SVM_w, SVM_n) and (SVM_{wt}, SVM_{nt}), respectively.

	TEXT	NETWORK	BOTH
LR	63.55 ±2.86	83.21±7.55	79.43±5.34
FF	68.19 ±3.78	80.66±5.93	79.83±5.11
RF	61.27 ±5.14	**87.43**±5.60	88.22±5.03
SVM	68.51 ±3.80	82.43±5.83	79.39±5.18
SVM_s	78.91±5.68	83.65±4.82	–
SVM_{st}	**80.30**±5.81	84.03±4.47	–
SVM_{sum}	–	–	85.22±3.64
MCKL	–	–	**88.31**±3.95

Figure 3: Change in performance (mean/standard deviation) compared to the results in Figure 2, after 100 experiments with added noisy features.

becomes available for each user. Table 3 shows the average (across-all-days) F-score by each model.

Temporal convolution kernels using **TEXT** (SVM_{wt}) significantly outperform the best text-based baseline ($p = .001$, Kruskal-Wallis test against SVM), with an average of 11.8% and 17.2% absolute and relative improvement respectively. This demonstrates the model's ability on capturing the similarities between different users on a per-tweet basis compared to simpler models using tweet aggregates. Also, SVM_w and SVM_{wt} implicitly capture similarities in the retweeting activity of the users. This is important, since network information might not be easily accesible (e.g., due to API limitations) while it is expensive to compute at each timestep. Hence, one can use SVM_{wt} to model user written content and partially capture network information.

Classification accuracy consistently improves when using the **NETWORK** representation (i.e., graph embeddings). RF achieves 94% F-score on the day before the referendum, whereas the worst-performing baseline (FF) still achieves 80.66% F-score on average. SVM_{nt} provides a small boost (1.6% on average) compared to the vanilla SVM, which uses only the user representations derived at the current time point. This implies that the current network structure is indicative of users' voting intention, probably because the referendum was the dominant topic of discussion at the time, e.g., most of the retweeting activity was relevant the referendum (see Section 9). This is also in line with recent findings of Aletras and
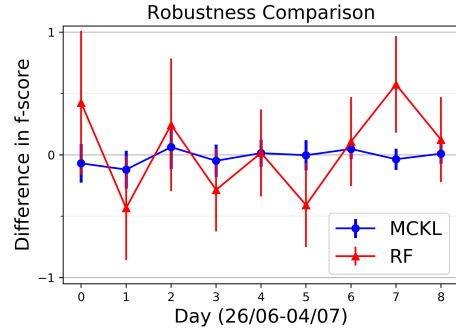
Chamberlain [3] on predicting occupation class and income where network information is more predictive than language.

When combining the user text and network representation (**BOTH**), the baselines fail to improve over using only NETWORK. In contrast, our MCKL improves by 4.28% over the best performing single convolution kernel model (SVM_{nt}). This demonstrates that MCKL can effectively combine information from both representations by weighting their importance, and further improve the accuracy of the best performing single representation model. Overall, MCKL significantly outperforms the best performing text-based baseline by approximately 20% in F-score ($p < .001$, Kruskal-Wallis test).

8.2 Robustness Analysis

Due to the semi-supervised nature of our task, it is impossible to judge whether the small difference between MCKL and RF stems from a better designed model. Furthermore, it is difficult to assess MCKL's effectiveness with respect to its ability to generalise over multiple and potentially noisy feature sources.

To assess the robustness of the best performing models (MCKL, RF) operating on **BOTH** information sources, here we perform experiments by adding random noise in their input. We assume that there is a noisy source generating an extra K-dimensional representation X for every user that we add as extra input to the models. We set $K = 25$, so that *(a)* we account for a smaller noisy input compared

to our features (25 vs 50) and *(b)* 1/5 of our kernels in MCKL and 25/125 input features in RF are noisy. We perform 100 runs, each time drawing random noise $X \sim N(0, 1)$.

Our results indicate that RF is more sensitive to the noisy input compared to MCKL (see Figure 3). On average, RF achieves a small boost (0.04%) in performance with the added noise. That together with the higher standard deviation reveal the vulnerability of RF to potentially corruption and stochasticity introduced in the input. On the contrary, MCKL is consistently robust, achieving only a tiny reduction in performance on average across all days (0.02%) while the respective average standard deviation is lower than the one achieved by RF (0.12 vs 0.41). This robustness is highly desirable is cases of such sudden political events and it also indicates that we can add kernels capturing different properties of our task (e.g., user-related information, images, etc.), without having to decide a-priori which of them are indeed predictive of the user's voting intention. We plan to investigate this in future work.

9 QUALITATIVE ANALYSIS

In this section, we provide insights into the temporal variation observed in the users' shared content and the network structure during this major political crisis. Besides performing a qualitative analysis during this time period, we believe that this analysis will also provide insights on *(a)* the reasons that trigger the significant improvement in performance of convolution kernels methods operating on TEXT, and *(b)* the reason that our non-temporally-sensitive baselines are rather competitive to our convolution kernel models, when using NETWORK information. In the current section we provide details on both of these aspects.

9.1 Language

We are interested in investigating which are the political-related entities that voters from both sides most likely mention. We expect that this will shed light on the main focus of discussion in the political debates between the YES/NO voters that occurred after the announcement of the referendum. For this, two authors manually compiled two lists of n-grams containing different ways of referring[5] to the *(a)* the six major political parties and *(b)* their leaders (see Table 1). We represent every YES/NO user in the test set as aggregated `tf-idf` values of the ngrams (1-3) appearing in his/her concatenated tweets; then, we compute an n-gram 's n score as $PMI(n, \text{YES}) - PMI(n, \text{NO})$. A positive score implies that it is highly associated with users who support the YES vote, and vice versa.

Figure 4 shows that the parties and leaders that supported one side, mostly appear in tweets of users supporting the opposite side. This is more evident when we consider tweets shared by the users *after* the announcement of the referendum. Examining the content of highly-retweeted tweets, revealed sarcasm and hostility for the opposite side in the majority of them (see Table 5). Hostility is a frequent phenomenon in public debates [28] and our findings corroborate previous work showing that the political discourse on Twitter is polarised [15, 20].

Finally, we examine the temporal variation of language over the same two periods. Table 4 shows the most similar words (translated
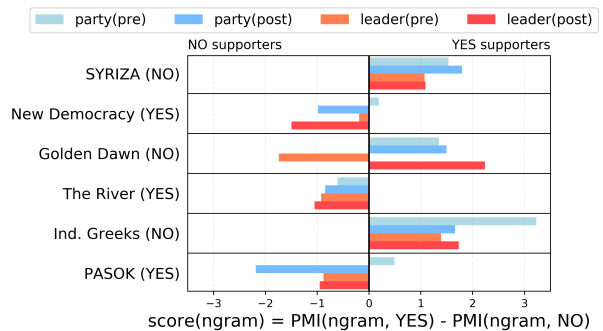
---

[5]Note that Greek is a fully inflected language. We opted not to apply stemming because inflected word forms carry meaningful information.

Figure 4: Scores of n-grams related to the political parties/leaders, pre (18/06-26/06) and post (27/06-05/07) the referendum announcement.

Table 4: Most similar words to YES and NO (translated to English), when training word2vec on different time periods.

	Before the announcement (18/06-26/06)	After the announcement (27/06-05/07)
YES	no, ok, nah, alright, sure, usrmnt, hahaha, alrighty, but, so	no, abstain, referendum, KKE, question, invalid, euro, clearly, clash, nai
NO	yes, only, sure, so (slang), disagree, mainly, especially, obviously, so (abbrv), agree	yes, abstain, KKE, referendum, clash, question, people, invalid, vote, clearly

to English) to the *yes* and *no* words, measured by cosine similarity, when training `word2vec` using the tweets of each time period. The difference of the cosine similarities $cos_{post} - cos_{pre}$ between the *yes/no* vectors and each of their corresponding most similar words over these two periods is shown in Figure 5. After the announcement, the context of the two words shifts towards the political domain. That might explain why text aggregates become noisy, as shown in our results. Convolution kernels are able to filter-out this noise since they operate on the tweet level by also taking the time into account. We plan to study the semantic variation in language [17] in a more fine-grained way in future work.

9.2 Network

We explore the differences in retweeting behaviour of users over the same periods (*(a)* before the announcement of the referendum and *(b)* after and until the day of the referendum), by training two different LINE embedding models using tweets from the each period respectively. Figure 6 shows the plots of the first two dimensions of the graph embeddings before and after the announcement using principal component analysis. The results unveil the effects of the referendum announcement and provide insights on the effectiveness of NETWORK information for predicting vote intention, as demonstrated in our results. Before, YES and NO users appear to have similar retweeting behaviour, which changes after the announcement. This finding illustrates the political homophily of the social

Table 5: Examples of highly re-tweeted tweets after the announcement of the referendum.

| Tweet	#RT
They say that there is a long queue of people in ATMs but they show only 6 people waiting; this is not a queue, this is <u>PASOK</u>.	686
Looking for any angry tweets by <u>SYRIZA</u> fans concerning Kasidiaris's (Golden Dawn MP) release from prison. Have you seen any?	246
I want to write something funny regarding the statements made by <u>Kammenos</u> (Ind. Greeks leader), but I cannot find something funnier than the statements made by <u>Kammenos</u>.	178
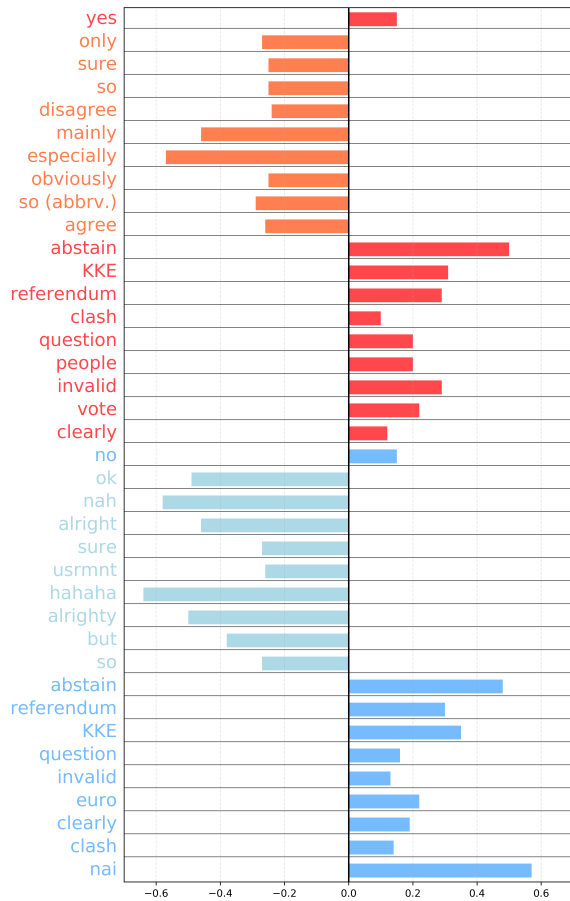
Figure 5: Difference in cosine similarity ($cos_{post}(w_{no/yes}, w) - cos_{pre}(w_{no/yes}, w)$) **between the *no/yes* (red/blue) word vectors** $w_{no/yes}$ **and each of their most similar words in the two periods.**

network [13] and highlights the extremely polarised pre-election period [62].

Next, we question whether the distance between the two classes of users through time changes according to time points at which
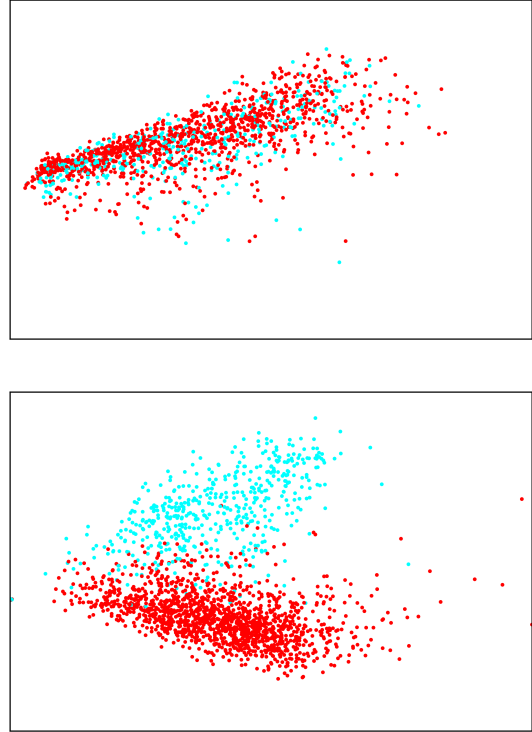

Figure 6: Network representations of YES/NO (blue/red) users, before (above) and after (below) the referendum announcement.

real-world events occur. To answer this, we compute the network embeddings of the train and test users every 12 hours, as in our experiments, and represent every class (YES/NO) at a certain time point t by the average representations (avg_Y^t, avg_N^t) of the corresponding users in the training set at t. Then, for every user u in the test set, we use the cosine similarity cos to calculate:

$$network\_score_u^t = cos(u^t, avg_Y^t) - cos(u^t, avg_N^t).$$

Finally, we calculate the average score of the YES and the NO users in the test set ($network_Y^t$, $network_N^t$) at every time point t and normalise the corresponding time series s.t. $network_Y(0)=network_N(0) = 0$. We also employ an alternative approach, by generating the network embeddings on a seven-day sliding-window fashion and following the same process. The results are shown in Figure 7. In both cases, the YES/NO users start to deviate from each other right after the announcement of the referendum, with an upward/downward YES/NO trend until the day of the referendum. This is effectively captured in our modelling and might explain the reason for the high accuracy achieved even by our baseline models, which are trained using the network representation of the users in the last day only. However, the YES/NO users start to again approach each other only in the sliding window approach after the referendum day, since in our modelling the representations are built based on re-tweets aggregates over the
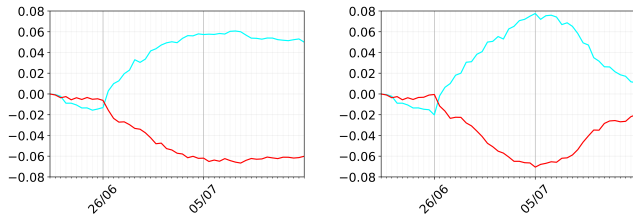
Figure 7: Normalised difference of similarity of YES/NO (blue/red) users in our modelling (left) and in a sliding window approach (right).

whole period. While this does not seem to have affected our performance, exploring the temporal structure of the network formations through time is of vital importance for longer lasting electoral cases.

10 LIMITATIONS AND FUTURE WORK

Despite working under a real-time simulation setting, we are aware that our results come with some caution, owed to the selection of the users in our test set. The limitations stem from the fact that we have selected highly active users that have used at least three polarised hashtags in their tweets after the announcement of the referendum. As previous work has shown [12, 49], we expect that the performance of any model is likely to drop, if tested in a random sample of Twitter users. We plan to investigate this, by annotating a random sample of Twitter users and comparing the performance in the two test sets, in our future work.

We also plan to assess the ability of MCKL to generalise, through exploring different referendum cases and incorporating more sources of information in our modelling. Finally, we plan to study the temporal variation of language and network in a more fine-grained way.

11 CONCLUSION

We presented a distant-supervised multiple convolution kernel approach, leveraging temporally sensitive language and network information to nowcast the voting stance of Twitter users during the 2015 Greek bailout referendum. Following a real-time evaluation setting, we demonstrated the effectiveness and robustness of our approach against competitive baselines, showcasing the importance of temporal modelling for our task.

In particular, we showed that temporal modelling of the content generated by social media users provides a significant boost in performance (11%-19% in F-score) compared to traditional feature aggregate approaches. Also, in line with past work on inferring the political ideology of social media users [1, 16], we showed that the network structure (in our case, the re-tweet network) of the social media users is more predictive of their voting intention, compared to the content they share. By combining those two temporally sensitive aspects (text, network) of our task via a multiple kernel learning approach, we further boost the performance, leading to an overall significant 20% increase in F-score against the best performing, solely text-based feature aggregate baseline. Finally, we provided qualitative insights on aspects related to the shift in online discussions and

polarisation phenomena that occurred during this time period, which are effectively captured through our temporal modelling approach.

ACKNOWLEDGEMENTS

REFERENCES

[1] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. 2012. Homophily and Latent Attribute Inference: Inferring Latent Attributes of Twitter Users from Neighbors. *ICWSM* 270 (2012), 2012.

[2] John H Aldrich, Rachel K Gibson, Marta Cantijoch, and Tobias Konitzer. 2016. Getting out the vote in the social media era: Are digital tools changing the extent, nature and impact of party contacting in elections? *Party Politics* 22, 2 (2016), 165–178.

[3] Nikolaos Aletras and Benjamin Paul Chamberlain. 2018. Predicting Twitter User Socioeconomic Attributes with Network and Language Information. In *Proceedings of the 29th on Hypertext and Social Media (HT '18)*. 20–24.

[4] Despoina Antonakaki, Dimitris Spiliotopoulos, Christos V Samaras, Polyvios Pratikakis, Sotiris Ioannidis, and Paraskevi Fragopoulou. 2017. Social media analysis during political turbulence. *PloS one* 12, 10 (2017), e0186836.

[5] Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for Computational Linguistics. *Computational Linguistics* 34, 4 (2008), 555–596.

[6] Anna Bosco and Susannah Verney. 2012. Electoral Epidemic: The Political Cost of Economic Crisis in Southern Europe, 2010–11. *South European Society and Politics* 17, 2 (2012), 129–154.

[7] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[8] Pete Burnap, Rachel Gibson, Luke Sloan, Rosalynd Southern, and Matthew Williams. 2016. 140 characters to victory? Using Twitter to predict the UK 2015 General Election. *Electoral Studies* 41, Supplement C (2016), 230 – 233.

[9] Fabio Celli, Evgeny Stepanov, Massimo Poesio, and Giuseppe Riccardi. 2016. Predicting Brexit: Classifying agreement is better than sentiment and pollsters. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*. 110–118.

[10] Andrea Ceron, Luigi Curini, Stefano M Iacus, and Giuseppe Porro. 2014. Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. *New Media & Society* 16, 2 (2014), 340–358.

[11] Derrick L. Cogburn and Fatima K. Espinoza-Vasquez. 2011. From Networked Nominee to Networked Nation: Examining the Impact of Web 2.0 and Social Media on Political Participation and Civic Engagement in the 2008 Obama Campaign. *Journal of Political Marketing* 10, 1-2 (2011), 189–213.

[12] Raviv Cohen and Derek Ruths. 2013. Classifying Political Orientation on Twitter: It's not Easy!. In *ICWSM*.

[13] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. 2014. Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data. *Journal of Communication* 64, 2 (2014), 317–332.

[14] Michael Collins and Nigel Duffy. 2002. Convolution kernels for natural language. In *NIPS*. 625–632.

[15] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. 2011. Political polarization on Twitter. In *ICWSM*, Vol. 133. 89–96.

[16] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. 2011. Predicting the Political Alignment of Twitter Users. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE 3rd International Conference on Social Computing (SocialCom), 2011 IEEE 3rd International Conference on*. IEEE, 192–199.

[17] Marco Del Tredici and Raquel Fernández. 2017. Semantic Variation in Online Communities of Practice. In *IWCS*.

[18] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *SemEval*. 69–76.

[19] Anjie Fang, Iadh Ounis, Philip Habel, Craig Macdonald, and Nut Limsopatham. 2015. Topic-centric classification of Twitter user's political orientation. In *SIGIR*. 791–794.

[20] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political Discourse on Social Media: Echo Chambers, Gatekeepers, and the Price of Bipartisanship. In *WWW*. 913–922.

[21] Daniel Gayo-Avello. 2012. I Wanted to Predict Elections with Twitter and all I got was this Lousy Paper–A Balanced Survey on Election Prediction using Twitter Data. *arXiv preprint arXiv:1204.6441* (2012).

[22] Miha Grčar, Darko Cherepnalkoski, Igor Mozetič, and Petra Kralj Novak. 2017. Stance and influence of Twitter users regarding the Brexit referendum. *Computational Social Networks* 4, 1 (2017), 6.

[23] David Haussler. 1999. *Convolution kernels on discrete structures*. Technical Report. Department of Computer Science, University of California at Santa Cruz.

[24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[25] Philip N. Howard. 2005. Deep Democracy, Thin Citizenship: The Impact of Digital Media in Political Campaign Strategy. *The Annals of the American Academy of Political and Social Science* 597, 1 (2005), 153–170.

[26] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. 2015. Multi-task, multi-kernel learning for estimating individual wellbeing. In *NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*, Vol. 898.

[27] Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *ECML*. 137–142.

[28] Charlotte Jørgensen. 1998. Public debate–an act of hostility? *Argumentation* 12, 4 (1998), 431–443.

[29] Andreas Jungherr, Pascal Jürgens, and Harald Schoen. 2012. Why the Pirate Party Won the German Election of 2009 or the Trouble with Predictions: A Response to Tumasjan, A., Sprenger, T.O., Sander, P.G., & Welpe, I.M. "Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment". *Social science computer review* 30, 2 (2012), 229–234.

[30] Georgios Kalamatianos, Dimitrios Mallis, Symeon Symeonidis, and Avi Arampatzis. 2015. Sentiment analysis of Greek tweets and hashtags using a sentiment lexicon. In *Proceedings of the 19th Panhellenic Conference on Informatics*. ACM, 63–68.

[31] Jonghoon Kim, François Rousseau, and Michalis Vazirgiannis. 2015. Convolutional sentence kernel from word embeddings for short text categorization. In *EMNLP*. 775–780.

[32] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[33] Vasileios Lampos, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2013. A user-centric model of voting intention from Social Media. In *ACL*, Vol. 1. 993–1003.

[34] Saskia Le Cessie and Johannes C Van Houwelingen. 1992. Ridge estimators in logistic regression. *Applied statistics* (1992), 191–201.

[35] Lawrence Leduc. 2002. Opinion change and voting behaviour in referendums. *European Journal of Political Research* 41, 6 (2002), 711–732.

[36] Julio Cesar Amador Diaz Lopez, Sofia Collignon-Delmar, Kenneth Benoit, and Akitaka Matsuo. 2017. Predicting the Brexit Vote by Tracking and Classifying Public Opinion Using Twitter Data. *Statistics, Politics and Policy* 8, 1 (2017), 85–104.

[37] Michal Lukasik and Trevor Cohn. 2016. Convolution Kernels for Discriminative Learning from Streaming Text. In *AAAI*. 2757–2763.

[38] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *CIKM*. 1751–1754.

[39] Panagiotis T Metaxas, Eni Mustafaraj, and Dani Gayo-Avello. 2011. How (not) to predict elections. In *Privacy, Security, Risk and Trust and IEEE Third Inernational Conference on Social Computing*. 165–171.

[40] Asimina Michailidou. 2017. Twitter, Public Engagement and the Eurocrisis: More than an Echo Chamber? In *Social Media and European Politics*. Springer, 241–266.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[42] Renato Miranda Filho, Jussara M Almeida, and Gisele L Pappa. 2015. Twitter population sample bias and its impact on predictive outcomes: a case study on elections. In *Advances in Social Networks Analysis and Mining*. 1254–1261.

[43] Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *SemEval*, Vol. 2. 321–327.

[44] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *ICWSM*, Vol. 11. 1–2.

[45] Elisavet Palogiannidi, Polychronis Koutsakis, Elias Iosif, and Alexandros Potamianos. 2016. Affective Lexicon Creation for the Greek Language. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*

[46] Marco Pennacchiotti and Ana-Maria Popescu. 2011. A Machine Learning Approach to Twitter User Classification. *ICWSM* 11, 1 (2011), 281–288.

[47] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. 701–710.

[48] Soujanya Poria, Haiyun Peng, Amir Hussain, Newton Howard, and Erik Cambria. 2017. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing* 261 (2017), 217–230.

[49] Daniel Preoţiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond Binary Labels: Political Ideology Prediction of Twitter Users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 729–740.

[50] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying Latent User Attributes in Twitter. In *Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents*. ACM, 37–44.

[51] Georgios Rizos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2017. Multilabel user classification using the community structure of online networks. *PloS one* 12, 3 (2017), e0173347.

[52] Wolfgang Rüdig and Georgios Karyotis. 2013. Beyond the Usual Suspects? New Participants in Anti-Austerity Protests in Greece. *Mobilization: An International Quarterly* 18, 3 (2013), 313–330.

[53] Lei Shi, Neeraj Agarwal, Ankur Agrawal, Rahul Garg, and Jacob Spoelstra. 2012. Predicting US primary elections with Twitter. In *Social Network and Social Media Analysis: Methods, Models and Applications, NIPS*.

[54] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. 2006. Large scale multiple kernel learning. *Journal of Machine Learning Research* 7, Jul (2006), 1531–1565.

[55] Ian Stewart, Yuval Pinter, and Jacob Eisenstein. 2018. Sío no, qué penses? Catalonian Independence and Linguistic Identity on Social Media. In *NAACL-HLT*.

[56] Sandesh Swamy, Alan Ritter, and Marie-Catherine de Marneffe. 2017. "I have a feeling Trump will win..................": Forecasting Winners and Losers from User Predictions on Twitter. In *EMNLP*. 1583–1592.

[57] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*. 1067–1077.

[58] Eftichia Teperoglou and Emmanouil Tsatsanis. 2014. Dealignment, Delegitimation and the Implosion of the Two-Party System in Greece: The Earthquake Election of 6 May 2012. *Journal of Elections, Public Opinion and Parties* 24, 2 (2014), 222–242.

[59] Adam Tsakalidis, Maria Liakata, Theo Damoulas, Brigitte Jellinek, Weisi Guo, and Alexandra Cristea. 2016. Combining heterogeneous user generated data to sense well-being. In *COLING*. 3007–3018.

[60] Adam Tsakalidis, Symeon Papadopoulos, Alexandra I Cristea, and Yiannis Kompatsiaris. 2015. Predicting elections for multiple countries using Twitter and polls. *IEEE Intelligent Systems* 30, 2 (2015), 10–17.

[61] Adam Tsakalidis, Symeon Papadopoulos, Rania Voskaki, Kyriaki Ioannidou, Christina Boididou, Alexandra I Cristea, Maria Liakata, and Yiannis Kompatsiaris. 2018. Building and evaluating resources for sentiment analysis in the Greek language. *Language Resources and Evaluation* (2018), 1–24.

[62] George Tsebelis. 2016. Lessons from the Greek crisis. *Journal of European Public Policy* 23, 1 (2016), 25–41.

[63] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting elections with Twitter: What 140 characters reveal about political sentiment. *ICWSM* 10, 1 (2010), 178–185.

[64] Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *NAACL-HLT*. 1268–1278.

[65] Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring User Political Preferences from Streaming Communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 186–196.

[66] Arkaitz Zubiaga, Bo Wang, Maria Liakata, and Rob Procter. 2017. Stance Classification of Social Media Users in Independence Movements. *arXiv preprint arXiv:1702.08388* (2017).