IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 10, 2021, accepted July 18, 2021, date of publication July 26, 2021, date of current version August 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099093

Improving Political Discourse Analysis on Twitter With Context Analysis

ARITZ BILBAO-JAYO AND AITOR ALMEIDA
DeustoTech, University of Deusto, 48007 Bilbao, Spain

Corresponding author: Aritz Bilbao-Jayo (aritzbilbao@deusto.es)

This work was supported in part by the Basque Government's Department of Education for the Deustek Research Group under Grant IT 1078-16 D, and in part by the Spanish Ministry of Science, Innovation and Universities through the Project Midiendo la competición política de las elecciones autonómicas en programas y twitter and the 10 años de Regional Manifestos Project under Grant RTI2018-095918-B-I00.

ABSTRACT In this study, we propose a new approach to perform political discourse analysis in social media platforms based on a widely used political categorisation schema in the field of political science, namely, the Comparative Manifestos Project's category schema. This categorisation schema has been traditionally used to perform content analysis in political manifestos, giving a code that indicates the domain or category of each of the phrases in the manifestos. Therefore, in this work we propose the application of this political discourse analysis technique in Twitter, using as training data of 100 publicly available annotated political manifestos in English with around 85,000 annotated sentences. Furthermore, we also analyse the improvement that using 5,000 annotated tweets could provide to the performance of the political discourse classifier already trained with political manifestos. Finally, we have analysed the 2016 United States presidential elections on Twitter using the proposed approach. As our main finding, we have been able to conclude that both datasets (political manifestos and annotated tweets) can be combined in order to achieve better results, achieving improvements in the F-Measure of more than 15 points. Moreover, we have also analysed if contextual information such as the previous tweet or the political affiliation of the transmitter could improve classifier's performance as it has already been proven for manifestos classification, introducing a novel method for political parties representation and finding that adding the previous tweet or the political leaning as contextual data does improve its performance.

INDEX TERMS Computational linguistics, data analysis, machine learning, natural language processing, text analysis.

I. INTRODUCTION

The rise of social media have offered both politicians and citizens new ways of interacting directly with each other without the direct mediation of traditional media. This phenomenon has allowed citizens to become participants in the construction of the political agenda, forcing political parties to use more direct means of communication than the mainstream press and media [1]. The most representative element of this paradigm is Twitter. Created in 2006, this social network has become one of the most important forms of communication between politicians and their electorate, reaching the point where some politicians bypass traditional media and exclusively release statements on social media. Furthermore, as all the members of these on-line social media platforms are treated as equals, any citizen can send a message to the politician, sometimes leading to a discussion between the politician and the citizens or between citizens themselves. Therefore, these social media platforms contain valuable data regarding citizens' concerns or the politicians' current talking points. In fact, as Dimitrova and Matthes [2] stated in their work analysing how social media behaves in Political Campaigning, *social media have become an indispensable part of modern political campaigning, both in the United States and internationally*.

However, since thousands of messages are created every hour [3], it is not feasible to manually analyse them. Thus, in order to analyse the political discourse in real time, the data analysing process has to be automated. For that purpose, in this manuscript we propose a multidisciplinary approach,

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

to build a text categorisation classifier using a publicly available dataset of manually annotated political manifestos. We have combined the political science knowledge from the social scientists involved in the annotation of the political manifestos with natural language processing, in order to be able to process large quantities of data and study how the political discourse evolves online.

The dataset of annotated political manifestos is available in the Comparative Manifestos Project(CMP)'s [4] website[1] where the categorisation schema (56 categories that represent different political ideas) and the annotation methodology [5] are explained. This methodology has been used worldwide for manifestos annotation by several research groups to later apply content analysis techniques in order to conduct political analyses such as: which are the policy preferences of a party regarding a specific topic or which topics are specially emphasised by the political party in the manifesto. For instance, Alonso *et al.* [6] analysed how some parties behave differently depending on the elections' context or Benoit [7] analysed how Irish political parties have varied their policy stances on European integration over the years.

However, in order to adapt political manifestos language to the language used in social media, we have annotated 5,000 political tweets from several British and North American politicians and analysed if fine-tuning with annotated tweets a classifier exclusively trained with annotated political manifestos improves its performance. Therefore, we go further than any other previous approach focused on analysing the online political discourse. Using up to 56 policy categories and avoiding the traditional left-right or democrat-republican statement analysis, we base our political discourse analysis on a widely used method by political scientists for political manifestos' content analysis. Moreover, we have checked if contextual information such as the previous tweet or the political affiliation of the transmitter could improve classifier's performance for tweets classification, proposing a novel approach of the representation of political parties based on their political orientation. The use of contextual data has already been proven to be useful when classifying political manifestos [8], achieving the state of the art results in 4 out 7 languages (among others, English). Finally, we have analysed with the introduced method the 2016 United States presidential elections on Twitter.

To sum up, the main contributions of this manuscript are: a novel approach for automatically classifying political tweets using a categorisation schema widely used by political scientists, a new representation method for the use of political parties as input feature in supervised classifiers using a disentangled representation based on their political orientation and a dataset of 5,000 tweets annotated with the previously mentioned annotation schema.

This paper is organized as follows. Section 2 offers an overview of previous related work on political social media analysis and the use of political manifestos as basis for the analysis of other types of political texts besides political manifestos. Section 3 describes our research framework: the dataset and designed neural networks architecture. Section 4 explains how the developed classifier has been evaluated and its results. In section 5, we explain a real use case of the proposed approach by analysing 2016 United States presidential election in Twitter. Finally, section 6 draws some conclusions and proposes further work.

II. RELATED WORK

Since its inception, Twitter has been seen by researchers of several fields as a new source of information with which they can conduct their researches. For instance, political scientists have identified Twitter as a platform where they can analyse what a subset of the population says without performing expensive surveys [9], study how politicians prioritise some topics over others or which ideas politicians want to send to their followers [10]. This phenomenon has offered to political scientists, on the one hand, and computer scientists on the other, a new research opportunity. In the first case, political science researchers have focused their work in manual approaches where each message or statement is manually analysed by a human, to later drawn some conclusions having as basis those manually annotated messages. On the contrary, computer scientists have taken a more automated approach where different aspects of the tweets are automatically analysed to later drawn conclusions from them. Either way, both approaches have led to a large number of research publications. In this research work we have aimed to combine both worlds and therefore, we have divided related work section in two parts: first, we have reviewed the most relevant works manually analysing the political discourse in social media and second, those research works using automated approaches.

On one hand, when it comes to manual approaches, we refer to those political analyses made with social media data (which have been probably gathered automatically using APIs or crawlers), but each of the posted message or tweet has been analysed manually, without the intervention of any supervised or semi-supervised tool.

Ramos-Serrano *et al.* [11] analysed the twitter activity of Spanish political parties during the 2014 European campaign. They manually analysed questions such as with whom are Spanish parties interacting, which topics are they tweeting about or what was the function of politicians tweets. Authors found that Spanish conservative parties retweeted less messages than the rest of the parties, while new parties retweeted messages the most. With regard to the topics of the tweets, "*Campaign and Party Affairs*" was the main topic. Then, topics such as "*Europe*", "*Corruption*" (mainly by minor parties) and "*Nationalism*" (by centre-right parties) were the most treated during campaign. In total, authors analyses 21 different topics.

Lopez-Garcia [12] studied the 2015 Spanish general elections campaign in Twitter. In particular, the research work focused on performing a quantitative analysis of main political parties' candidates' tweets. The content analysis consisted

[1] https://manifesto-project.wzb.eu/

in classifying politicians tweets in 4 different categories: political, policy, campaign and personal, far from the CMP's 56 categories designed for political content analysis.

Casero-Ripollés *et al.* [13] analysed the messages sent by the political party Podemos-We can. The authors performed a quantitative analysis focusing on the issues and functions of the messages sent by the party. On one hand, in order to study the functions, the authors created an ad hoc taxonomy of 13 categories: agenda and organization of political actions, electoral program, management of political achievements, criticizing opponents, etc. On the other hand, another taxonomy of 18 elements was created in order to study the topics of the tweets. Among the most relevant categories were: economy, social policy, science and technology, state territorial model, or relationship with the media.

Following the same taxonomy of 13 categories previously introduced, López-Meri *et al.* [14] performed a quantitative content analysis of the 2016 Spanish electoral campaign, studying the tweets published by the four parties and corresponding candidates that receive the most votes. Among the findings, the manuscript revealed that first, most of political parties effort was focused on the dissemination of their political proposals. Also, a low degree of personalization was detected in politicians' tweets. Finally, the authors found that political parties tried to combine both old and new media in a complementary way.

In [15] Alonso-Muñoz and Casero-Ripollés extended the work made in [14], answering some new questions such as which was political parties' agenda, what was more relevant for Spanish political actors to share their programmatic proposals or to follow a strategy to obtain votes, and finally, to analyse which topics received the best response from the public.

Russell [16] studied the U.S. Senators party polarization identifying those messages with a partisan rhetoric. To do so, Russell catalogued U.S. senators Twitter activity during the first 6 months of the 113th (Democratic majority) and 114th (Republican majority) congresses reaching interesting outcomes. Reference [16] analysed two congresses with different majorities expecting changes in political parties' rhetoric. As it is stated in [16], when this manuscript is being written, the political situation in the United States is highly party-polarized. Having this a fact, Russell categorised tweets sent by Democrat and Republican senators in the previously mentioned period in a partisan, non-partisan classification. To clarify, partisan rhetoric could be defined as those statements praising their own political parties or criticising the opponent's parties. In 2013, with the democrats as majority, 17.3% of Republicans tweets contained partisan rhetoric, unlike Democrats, where 4.5% included this rhetoric. In 2015, even though the majority shifted towards Republicans, they maintained as the party with most partisan rhetoric, 11.75%, in contrast of the 5.43% of Democrats messages. Moreover, the manuscript analysed if those partisan tweets were positive or negative, concluding that two thirds of Republicans' partisan tweets included negative rhetoric, this percentage decreased to 50% with Democrats. All these partisan rhetoric is related to the *Political Authority* category in CMP which is used in order to analyse the political discourse in Section V.

To sum up, all the reviewed manual approaches have used different categorisation schemas for several purposes with diverse number of topics, being most of those schemas created ad hoc for the research: 21 in [11], 4 in [12] or 13 [13]–[15]. All these schemas are far from the 56 categories of the CMP, which have been already used for manifestos analysis and offer a more in-depth analysis due its low level granularity.

On the other hand, automated approaches have been used for several tasks related to political analysis in social media. For instance, detecting the political orientation of a sentence [17] or to classifying tweets on democratic or republicans. [18]. However, these approaches rely on data created in its entirety in Twitter. Unfortunately, this data gathered from Twitter could have been manipulated by third party actors or institutions. As Ratkiewicz *et al.* [19] introduce in their study about astroturfing in political campaigns on Twitter, there are individuals whose objective is to launch controlled campaigns in favour or against a precise political organization, candidate or idea using centrally-controlled accounts. Other researchers have worked detecting rumours [20] in social media which may introduce new topics of conversation to the network or influence user's opinion about some subjects.

Therefore, in this work we have focused our work on the most reliable political data Twitter contains: messages sent by politicians and political parties. Thus, from now on, the state of the art analysis will be centred on research works using this type of reliable data.

The first example of this kind of political analysis on Twitter using reliable data is [21]. Stier *et al.* analysed the 2013 German federal election campaign in Twitter and Facebook, studying how aligned are the topics discussed by politicians compared to the most important topics for the electorate according to a survey, and how their communication strategy vary depending on the social media platform where ideas are spread. To do so, the authors classified the tweets on topics using a human-interpretable Bayesian language model. The topics were defined by known survey classes and additional social-media-specific topics. They used the German Longitudinal Election Study Survey that collected the opinion of 7,882 people before and after elections. In particular, Stier *et al.* coded the open-ended responses with GLES [22] categorisation schema which consists in three high level dimensions, politics, polity and policy, ending with a total of 18 topic classes. With regard to the gathered social media data, the authors collected Twitter and Facebook posts from candidates and social media users. However, even though authors gathered data from both candidates and social media users, authors split their findings depending on the used data, therefore the findings obtained exclusively using candidates messages could be taken into account. Among their findings, the most noteworthy discoveries are:

- Politicians prefer Twitter over Facebook to comment events such as TV debates.
- Politicians use Twitter and Facebook differently. Whereas Facebook is used to mobilize users to attend campaign celebrations or similar events, Twitter is used for political debates where politicians discuss about several policies giving their own opinion. This is relevant for the proposed approach in this research work, since it validates our decision of choosing Twitter as the analysed social media platform.
- Politicians discuss different topics with respect to the priorities shown by electors on the surveys.

Yaqub *et al.* [23] analysed the 2016 US presidential elections' political discourse on Twitter from two points of view: studying public opinion gathering Tweets of over a million users in order to identify their talking points and behaviour(if they share original opinions, interact with other people, etc.) and, analysing the sentiment of the tweets sent by the Republican and Democrat presidential candidates. In this case, we are going to focus in the latter analysis as it has been previously mentioned. They assigned to each candidates tweets a sentiment score using a tool named SentiStregnth. Among their conclusions, the most noteworthy are that Donald Trump offered more optimistic messages than Hillary Clinton, with an average sentiment score of 0.3925 versus the negative average sentiment score of Hillary Clinton, −0.0125. They also performed a very simple analysis of the most frequently used terms by the candidates: Hillary, Donald/Trump and Vote from Hillary and Thanks, Hillary/Clinton and Great from Trump. In both cases, when a candidate was referring to the other, the sentiment average score was negative, confirming that both candidates used partisan rhetoric.

In conclusion, the number of categories used in both manual and automated approaches is far from the 56 categories presented in CMP. Moreover, most of the analysed works have designed their coding schema for specific tasks or goals, whereas CMP's categorisation schema allows the analysis in different areas with the advantage of already having annotated datasets.

Regarding the use of annotated political manifestos in order to create natural language processing models for the analysis of other types of political texts beyond political manifestos, several works have arisen recently. Nanni *et al.* [24] used annotated political manifestos and speech to analyse the speeches from the last three US presidential campaigns using the 7 main political domains defined by the Manifestos Project. Bilbao-Jayo and Almeida [25] analysed the political discourse of the Spanish 2015 and 2016 general elections in Twitter, exclusively using as training data manually annotated political manifestos and a simplified political message taxonomy. In [26], Nanni *et al.* used English political manifestos to measure the level of Euroscepticism transcripts of speeches from the European parliament. To do so, they only used the relevant policy categories for this task: European Community/Union (Positive and Negative) and National Way of Life(Positive and Negative). 4 categories out of the 56 categories we are working with in this research work.

III. POLITICAL DISCOURSE ANALYSIS

During this section the proposed approach for automated political discourse analysis in social media will be extensively explained. First, CMP's categorisation schema and dataset will be explained in order to give an overall view of the proposed approach. Then, we are going to describe the annotation methodology followed for the annotation of the 5,000 political tweets. Later, we are going explain which types of contextual information have been used as additional data, to finally explain how Convolutional Neural Networks (CNNs) and Bidirectional Encoder Representations from Transformers (BERT) have been adapted for this task.

A. THE ANNOTATED MANIFESTOS PROJECT DATASET

The CMP is one of the most ambitious and accurate attempts done by political scientists to perform content analysis of parties' electoral manifestos to later derive policy positions of each political party depending on what each party claim in their manifestos.

The precursors of this methodology were the Manifesto Project, formerly known as the Manifesto Research Group (MRG), and nowadays as Comparative Manifestos Project (CMP) [27]. In 2001, they created the Manifesto Coding Handbook [28] which has evolved over the years. The handbook provides instructions to the annotators about how political parties' manifestos should be coded for later content analysis and a category schema that indicates the set of codes available for codification. Nowadays, the category schema for manifestos annotation consists in 56 categories (see Table 1) grouped into seven major policy areas [29]): *External Relations*, *Freedom and Democracy*, *Political System*, *Economy*, *Welfare and Quality of Life* and *Social Groups*. Moreover, recently the CMP has added new subcategories for manifestos from countries which have recently transitioned or are transitioning from authoritarian regimes to a democratic system.

The annotation process is a two-step task: unitising and coding. Unitising consists in splitting each manifestos' text into quasi-sentences or coding units. Since one full sentence can contain more than one statement or message, there are some cases where a sentence has to be split into more than one quasi-sentences where each quasi-sentence contains a different message. Once the text has been unitised, a category is assigned to each of the quasi-sentences.

The dataset of political manifestos used in this research is the public CMP's dataset. We have downloaded from their website and preprocessed (remove the stopwords, convert all the text to lowercase and tokenize the sentences) 115 manifestos with 86,500 manually annotated sentences in English from several countries (Australia, South Africa, United Kingdom, United States, etc.). Regarding the distribution of sentences per category, the seven major policy areas are distributed in the following way: external relations (6.5%), freedom and democracy (4.42%), political system (10.64%), economy (25.45%), welfare and quality of life (31.77%), fabric of society (11.2%) and social groups (9.99%). As it can be seen, the distribution of samples over the seven domains is highly imbalanced. Therefore, the distribution of samples over the 56 categories or subdomains is even less balanced, having some of the subdomains less than 1% of the samples.

B. ANNOTATED POLITICIAN'S TWEETS

In order to evaluate the performance of the proposed approach for political discourse analysis in social media, we have annotated 5,000 tweets using CMP's categorisation schema. To do so, we downloaded the last 3,000 tweets from the Twitter accounts of politicians from the United Kingdom and United States. The time period of the downloaded tweets varies depending on the amounts of tweets published by each of the politicians since Twitter's API limited us. The API only allows retrieving the last 3000 tweets of each Twitter account. Therefore, our dataset contains tweets from 2011 to 2019. We used two publicly available twitter-lists to gather them: *cpsan/members-of-congress*[2] and *twittergov/uk-mps*.[3] Then, we randomly selected 5,000 tweets to annotate them. It is important to note that Manifestos Project's categorisation schema was designed to annotate each sentence's topic inside the political manifesto. However, when it comes to tweets, our goal is to classify the whole tweet in one of the CMP's categories, avoiding the categorisation of each of the sentences that a tweet can contain. Therefore, when it comes to annotating the tweet, we have selected the topic that best summarises the tweets' meaning. However, in those tweets containing more than one concept, we have added some extra categories apart from the most important one in order to analyse the feasibility of transforming this multiclass classification problem to a multi-label classification one. Also, it should be mentioned that each of the tweets has been anonymized, in other words, the annotator was not aware of who had post the tweet during the annotation process in order to avoid any bias introduced by the annotator.

As it happens in political manifestos, the distribution of samples over the seven domains is highly imbalanced (as it can be seen in Figure 1): external relations (10.61%), freedom and democracy (5.58%), political system (15.16%), economy (16.35%), welfare and quality of life (28.59%), fabric of society (15.18%) and social groups (8.52%). When it comes to the distribution of subdomains', as it can be seen in Figure 1, the 59.08% of the samples are divided in 10 categories, whereas the rest of the samples, 40.92% are divided in the remaining 46 categories. Therefore, the most repeated categories in a descending order are: Political Authority (305), Welfare State Expansion (504), Equality (503), Environmental Protection (501), Law and Order (605), Technology and Infrastructure (411), Labour Groups: Positive (701), Market Regulation: Positive (403) and Incentives: Positive (402).

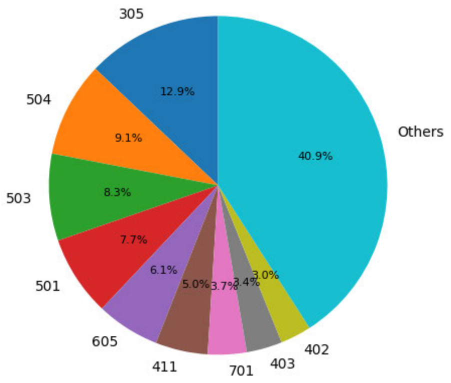


FIGURE 1. Subdomain distribution of annotated tweets.

With regard to the preprocessing of the annotated tweets, they have been preprocessed for the experiments with CNNs: removing stopwords and URLs, converting all the text to lowercase, tokenizing the sentences and maintaining hash-tags and user-names. However, in the case of BERT, no preprocessing has been performed, since the used word segmentation technique, WordPiece, deals with this.

C. USING CONTEXTUAL INFORMATION FOR POLITICAL DISCOURSE ANALYSIS

Based on the fact that it has already been proven by Bilbao-Jayo and Almeida [8] that contextual information such as the previous statement and the political leaning of the manifestos improve the performance of automatic political manifestos, in this research work we have followed a similar strategy: to use contextual information in order to improve the proposed tool for political discourse analysis.

The reason why the previous phrase was chosen is due to how political manifestos are annotated. During the annotation process, sentences containing more than one idea or category are divided into quasi-sentences. Therefore, it may happen that quasi-sentences of very few words without any other information are impossible to classify correctly without additional context which in this case would be the previous quasi-sentence. Therefore, a similar approach is also usable in Twitter where due to the character limitations of Twitter, a message sent by a politician or political party could take more than one tweet, creating a thread of tweets. Therefore, knowing the previous tweet could give some insight about what is talking about and clarify the meaning of the analysed tweet.

The second contextual data is the sender of the message. In the case of the manifestos, the sender is the political party who has written it. Conversely, on Twitter, tweets can be sent by political parties' official twitter accounts or by politicians who are part of a political party. Therefore, we have to represent the sender of the message in a way usable in both worlds, manifestos and Twitter. Thus, even though there are some cases where politicians' language or discourse may

[2] https://twitter.com/cspan/lists/members-of-congress

[3] https://twitter.com/twittergov/lists/uk-mps

TABLE 1. Categories in seven policy domains [29].

Domain 1: External Relations	**Domain 5: Welfare and Quality of Life**
101 Foreign Special Relationships: Positive	501 Environmental Protection: Positive
102 Foreign Special Relationships: Negative	502 Culture: Positive
103 Anti-Imperialism: Positive	503 Equality: Positive
104 Military: Positive	504 Welfare State Expansion
105 Military: Negative	505 Welfare State Limitation
106 Peace: Positive	506 Education Expansion
107 Internationalism: Positive	507 Education Limitation
108 European Integration: Positive	**Domain 6: Fabric of Society**
109 Internationalism: Negative	601 National Way of Life: Positive
110 European Integration: Negative	602 National Way of Life: Negative
Domain 2: Freedom and Democracy	603 Traditional Morality: Positive
201 Freedom and Human Rights: Positive	604 Traditional Morality: Negative
202 Democracy	605 Law and Order
203 Constitutionalism: Positive	606 Civic Mindedness: Positive
204 Constitutionalism: Negative	607 Multiculturalism: Positive
Domain 3: Political System	608 Multiculturalism: Negative
301 Decentralisation: Positive	**Domain 7: Social Groups**
302 Centralisation: Positive	701 Labour Groups: Positive
303 Govern. and Admin. Efficiency	702 Labour Groups: Negative
304 Political Corruption: Negative	703 Agriculture and Farmers
305 Political Authority: Positive	704 Middle Class and Professional Groups: Positive
Domain 4: Economy	705 Minority Groups: Positive
401 Free-Market Economy: Positive	706 Non-Economic Demographic Groups: Positive
402 Incentives: Positive	
403 Market Regulation: Positive	000 No meaningful category applies
404 Economic Planning: Positive	
405 Corporatism: Positive	
406 Protectionism: Positive	
407 Protectionism: Negative	
408 Economic Goals	
409 Keynesian Demand Management: Positive	
410 Economic Growth	
411 Technology and Infrastructure: Positive	
412 Controlled Economy: Positive	
413 Nationalisation: Positive	
414 Economic Orthodoxy: Positive	
415 Marxist Analysis: Positive	
416 Anti-Growth Economy: Positive	

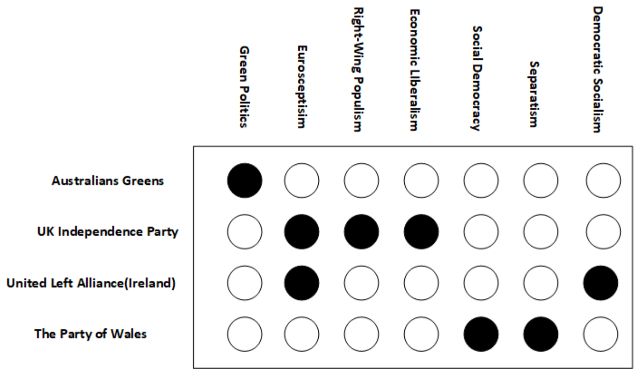
differ, we have decided to represent each politician as its political party, supposing that most of the politicians will have a similar discourse to that of his/her political party.

Moreover, we introduce a novel approach for political parties orientation based on their political orientation.

Thus far, political parties have been represented using a one hot encoding representation. This method represents each categorical variable as a list of 0s with a length equal to the number of categorical variables to represent. Then, in order to have unique representation of each variable one of the 0s is replaced by 1 and in the end, each variable will have the non zero value in one specific position which indicates which categorical variable is representing the encoding. For instance, if there were two parties, there would be an array of size 2, [1, 0] representing the first party and [0, 1] the second one. However, this approach has a priori two major drawbacks. First, it does not provide any information regarding parties political orientation and therefore, each party is equal to each other at the beginning of the training process even though they are diametrically opposed. Second, since the number of political parties has to be defined before training the model, every time a manifesto of a new political party wants to be added to the model, it would have to be retrained from scratch. Furthermore, manifestos of new political parties could not use this contextual information because those new parties would be unknown for the model, which derives in a scalability issue.

Therefore, in order to address this issue we propose a new method: using parties' political orientation to build a disentangled representation of the parties. We have extracted each political party's political orientation for European political parties from [30], a guide with the parliamentary elections and governments since 1945 where more than 700 parties are listed with their respective political orientations, and from Wikipedia for the rest of world parties, obtaining only those orientation with references. This approach is based on the concept of disentangled representation [31], distributed representations whose latent variables (dimensions of the vector) are semantically interpretable. In this case, a disentangled representation has been used in order to encode political parties using their political orientation. Therefore, each possible political orientation will be a dimension in the vector which represents the party and if the party follows a particular political orientation, the dimension corresponding to the orientation will be activated in the parties' representation.

For explanatory purposes, a small example where three parties are codified will be introduced. Assuming that there are only 7 possible political orientations (see Figure 2), Green Politics, Euroscepticism, Right-Wing Populism, Economic Liberalism, Christian Democracy, Separatism and Democratic Socialism, we want to represent the following political parties: Australian Greens (Green Politics), UK Independence Party (Euroscepticism, Right-Wing Populism, Economic Liberalism) and United Left Alliance (Euroscepticism and Democratic Socialism). Each of the vectors representing the parties would have 7 dimensions (one per possible political orientation). Australian Greens would be represented as [1, 0, 0, 0, 0, 0, 0],



	Green Politics	Euroesceptisim	Right-Wing Populism	Economic Liberalism	Social Democracy	Separatism	Democratic Socialism
Australians Greens	●	○	○	○	○	○	○
UK Independence Party	○	●	●	●	○	○	○
United Left Alliance(Ireland)	○	●	○	○	○	○	●
The Party of Wales	○	○	○	○	●	●	○

FIGURE 2. **Disentangled representation of 3 known parties and 1 unknown party for the model.**

UK Independence Party [0, 1, 1, 1, 0, 0, 0] and United Left Alliance [0, 1, 0, 0, 0, 0, 1]. This allows the addition of new political parties which were not in the training process. For instance, if we want to add The Party of Wales (Social Democracy and Separatism), it would be simple, [0, 0, 0, 0, 0, 1, 1].

D. ADAPTING CLASSIFIERS FOR THE POLITICAL DISCOURSE CLASSIFYING

In order to build the proposed political discourse classifier we have evaluated our approach adapting two different text categorisation models: CNNs with Word2Vec embeddings and BERT.

1) CONVOLUTION NEURAL NETWORK FOR TEXT CLASSIFICATION

On one hand, we have used CNNs which have achieved excellent results in several text classification tasks such as [32], [33] or [34]. This, combined with the fact that this type of classifier allows the extraction of knowledge from non-annotated texts using word embeddings which later are fine-tuned to the task, has resulted in a competitive deep learning architecture.

First, the flowchart of the model will be explained as an introduction (see Figure 3) and after that, the model is explained in more detail. The simplified flowchart of the model is the following:

1) The phrase and the previous phrase are inserted as a list of words.
2) The embedding matrix replaces each word with its corresponding word vector, generating a sequence of word vectors from a sequence of words.
3) The phrase and the previous phrase are fed into two different structures of convolutional neural networks with 100 filters and filter sizes of $2 \times d$, $3 \times d$ and $4 \times d$, being d the dimension of the word embedding.
4) The 1-max-pooling reduces the dimensionality of the feature maps generated by each group of filters.
5) Once their dimensionality has been reduced, the feature maps generated from the phrase and the previous phrase are concatenated.
6) If the political party to which the text belongs to is used, its representation is concatenated with the feature extracted from the CNNs.
7) A dropout rate of 0.5 is applied to the concatenation between the extracted features and the representation of the party.
8) Then, to classify the phrases to the objective political topics, a fully connected layer with ReLu as activation function is used.
9) A dropout rate of 0.5 is applied to the fully connected layer.
10) The fully connected layer with softmax as activation function computes the probability distribution over the labels.

The inputs of the model are the sentences (from manifestos or tweets) which are fed to the neural network as sequences of words. These sequences have a maximum length of 60 words. The maximum length has been decided after an analysis of the corpus' sentences' length and detecting that most of the sentences have 60 or less words.

However, the words are not provided as raw text to the convolutional neural network. The words are presented as word vectors, a multidimensional representation of each word. Those word vectors have been generated using the Word2Vec [35] unsupervised learning algorithm, which produces a large vector space having non-annotated raw text as input. Using Word2Vec, each word of the corpus is positioned in a multidimensional vector space taking into account its context (its surrounding words). Word's position in the N-dimensional vector space (being N the number of dimensions of the defined vector space) is used as its representation (word vector).

For example, given a sentence $S = [w_1, w_2, w_3 \ldots w_n]$ (n is the number of words in the sentence), the context of the word w_i would be $Context_k(w_i) = [w_{i-k}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+k}]$ where $2k$ is the window size for the context. Then, the log-likelihood is maximized in order to compute the word vector of each word:

$$J_{ML} = logP(w_i|Context_k(w_i))$$

300 has been chosen as word vectors' size (number of dimensions of the multidimensional space where the words are positioned) to take advantage of already pre-trained Word2Vec models. In this case, we have used a Word2vec model pretrained with Google News corpus (3 billion running words).

Once all the word vectors have been computed, the following operation is performed. First of all, a dictionary D where words are mapped to indexes $(1, \ldots, |D|)$ is created, being $|D|$ the number of unique words in the corpus and saving the 0 index for padding purposes. Therefore, the input sequences of words are transformed into a sequences of 60 indexes, padding with 0s those phrases which have a length of less than 60 words, since CNNs does no admit different sizes for the input data once the input size has been set. Then, these indexes are transformed into their corresponding word vector using an embedding layer or matrix. This embedding matrix acts as a dictionary: having the word index, the embedding matrix returns the corresponding word vector which has been previously computed. The embedding matrix is generated concatenating all the vector representations of all the existing words in D, creating a matrix $W \in \mathbb{R}^{|D| \times d}$, where d represents the vector size of the word embeddings which is 300 in this research.

Therefore, the embedding matrix works as a dictionary whose input is the word index and its output is the vector representation of the word. The embedding matrix can be both static or non-static. On one hand, the static approach treats all the word vectors as static values which cannot change through the training process and therefore all those weights per word defined by Word2Vec remain constant through all the training. On the other hand, a non-static embedding matrix changes as the training process evolves since the word vectors are interpreted as new parameters for the model and they are fine-tuned during the training. In this case, non-static Word2Vec word-embeddings have been used since it improves the model's performance [32].

Once the phrase has been transformed from a sequence of words to a sequence of word indexes and finally to a sequence of word vectors, the phrase can finally be fed into the convolutional neural network, since the sequence of word vectors are in fact a matrix which dimensions are $60 \times d$ where convolution operations can be performed.

CNNs are a specific type of neural networks with neurons, weights and biases where convolution operations are performed and have been traditionally used for recognizing visual patters directly from images (pixels) [36]. However, as previously has been explained, in recent years, CNNs has also been used for text classification. In brief, convolution operations consist in moving different windows (filters made of neurons) with different sizes, s (filter sizes) analysing different regions in the matrix (an image or a list of word vectors) to extract different features. The proposed model performs convolution operations with 3 different filter sizes, batch normalization [37] and ReLU as the activation function. Batch normalization acts as an extra regularizer and increases the performance of the model.

The defined filter sizes are $2 \times d$, $3 \times d$ and $4 \times d$. These filter sizes can be compared to a selection of n-grams: bigrams, trigrams and fourgrams respectively. Therefore, each row in input matrix of the sentence or tweet represents a word and therefore a filter size of $2 \times d$ will take the whole width of all the possible bigrams of the sentence, filter size of $3 \times d$ all the possible trigrams and filter size of $4 \times d$ all the possible fourgrams. This is how a single filter would work, however, as it is stated in [38], multiple filters should be used in order to learn complementary features. The model has 100 filters per different filter size. Once a filter has been applied, a feature map is generated. Therefore, a different feature map is generated per applied filter.

Then, the following operation is performed once per filter, being the filter size fs, embeddings dimensionality d and phrases length p, the input sentence is the matrix $S \in \Re^{p \times d}$. Thus, the convolution can be represented as:

$$O_j = f(W_j \circ [_1, \ldots, s_{p-fs+1}] + b) \tag{1}$$

$O_j \in \Re^{p-fs+1}$ is the result of the convolution. W_j and b are the parameters that are being trained. $f()$ is the activation function for the convolution, which in our case is a ReLU activation [39]. Finally, $W \circ S$ represents the element-wise multiplication of the elements. Being the number of filter maps d_o, the output of the convolution is $O = [O_1, \ldots, O_{d_o}] \in \Re^{(p-fs+1) \times d_o}$.

After the convolutional layer, there is a pooling layer whose objective is to reduce the dimensionality of the incoming data. There are different pooling strategies: average pooling, max-pooling, 1-max-pooling, etc. We have opted for the 1-max-pooling [40] strategy since it has been proved in [38] that is the best approach for natural language processing tasks. It captures the most important feature (the highest value) from each of the feature maps. Therefore, the output of the pooling is a feature per filter which are later concatenated into a feature vector.

Next, a dropout [41] rate of 0.5 is applied as regularization in order to prevent the network from over-fitting, followed by a fully connected layer with ReLU as the activation function and batch normalization. Then a 0.5 dropout is applied [38]. Finally, the softmax function computes the probability distribution over the labels.

The categorical cross-entropy loss has been used as training objective function since it supports multiclass classifications. The optimization has been performed using Adam [42] with the parameters of the original manuscript.

Regarding how the previous phrase has been added to the model as a new input in order to improve the performance of the model as it will be demonstrated in Section IV, we have replicated for the previous tweet, the same convolution-pooling process it is used in the actual tweet that is going to be analysed.

With regard to the political leaning, it is represented with using one-hot encoding representation or the disentangled representation of the party using its political orientation (each representation method is explained in Section III-C).

2) BERT

On the other hand, BERT proposed by [43] has meant a considerable improvement on the NLP field. When BERT was presented, this new NLP model achieved state of the art results on eleven NLP tasks without the need to make substantial changes to the architecture. Moreover, BERT has been the first really successful attempt of transfer learning in NLP, a technique that had been successfully applied on other tasks such as computer vision but similar performances had not being achieved for NLP problems.

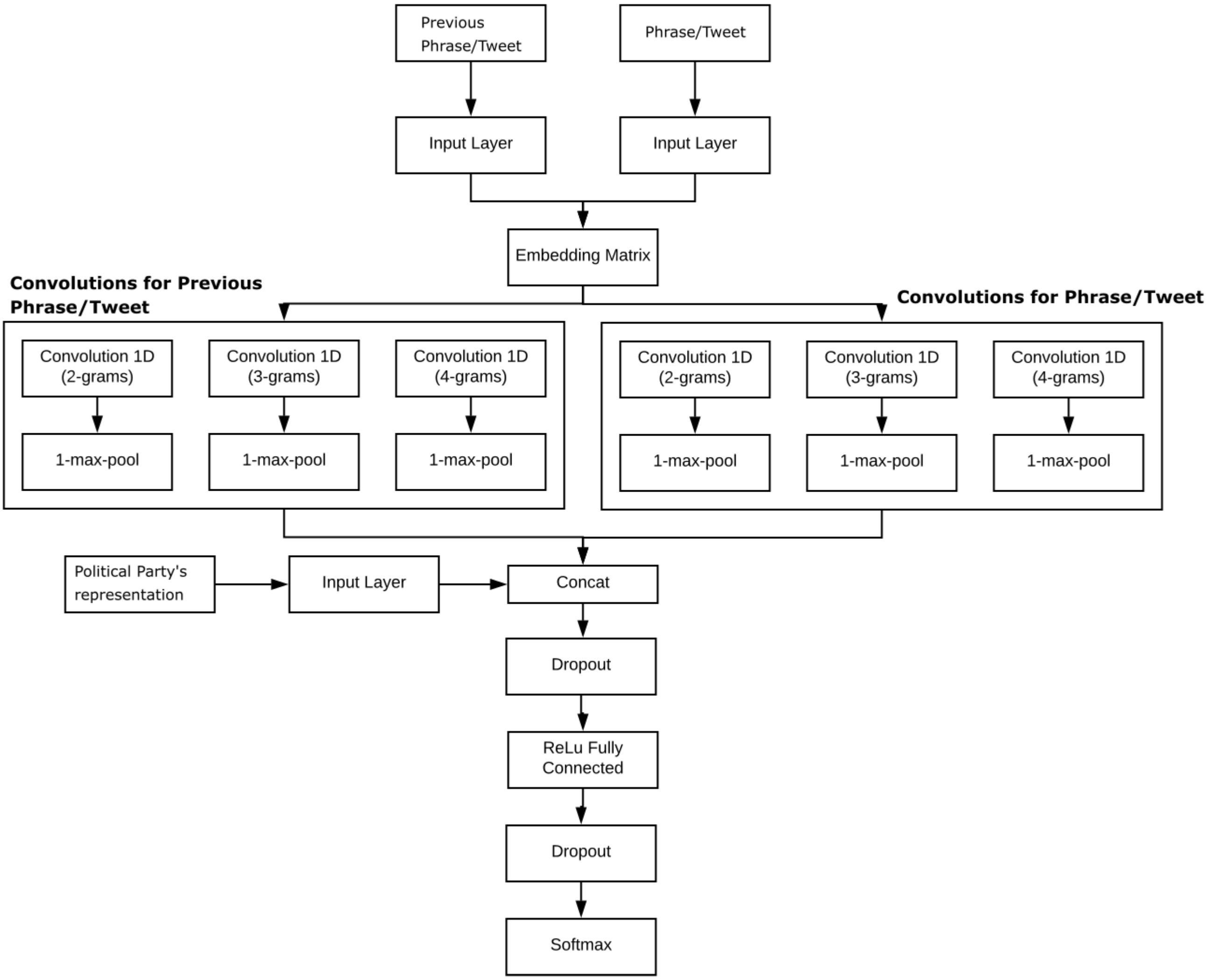


FIGURE 3. Designed architecture with convolutional neural networks and Word2Vec embeddings for political discourse classification using two types of contextual data: the previous phrase or tweet and the political party.

In particular, BERT is a pre-trained language model (LM). LMs have already shown their effectiveness on improving other model's performance in several NLP task as it is stated in [43]. There are two approaches when it comes to using pre-trained language representations or models in other tasks: feature based and fine-tuning. Feature based approaches use architectures specifically designed for the task including pre-trained representations as features, whereas fine-tuning approaches have very few task specific parameters.

However, according to Devlin *et al.* both approaches share a major limitation: most of the models are unidirectional (GPT [44]) or use shallow concatenations of left to right and right to left unidirectional language models such as ELMO [45]. The authors give as an example of this phenomenon OpenAI's GPT [44], where each token is only able to see the previous tokens in the self attention layers of the used Transformer [46]. However, according to them it is necessary to use context from both directions as BERT does.

In order to pre-train BERT the authors used two corpora: the BookCorpus(800M words) [47] and the English Wikipedia (2,5000M words). Both corpora contain text at document level, something essential according to the authors, since sentence level corpora does not have the same performance as these type of corpus, since they do not provide the necessary long contiguous sentences.

BERT is pre-trained using two unsupervised tasks with the previously mentioned datasets: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP).

MLM consists in randomly masking 15% of the input tokens to later predict the masked word using the left and right context. Once a predefined percentage of the input tokens are masked, the model (BERT in this case) is trained to predict which word the *[MASK]* token is replacing. Therefore, in this case the final layer of the model is a softmax of the same size as the vocabulary where a vocabulary ID corresponding to the replaced word is predicted.

However, the masking technique can only be applied in the pre-training process, when the model is fine-tuned words are not masked because otherwise, in other tasks, such as sentence classification or machine translation, valuable information would be lost if some word were replaced by the masking token. Therefore, in order to ease the this issue, the words chosen to be masked are not always replaced by *[MASK]*. 80% of the time are replaced by the *[MASK]* token, 10% by a random token and 10% the token remains unchanged maintaining the original token.

NSP consists in training the model to understand the relationship between two sentences for tasks such as Question Answering (QA) and Natural Language Inference (NLI). To do so, they created a corpus for a next sentence prediction task. The corpus was created randomly choosing sentences (A), being each sample a training instance, and assigning to each A, a B sentence which 50% of the times was the true next sentence and 50% a random sentence extracted from the corpus. This pre-training task has been demonstrated to be beneficial for QA and NLI tasks.

With regard to how the input sentences are processed, BERT uses a tokenization technique called WordPiece Model (WPM) [48]. This segmentation technique was designed in order to tokenize input sentences in a deterministic way dealing with out of vocabulary words. To do so, words are divided into wordpieces or subwords that can be reverted to their original form using reserved boundary symbols. In this manner, unknown words can be decomposed into known subwords and some knowledge can be extracted from them. In particular, BERT has a 30,000 token vocabulary with some reserved special tokens such as [*CLS*] which is always the first token of each sequence or [*SEP*] to divide sentence pairs.

However, once the input sentence has been tokenized, for each of the given tokens an input representation must be built. This new token representation is constructed adding three different embeddings as it can be seen in Figure 4. The token embedding represents the semantic meaning of the token on a multidimensional space; the sentence embedding indicates if the token belongs to the first sentence or to the second; finally, transformer positional encoding indicates the order of the token inside the sequence of tokens.

Therefore, the first part of BERT's architecture that should be explained is the Transformer, its core module. The transformer is based on the use of self-attention for training and modelling of sequences (machine translation, language generation, etc.) without using recurrent models such as RNNs or LSTMs. To do so, they use an encoder-decoder architecture. However, BERT only uses the encoder side of the transformer. Just as BERT, the transformer encoder's needs positional encodings in order to know the place of each token in the sequence since no recurrence of any kind is used. Apart from some normalization and feed forward layers, there is a structure named Multi Head Attention inside the encoder which is the most important element of the encoder. Each of this Multi-Head attentions implement an attention technique named *Scaled-Dot Product Attention*. This attention method consist in three inputs matrices: queries (Q) and keys (K) of dimension d_K and values of dimension d_v. Then, this attention mechanism is replicated N times (or N heads) in order to learn different features from each attention mechanism.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Devling *et al.* built two different BERT model sizes. BERT-Base was built in order to be comparable to other approaches in the state of the art in term of parameters (12 stacked encoders, 12 self attention heads and 110M parameters); and, BERT-Large to obtain state of the start results (24 stacked encoders, 16 self attention heads and 340M parameters).

In this work we have used the BERT-Base model due to technical limitations. However, even though we have used the BASE model we have not been able to fine-tune all the model as [43] recommends in their manuscript due to again, hardware limitations. Therefore, all the results given in Section IV have been computed with all the layers frozen except the last (12th) encoder which is fine-tuned.

Finally, regarding how BERT is converted into a model to solve multiclass classification problems, a softmax function has to be added at the end of the 12th encoder whose output is (128, 768), where 128 is the number of words and 768 is the size of the hidden state which represents each word. These values are predefined by BERT-BASE. As is in the case of CNNs, categorical cross-entropy loss has been used as training objective function and Adam as optimizer. Conversely, in order to adapt BERT to a multi-label classification problem a sigmoid function has been used instead of softmax, and binary crossentropy as loss function.

With regard to how the previous tweet has been added to the model, we have taken advantage of BERT's design. Bert allows an input pair of sentences and using the sentence embedding shown in 4, is able to differentiate between the introduced pair of sentences. Also, BERT adds the [SEP] token in order to distinguish the sentences.

Regarding the political leaning, in this case, the representation of the party (each representation method is explained in Section III-C) is concatenated to the parameters coming from the last encoder. Once the output of the encoder and the representation is concatenated, the probability distribution over the target labels is computed using softmax for multiclass classifications and sigmoid for multilabel.

IV. EVALUATION

This section has been divided in two parts. First, we are going to evaluate the proposed method for political parties representation comparing the results achieved with this approach with the state-of-the-art results for automated manifestos classification to verify if this new approach is better or comparable to the one-hot-encoding representation. Second, the evaluation with annotated tweets has been made. Similar to manifestos, first of all we have evaluated the performance of our model addressing the problem as a multi-class classification task. However, as a tweet may contain more than one idea, we have

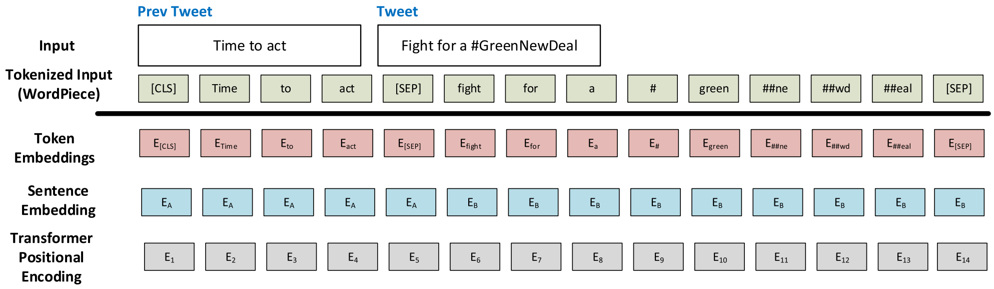


FIGURE 4. Example of how a tweet and its previous tweet would be fed to BERT. Based on the figure shown in [43].

also evaluated the tweets' task as a multi-label classification problem, where out of N classes, at least one class has to be selected, in other words, more than one class can be assigned to a tweet. We run all experiments on a single NVIDIA GTX 1080.

To conclude the introduction of this section, it should be clarified how the evaluation has been performed. Due to the imbalanceness of the datasets and a large number of categories, the results have been presented using three different measures: accuracy rate, F-Measure (Macro) and Geometric-Mean.

A. EVALUATION OF THE NOVEL APPROACH FOR POLITICAL PARTIES REPRESENTATION USING THEIR POLITICAL ORIENTATION

In order to evaluate the proposed approach for political parties representation and obtain comparable results to those obtained in the previous work where manifestos were automatically annotated using contextual information, we have replicated the architecture and evaluation methodology with 5-fold cross validation.

Moreover, we have also performed the same experiments with BERT in order to analyse if adding contextual information to this language model still improves its performance.

As it can be seen in tables 2 and 3, the proposed disentangled representation for political parties based on their political orientation improves the performance compared with the baseline without contextual data for the classification of domains and subdomains with both text classification models, CNNs and BERT. However, there is not a remarkable improvement when its results are compared with the metrics obtained with the one-hot-encoding representation.

Nonetheless, as it has been previously explained in Section III-C, one of the advantages that this approach could have is the easy addition of new political parties to the designed tool, without the need of retraining the whole model and using the knowledge obtained from other political parties' manifestos. Therefore, in other to evaluate if this approach for political parties' representation is easily scalable we have performed the following experiment. We have removed 5 political parties from the English training dataset and then, we have evaluated model's performance predicting this parties manifestos without any contextual data (D1), providing the political party using one hot encoding representation (D2) and its disentangled representation (D5). As it can be seen in Table 4, for Congress of the People,

TABLE 2. Domain (7 categories) classification results for each one of the experiment configuration and model (CNNs or BERT). The accuracy (acc), F-measure (F1) and G-mean (G-M) of each experiment is shown.

Experiment	English - CNNs	English - BERT
Baseline with no context	Acc: 64.29% F1: 60.04 G-M: 75.12	Acc: 65.9% F1: 61.47 G-M: 75.42
Political (one-hot-encoding)	Acc: 64.63% F1:60.17 G-M: 75.09	Acc: 66.8% F1:62.9 G-M: 76.17
Political party (disentangled repr.)	Acc: 64.72% F1: 60.37 G-M: 75.36	Acc: 66.36% F1:62.32 G-M: 75.92
Previous phrase + political party (one-hot-encoding)	Acc: 69.02% F1:65.32 G-M: 78.41	**Acc: 69.66%** **F1: 65.79** G-M: 78.2
Previous phrase + political party (disentangled repr.)	Acc: 69.04% F1: 65.46 **G-M: 78.87**	Acc: 69.5% F1: 65.68 G-M: 78.32

TABLE 3. Subdomain (56 categories) classification results for each one of the experiment configuration and model (CNNs or BERT). The accuracy (acc), F-measure (F1) and G-mean (G-M) of each experiment is shown.

Experiment	English - CNNs	English - BERT
Baseline with no context	Acc: 50.65% F1: 35.32 G-M: 58.51	Acc: 53.37% F1: 40.42 G-M: 62.43
Political (one-hot-encoding)	Acc: 52.18% F1: 38.89 G-M: 61.24	Acc: 55.46% F1: 44.06 G-M: 65.37
Political party (disentangled repr.)	Acc: 52.16% F1: 38.38 G-M: 60.81	Acc: 55.1% F1: 43.45 G-M: 64.5
Previous phrase + political party (one-hot-encoding)	Acc: 56.85% F1: 42.73 G-M: 64.56	**Acc: 58.64%** **F1: 47.1** **G-M: 67.8**
Previous phrase + political party (disentangled repr.)	Acc: 56.64% F1: 43.48 G-M: 65.2	Acc: 58.29% F1: 46.68 G-M: 67.36

TABLE 4. Comparison between one hot encoding representation and disentangled representation using political orientation for classifying manifestos of unknown parties for the trained model. D1 without any contextual data; D2 using one-hot encoding representation; D5 using the disentangled representation.

-	D1	D2	D5
Congress of the People (South Africa-181420)	Acc: 51.12% F1: 35.6	Acc: 55.24% F1: 32.27	Acc: **55.68%** F1: **37.33**
Labour Party (UK-51320)	Acc: **48.8%** F1: **29.52**	Acc: 48.14% F1: 29.25	Acc: 48% F1: 28.94
Anti-Austerity Alliance (Ireland-53240)	Acc: 42.99 % F1: 21.63	Acc: 45.37% F1: 23.22	Acc: **50.04%** F1: **27.75**
Australian Greens (Australia-63110)	Acc: 48.57% F1: 22.37	Acc: 48.32% F1: 22.8	Acc: **49.68%** F1: **23.65**
Scottish National Party (UK-51902)	Acc: **43.34%** F1: 25.66	Acc: 43.23% F1: 26.42	Acc: 43.07% F1: **26.92**

Anti-Austerity Alliance and Scottish National Party there is a considerable improvement from D1 to D5 in terms of accuracy and F-Measure. In particular, for Anti Austerity Alliance there is an improvement of 7 points in accuracy and 6 in F-Measure. In this case, the Geometric-Mean metric has not been reported because is computed among the 56 labels and there are some cases in these experiments where the manifestos corresponding to the party do not contain all the labels. However, the F-Measure ignores the label if there are no samples available.

B. EVALUATING WITH ANNOTATED POLITICAL TWEETS

In order to evaluate our approach we have used the two datasets previously mentioned: Manifestos Project's annotated 115 political manifestos and 5,000 annotated political tweets. Since our main goal is to analyse if annotated political manifestos with the contextual information previously introduced and tweets can work together as complementary training data for our political discourse classifier, we have divided our evaluation effort in three ways. The same test set of annotated political tweets is used for the three configurations so that results are comparable.

- *T1:* Trained exclusively with annotated political manifestos and evaluated with annotated political tweets.
- *T2:* Trained exclusively with annotated political tweets and evaluated with annotated political tweets.
- *T3:* Trained with annotated political manifestos and fine-tuned with annotated political tweets to later evaluate it with political tweets.

Therefore, the datasets has been split in the following way using stratification to maintain category distribution over all the sets:

- Annotated political manifestos: train set (70%), eval test (15%) and test set (15%).
- Annotated political tweets: train set (70%), eval test (15%) and the test set (15%) used in all the experiments.

Moreover, per each of the evaluations mentioned above, the following experiments has been conducted in order to analyse if the contextual data that we have previously proven that does work for manifestos classification, does also work on political tweets classification:

- Analyse if the previous tweet (a tweet has a preceding tweet if the tweet is answering or quoting another one) improves the performance of the classifier.
- Analyse if the political party to which the politician posting the tweet belongs to, improves the performance of the classifier. In this case, we have tested with the two representations which had the best performances: one hot encoding and disentangled representation using parties' political orientation.
- Analyse if the previous tweet and the party responsible of the tweet, using the two representation methods, are complementary features when are used together.

Unlike in the manifestos evaluation, in this case we have not used cross-validation in the evaluation. The reason behind this decision is the low number of annotated tweets compared to the number of samples for manifestos that would have resulted in a high variability between runs. In this experimentation, our goal has been to analyse the complementarity of manifestos datasets with respect to annotated tweets with the same codification. Therefore, in T1 the training data is the whole manifestos dataset and the test data is a subset of the annotated tweets which always will be the same during all the experiments. In T2, in order to be as comparable as possible with T1, the training data is all the annotated tweets excluding the test set. In T3, the model is first trained with the manifestos data used in T1 and then fine-tuned with the annotated tweets used in T2 for training.

We have split each dataset in 3 subsets because we have used early stopping [49] in order to stop's model training as soon as start over-fitting to the train set. In T1 the evaluation set used for early stopping is a subset of manifestos, in T2 a subset of annotated tweets and in T3, first a subset of manifestos and when the model is fine-tune, a subset of annotated tweets. Again, in all the experiments, the test set will always be the same set of tweets.

We have performed the experiments presented above with the two classification approaches used for manifestos: CNNs and BERT. Even though at first sight, after seeing the improvement achieved using BERT with respect to CNNs, it could be seen as something obvious that BERT would obtain better results than CNNs, the goal with these experiments (apart from analysing if contextual data helps), was to analyse if a language model such as BERT would perform better with tweets and without manifestos, than CNNs with tweets and manifestos. If so, this would demonstrate how powerful BERT's language model is and how good it generalises.

Also, even though we have tried to avoid the variability between executions not using cross-validation, we have found that the achieved results vary considerably among different executions. These differences were not that significant when classifying manifestos. Therefore, we have run each experiment 5 times and added to each metric its standard deviation in 5 runs.

C. DISCUSSION

After analysing the results shown in Table 5 and Table 6 the following conclusions can be drawn when it comes to classifying tweets in the 7 high level domains of the Manifestos Project categorisation scheme:

- As expected, BERT obtains better results than CNNs with Word2Vec embeddings. In terms of F-Measure, the highest F-Measure achieved with BERT is 64.55, exclusively trained with annotated tweets and both contextual information (party with one hot encoding and previous tweet) as extra features. In addition, CNNs obtain their best F-Measure result 57.65, trained with manifestos, fine-tuned with annotated tweets and with the previous tweet and political party using disentangled representation as extra feature. Therefore, we can affirm that at least for the classification of the tweets in 7 domains, BERT's language model fine-tuned with annotated tweets is more powerful than CNNs trained with manifestos and fine-tuned with annotated tweets.
- Even though there are much less training samples, BERT achieves better results when the model is trained exclusively with annotated tweets than with political manifestos. This may happen because first, the language used in Twitter differs from the language used in political manifestos; second, because the language used in Twitter could be can be simpler than the one used in political manifestos; and third, BERT's pre-trained language model would be enough for domain classification.
- In case of the CNNs, fine-tuning the model exclusively trained with political manifestos with annotated tweets drastically improves models performance in both accuracy and F-Measure, achieving an improvement of almost 9 points in both measures with respect to T1. On the contrary, BERT does not achieve the best results fine-tuning it with manifestos and tweets. As it has been mentioned before, the best results are achieved ignoring annotated manifestos and fine-tuning the model using annotated tweets.
- With regard to the use of contextual data, we cannot affirm that the previous tweet (quoted or answering to) does contribute to an improvement when CNNs are used. However, it is true that with BERT, in those experiments where annotated tweets are part of the fine-tuning process (T2 and T3), the best results are achieved when the previous tweet is part of the used contextual data. Moreover, it is also worth mentioning that the highest standard deviation values are seen when the previous tweet is used as contextual data.
- As for the use of the political party to which the politician who has written the tweet belongs, a clear improvement can be perceived every time this contextual data is used, both with CNNs and BERT. In fact, CNNs obtain their best results when the political party is used. In this case, one hot encoding is in most of the cases the method of representation that best works.
- With regard to the complementarity of the proposed contextual data, we can affirm that they are complementary in the experiments T2 and T3 of BERT and CNNs where the best results of this approach are achieved.

With regard to the results shown in Table 7 and Table 8 the following conclusions can be drawn when it comes to classifying tweets in the 56 subdomains of the Manifestos Project categorisation scheme:

- On the contrary of what happens with Domains, in this case the best results are achieved with BERT fine-tuned with political manifestos and annotated tweets, obtaining a F-Measure of 50.07 and a G-Mean of 62.4. This values are obtained using the political party as an extra feature.
- As it happens with the high level domains, fine-tuning the model with annotated political tweets drastically improves models performance when classifying the tweets in the 56 categories. Achieving improvements in the F-Measure of more than 15 points using BERT and 10 using CNNs.
- Again, it is noteworthy mentioning the improvement gained training the model exclusively with annotated tweets (T2) compared to training it with annotated manifestos (T1). As it happens with the high level domains, this may happen due to the different language used by politicians in manifestos and Twitter. Also, this difference is significantly bigger when classifying subdomains. However, in this case the best results are achieved when annotated manifestos and tweets are combined (T3).
- Using the previous tweet as contextual data improves the performance in CNN-T2, CNN-T3, BERT-T2 and BERT-T3. This is similar to what happens when classifying high level domains, where every-time annotated tweets are used in the fine-tuning process, the previous tweet improves model's performance. This could mean that the model is not able to adapt the meaning that the previous statement has in manifestos classification task to the meaning that the previous tweet could have when classifying Tweets. Therefore, models classifying tweets are not able to take advantage of the previous tweet/sentence until they are trained with annotated tweets, where the model is able to adapt to the new classification problem.
- The political party to which the politician who has written the tweet belongs, obtains the best results in both approaches: CNNs-T3 and BERT-T3. Regarding the method of representation, both disentangled and one-hot representation achieve similar results, being the latter the best performing in most of the cases.

TABLE 5. Domain results with CNNs (the average results of 5 runs per experiment are shown) for each one of the experiment configuration. The accuracy (acc), F-measure (macro), G-mean and their respective standard deviation is shown.

Experiment	Accuracy	F-Measure	G-Mean
T1	51.57% ±0.4	47.40 ±0.5	66.93 ±0.6
T1 + Prev.Tweet	49.93% ±0.7	45.13 ±0.4	64.24 ±0.3
T1 + P.Party (One hot)	**52.16%** ±0.9	**48.03** ±0.5	**67.21** ±0.5
T1 + P.Party (Disentangled)	51.4% ±1.5	46.97 ±1.4	66.39 ±1.14
T1 + P.Party (One hot) + Prev.Tweet	50.12 ±1.3	46.24 ±1.4	65.12 ±1.1
T1 + P.Party (Disentangled) +Prev.Tweet	50.63 ±0.8	46.68 ±0.8	65.7 ±0.7
T2	57.83% ±0.4	54.62 ±1	71.21 ±0.7
T2 + Prev.Tweet	58.09% ±1.9	54.16 ±2.8	71.05 ±1.9
T2 + P.Party (One hot)	58.31% ±0.9	54.8 ±0.8	71.15 ±0.6
T2 + P.Party (Disentangled)	58.33% ±1.7	54.6 ±1.9	71.06 ±1.2
T2 + P.Party (One hot) + Prev.Tweet	**58.68%** ±0.8	**55.5** ±1.1	**71.74** ±0.7
T2 + P.Party (Disentangled) + Prev.Tweet	58.04 ±1.15	55.2 ±0.9	71.67 ±0.8
T3	59.52% ±1.8	56.2 ±2.4	71.99 ±1.6
T3 + Prev.Tweet	58.58% ±1.7	55.42 ±2.1	71.5 ±1.5
T3 + P.Party (One hot)	60.13% ±0.7	57.14 ±0.8	72.6 ±0.4
T3 + P.Party (Disentangled)	59.51% ±0.8	56.54 ±0.6	72.42 ±0.5
T3 + P.Party (One hot) + Prev.Tweet	59.73 ±1.24	57.07 ±1.36	72.96 ±1
T3 + P.Party (Disentangled) + Prev.Tweet	**60.83%** ±1.1	**57.65** ±0.9	**73.28** ±0.6

TABLE 6. Domain results with BERT (the average results of 5 runs per experiment are shown) for each one of the experiment configuration. The accuracy (acc), F-measure (macro), G-mean and their respective standard deviation is shown.

Experiment	Accuracy	F-Measure	G-Mean
T1	55.83% ±0.8	51.39±1.25	68.65 ±1.2
T1 + Prev.Tweet	54.73% ±2.44	51.24 ±1.6	68.62 ±1
T1 + P.Party (One hot)	**57.5**% ±1.76	**53.5** ±1.88	**70.69** ±1.25
T1 + P.Party (Disentangled)	56.21% ±2.5	52.44 ±2.7	69.65 ±1.9
T1 + P.Party (One hot) + Prev.Tweet	56.88% ±1.7	52.74±1.7	69.82 ±1.3
T1 + P.Party (Disentangled) + Prev.Tweet	57.15% ±0.8	52.43 ±1	69.94 ±0.9
T2	66.79% ±1.9	63.48±2	76.7 ±1.5
T2 + Prev.Tweet	67.73% ±1-85	63.73±2.17	77.09 ±1.5
T2 + P.Party (One hot)	67.08% ±0.75	64.19 ±0.4	**77.49** ±0.7
T2 + P.Party (Disentangled)	67.22% ±1.2	63.84 ±1.1	77.23 ±0.8
T2 + P.Party (One hot) + Prev.Tweet	**67.91**% ±1.7	**64.55**±1.97	77.4 ±1.4
T2 + P.Party (Disentangled) + Prev.Tweet	67.19% ±0.7	63.84 ±0.85	77.26 ±0.8
T3	65.36% ±1.18	61.88 ±1.82	75.78 ±1.2
T3 + Prev.Tweet	66.79% ±0.9	63.44 ±1.39	76.99 ±1.14
T3 + P.Party (One hot)	65.77% ±0.7	62.23 ±2.5	76.11 ±1.7
T3 + P.Party (Disentangled)	67.16% ±0.9	63.6 ±1.4	77.02 ±0.9
T3 + P.Party (One hot) + Prev.Tweet	**67%** ±0.7	**63.57** ±1.05	**77.04** ±0.75
T3 + P.Party (Disentangled) + Prev.Tweet	66.95% ±0.7	62.75 ±1.7	76.39 ±1.1

- In this case, the proposed contextual data are not complementary since the best results are obtained using exclusively the political party.

TABLE 7. Subdomain results with CNNs (the average results of 5 runs per experiment are shown) for each one of the experiment configuration. The accuracy (acc), F-measure (macro), G-mean and their respective standard deviation is shown.

Experiment	Accuracy	F-Measure	G-Mean
T1	37.20% ±2.1	22.83 ±1	47.68 ±1.29
T1 + Prev.Tweet	34.74% ±2.3	21.79 ±2.1	45.68 ±2
T1 + P.Party (One hot)	35.96% ±1.2	23.28 ±1.8	47.66 ±1.26
T1 + P.Party (Disentan.)	**37.57%** ±2.6	**24.14** ±2.3	**48.4** ±1.7
T1 + P.Party (One hot) + Prev.Tweet	37.14 ±1.9	23.64 ±1.4	47.83±1.46
T1 + P.Party (Disentan.) + Prev.Tweet	37.57% ±2.6	22.54 ±2.1	46.46 ±2.3
T2	42.88% ±1.88	27.58 ±1.94	45.8 ±1.33
T2 + Prev.Tweet	43.93% ±1.4	28.67 ±2.22	47.4 ±1.88
T2 + P.Party (One hot)	44.14% ±1.75	28.27 ±1.6	46.67 ±1.18
T2 + P.Party (Disentan.)	**44.66%** ±1.21	**30** ±1.75	**47.92** ±1.22
T2 + P.Party (One hot) + Prev.Tweet	42.58% ±2.2	28.21 ±2.1	47.68 ±1.3
T2 + P.Party (Disentan.) + Prev.Tweet	44.2% ±0.5	28.32 ±1.3	46.82 ±1.07
T3	49.46% ±1.4	38.06 ±2.83	54.75 ±2.1
T3 + Prev.Tweet	50.32% ±0.7	38.58 ±2.87	54.8 ±2.48
T3 + P.Party (One hot)	49.76% ±1.6	39.20 ±1.96	55.23 ±1.58
T3 + P.Party (Disentan.)	**51.99%** ±1.05	**41.43** ±1.35	**57.03** ±1
T3 + P.Party (One hot) + Prev.Tweet	50% ±2.5	37.75 ±3.4	54.64 ±2.6
T3 + P.Party (Disentan.) + Prev.Tweet	50.18% ±1.89	39.15 ±3.96	55.82 ±3.23

TABLE 8. Subdomain results with BERT (the average results of 5 runs per experiment are shown) for each one of the experiment configuration. The accuracy (acc), F-measure (macro), G-mean and their respective standard deviation is shown.

Experiment	Accuracy	F-Measure	G-Mean
T1	44.03 % ±1.8	28.29±2	50.53 ±1.5
T1 + Prev.Tweet	43.61% ±1.79	27.81 ±0.7	50.48 ±0.7
T1 + P.Party (One hot)	45.25% ±1.47	27.8 ±0.9	51.49 ±1.1
T1 + P.Party (Disentan.)	45.71% ±1.83	**29.08** ±2.09	**52.13** ±1.82
T1 + P.Party (One hot) + Prev.Tweet	**45.95%** ±1.84	28.19±1	51.55 ±1.2
T1 + P.Party (Disentan.) + Prev.Tweet	45.57% ±1.95	28.71 ±2.26	50.87 ±2.18
T2	59.16% ±1.1	45.86 ±2	59.98 ±1.06
T2 + Prev.Tweet	59.73% ±1.7	47.01 ±2.5	60.51 ±1.9
T2 + P.Party (One hot)	58.51% ±1.89	44.06 ±2.3	58.76 ±1.4
T2 + P.Party (Disentan.)	**60.24%** ±0.6	46.66 ±1.6	60.21 ±0.5
T2 + P.Party (One hot) + Prev.Tweet	59.16% ±1.3	47.51 ±2.47	**60.89** ±1.78
T2 + P.Party (Disentan.) + Prev.Tweet	60.21% ±1.07	**47.74** ±2.6	60.57 ±1.48
T3	60.67% ±1.3	48.02±2.56	60.57 ±1.75
T3 + Prev.Tweet	60.48% ±1.8	48.19 ±2.38	60.64 ±1.64
T3 + P.Party (One hot)	60.51% ±1.08	**50.07**±3.1	62.4±2.58
T3 + P.Party (Disentan.)	**61.64%** ±1.05	49.81 ±2.66	61.83 ±1.44
T3 + P.Party (One hot) + Prev.Tweet	60.29% ±1.2	48.68 ±2.5	60.96±1.8
T3 + P.Party (Disentan.) + Prev.Tweet	60.08% ±0.6	49.94 ±1.09	**62.44** ±1.13

Finally, we have analysed how feasible would be to change from the multiclass classification problem that we are been dealing with during this work, to a multilabel classification problem where those secondary ideas some tweets could contain are also taken into account. As it has been already explained in III-B, we annotated some secondary categories (apart from the principal one), in those tweets with more than one concept. In this case, we have only used BERT as

TABLE 9. Multilabel subdomain results with BERT with a strict evaluation(the average results of 5 runs per experiment are shown) for each one of the experiment configuration. The accuracy (acc), F-measure (macro) and their respective standard deviation is shown.

Experiment	Accuracy	F-Measure
ML1	29.95% ±0.5	25.02 ±1.81
ML1 + Previous Tweet	30.71% ±1.32	24.5 ±1.6
ML1 + Political party (One hot)	30.85% ±0.6	24.4 ±1.5
ML1 + Political party (Disentan.)	30.17% ±1.03	25 ±1.48
ML1 + Political party (One hot) + Previous Tweet	**31.32**% ±1.27	25.86 ±1.2
ML1 + Political party (Disentan.) + Previous Tweet	31.18% ±1.22	**25.97** ±2.47
ML2	**39.6**% ±1.7	41.42 ±1.93
ML2 + Previous Tweet	38.74% ±2.22	40.7 ±4
ML2 + Political party (One hot)	39.35% ±	40.5 ±3.8
ML2 + Political party (Disentan.)	39.1% ±1.5	41.23 ±3.3
ML2 + Political party (One hot) + Previous Tweet	39.16% ±1.58	41.43±2.1
ML2 + Political party (Disentan.) + Previous Tweet	39.16% ±0.5	**42.34** ±1.4
ML3	37.35% ±2.58	39.25 ±3
ML3 + Previous Tweet	37.87% ±1.14	39.71 ±2
ML3 + Political party (One hot)	**39.27**% ±0.9	39.12 ±3.13
ML3 + Political party (Disentan.)	38.91% ±1.09	41.27 ±1.14
ML3 + Political party (One hot) + Previous Tweet	39.23% ±1.87	40.13 ±1.4
ML3 + Political party (Disentan.) + Previous Tweet	38.88% ±2.74	**42.71** ±2.14

TABLE 10. Multilabel subdomain results with BERT with a less strict evaluation(the average results of 5 runs per experiment are shown) for each one of the experiment configuration. The accuracy (acc), F-measure (macro) and their respective standard deviation is shown.

Experiment	Accuracy	F-Measure
ML1	42.1% ±1.1	25.41 ±1.64
ML1 + Previous Tweet	41.04% ±0.8	25.4 ±0.7
ML1 + Political party (One hot)	40.3% ±1.37	24.51 ±1.75
ML1 + Political party (Disentangled)	40.72% ±0.9	26.3 ±1.4
ML1 + Political party (One hot) + Previous Tweet	**43.78%** ±1.83	26.52 ±1.17
ML1 + Political party (Disentangled) + Previous Tweet	43.15% ±1.89	**27.58** ±2.56
ML2	55.37% ±0.9	43.72 ±1.33
ML2 + Previous Tweet	**56.2%**±1.4	**46.52** ±4.3
ML2 + Political party (One hot)	55.53% ±0.6	43.67 ±1.86
ML2 + Political party (Disentangled)	55.06% ±0.5	45.54±3.6
ML2 + Political party (One hot) + Previous Tweet	55.46% ±0.9	45.83 ±3.06
ML2 + Political party (Disentangled) + Previous Tweet	55.49% ±1.08	46.1 ±1.43
ML3	54.22% ±1.38	42.87 ±5.87
ML3 + Previous Tweet	**54.78%** ±1.07	43 ±2.1
ML3 + Political party (One hot)	55.15% ±1.02	45.25 ±2.8
ML3 + Political party (Disentangled)	55.37% ±1.42	46.32±1.38
ML3 + Political party (One hot) + Previous Tweet	55.04% ±2.58	44.52 ±3.18
ML3 + Political party (Disentangled) + Previous Tweet	54.75% ±2.1	**47.5** ±2.3

classification model and subdomains as objectives. We have used BERT because is the model that has given the best results and we have decided not to use high level domains because in most of the cases ideas inside a Tweet would belong to the same high level domain.

First, we have evaluated this task being as strict as possible, considering a corrected predicted sample a Tweet where all the labels were correctly predicted. These results are reported in Table 9. In this case, we have not used the G-Mean as an evaluation metric because it was not designed for multi-label evaluation. As expected since a multi label problem in this context is more complex than a multiclass problem, the results are worse than those achieved previously in the multiclass classification problem for subdomains. The best results in terms of F-Measure (Macro) has been obtained in ML2 and ML3 using all contextual information as extra features, 42.34 and 41.71 respectively. However, ML2 without any contextual information achieves the best accuracy rate (+0.44%) but it has worse F-Measure, −0.92. Regarding the complementarity of annotated manifestos and tweets in this task, even though the best results in terms of F-Measure is achieved in ML3 (manifestos + tweets), the difference with respect to ML2 (only tweets) is minimal: +0.37. Therefore, we cannot conclude that in this case both datasets are complementary.

Second, we have evaluated the task being less strict, considering a corrected predicted sample a Tweet where at least one of the labels were correctly predicted. These results are reported in Table 10. Predictably, the results have improved with respect to strict evaluation shown in Table 9. ML2 achieves its best results using the preceding tweet, 3 points better in F-Measure compared with the baseline. Also, disentangled representation for political parties improves baseline's performance and outperforms by a wide margin the one-hot encoding representation. However, in this case previous tweet and political party are not complementary data. With regard to ML3, it achieves the best results, confirming the fact that annotated manifestos and tweets are complementary. In this case, previous tweet and political party are complementary data.

V. USE CASE SCENARIO: ANALYSIS OF 2016 UNITED STATES PRESIDENTIAL ELECTIONS

In order to demonstrate how useful the proposed approach is, we introduce a possible use case scenario for the designed political discourse classifier: to analyse the tweets of the presidential (Hillary Clinton and Donald Trump) and vice-presidential (Tim Kaine and Mike Pence) candidates for the 2016 United States presidential elections.

We used a dataset of the 2016 United States Presidential Election Tweet IDs [50] with tweets gathered between July 13, 2016 and November 10, 2016. However, we only used a small part of the dataset: presidential and vice-presidentials candidates' timelines (ignoring RTs) during the previously mentioned time period: 5346 tweets from Hillary Clinton, 3364 from Tim Kaine, 4510 from Donald Trump and 1744 from Mike Pence. We processed candidates' tweets with the same procedure used for the annotated tweets:

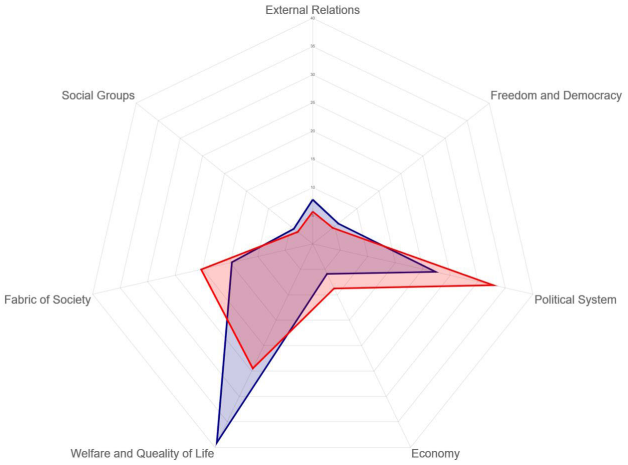


FIGURE 5. Distribution of tweets among 7 high level domains of the tweets created by democratic (blue) and republican (red) candidates.

tokenization, removing stopwords and URLs and maintaining hashtags.

First of all, we performed a preliminary analysis classifying candidates' tweets in the previously mentioned 7 high level policy domains in order to have a general overview of each political parties (democratic and republican) preferences. To do so, we used BERT trained with political manifestos and fine-tuned with annotated political tweets.

Furthermore, the political affiliation of the transmitter and the previous tweet was used as contextual data (the best results were achieved using the political leaning as an extra feature, see Table 6).

In Figure 5 the distribution of tweets of the Republican (red) and Democratic (blue) parties over the 7 high level policy domains can be seen.

The first worth mentioning aspect is how *Political System* is the dominant category for Republicans, whereas *Welfare and Quality of Life* is for Democrats. However, the Democratic party also emphasises in the Political system being their second priority. One of the reasons behind this could be that inside the high level *Political System* domain, there is a category named *Political Authority* which encompasses messages related with politician's competence to govern or the political opponent's lack of such competence. Therefore, tweets complimenting his or her allies and criticising his or her opponents would belong to *Political System* domain. Concerning the *Fabric of Society* domain, Republicans emphasise more than democrats in this high level policy. In the rest of high level policy domains, both parties have similar distributions.

However, this kind of political discourse analysis based on 7 high policy domains does not offer an accurate view of what is really happening, it only offers a general overview. This is the reason why we are proposing a more precise approach for political discourse analysis in social media using the 56 categories defined in Table 1.

Therefore, we have applied this new approach for analysing 2016 presidential elections. Nonetheless, in this use case we are going to emphasise on those categories that are not marginal. Marginal categories are those with less than 0.5% of the total amount of tweets in both political parties.

In Figure 6 what highlights the most is the fact that more than 25% of the tweets from both parties have been classified as *Political Authority* (305), which means that most of the political discourse during this elections was focused on attacking the opponent or praising themselves. However, the discourse from the republicans was more *Political Authority* centred compared to Democratic Party. This results coincide with the results obtained manually by [16].

Another point of interest would the disparity between republicans and democrats regarding *Equality* (503) category, which includes policies related with social justice, fair treatment of all people and the end of discrimination according to the Manifestos' Project handbook.[4] It is also remarkable that this disparity can also be detected with *Welfare State Expansion* (504) category in a similar way. Moreover, republicans talk more about *Welfare State Limitation* (505) than democrats do.

In respect of their policy preferences regarding Economy, Republicans focused on tweets about *Free Market Economy (401)*, whereas Democrats where more concerned about *Market Regulation (403)*. Nonetheless, both parties have similar percentages of tweets related with *Incentives (402)* for businesses.

Regarding nationalism and immigration, it can be seen clearly in those categories related with immigration that Republicans sent more anti-immigration tweets than Democrats.

For instance, 5.26% of the Republican tweets have been classified in *the National Way Of Life - Positive* (601) category, where statements about patriotism and against the process of immigration are included, unlike Democrats whose 1.96% of the tweets are related to this matter. Moreover, Republicans have a marginal representation in those categories that promote immigrants' rights: *National Way of Live - Negative* (602) and *Multiculturalism* (607).

To conclude, even though there are small differences, Republicans were more concerned about *Military: Positive* (104) and *Law and Order* (605) categories than Democrats. However, the opposite happens with *Environmental protection (501) and Labour Groups - Positive* where democrats shown more interest (701).

VI. CONCLUSION AND FUTURE WORK

In this research, we have introduced a novel approach for automatically classifying political tweets using a categorisation scheme widely used by political scientist. To do so, we have been able to prove how annotated political manifestos and annotated political tweets are complementary information when it comes to training our political discourse classifier.

[4] https://manifesto-project.wzb.eu/down/papers/handbook_2014_version_5.pdf

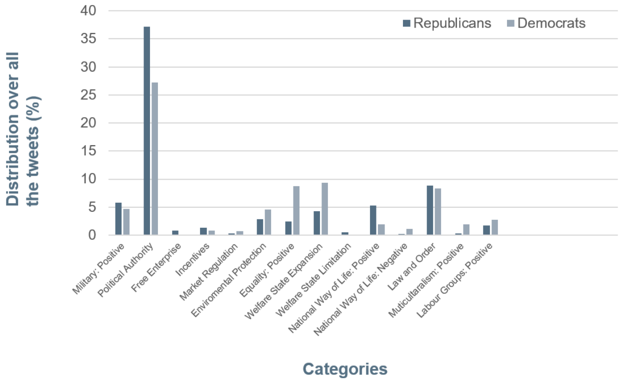


FIGURE 6. Distribution among 56 subdomains of the tweets created by democratic and republican candidates.

Moreover, we have also proven how the political leaning of the tweet's author is a useful contextual feature as well as the previous tweet. Finally, it is also noteworthy mentioning how the language used in manifestos and social media differ, as we have proven in Section IV. Moreover, we have introduced a use case scenario explaining how our approach could be used to analyse the political discourse in social media using an approach widely used by political scientists. Finally, the dataset of 5,000 tweets annotated with the CPM coding schema.[5] has been published as well as the source code.[6]

Inspired by the limitations of the research presented in this manuscript, we have identified the following further research lines. First, we have analysed the results of a multi-label classification problem, assuming that a tweet could contain more than one idea. However, the results obtained with this assumption are considerably worse than the metrics obtained considering that each tweet represent a principal political idea. Therefore, we believe that in order to improve the results in the multi-label classification task, first, a bigger dataset of annotated tweets should be needed, and second, a multi-label specific neural network architectures should be tested. Second, it would be interesting to analyse the subdomain *305 Political Authority* in social media from a positive or negative point of view. This category encompasses those statements with a partisan rhetoric where politicians praise their policies or actions, whereas criticise their rivals. Unfortunately, this category does not differentiate the first from the latter as [16] did. Moreover, it would interesting to apply Named Entity Recognition (NER) techniques in this category in order to analyse who are they talking about and how. Thus, we consider that this addition would enrich the political discourse analysis in social media.

REFERENCES

[1] A. Casero-Ripollés, "Research on political information and social media: Key points and challenges for the future," *El Profesional de la Información*, vol. 27, no. 5, p. 964, Sep. 2018.

[2] D. V. Dimitrova and J. Matthes, "Social media in political campaigning around the world: Theoretical and methodological challenges," *Journalism Mass Commun. Quart.*, vol. 95, no. 2, pp. 333–342, Jun. 2018, doi: 10.1177/1077699018770437.

[3] D. Sayce. (2019). *The Number of Tweets Per Day in 2019*. [Online]. Available: http://www.dsayce.com/social-media/tweets-day

[4] A. Volkens, W. Krause, P. Lehmann, T. Matthieß, N. Merz, S. Regel, and B. WeßYels. (2018). *The Manifesto Data Collection. Manifesto Project (MRG/CMP/MARPOR)*. Version 2018b. Berlin, Germany. [Online]. Available: https://doi.org/10.25522/manifesto.mpds.2018b

[5] A. Werner, O. Lacewell, and A. Volkens. (2011). *Manifesto Coding Instructions (4th Fully Revised Edition), May 2011*. [Online]. Available: http://goo.gl/g512Q

[6] S. Alonso, L. Cabeza, and B. Gómez, "Disentangling peripheral parties' issue packages in subnational elections," *Comparative Eur. Politics*, vol. 15, no. 2, pp. 240–263, Mar. 2017.

[7] K. Benoit, "Irish political parties and policy stances on European integration," *Irish Political Stud.*, vol. 24, no. 4, pp. 447–466, Dec. 2009.

[8] A. Bilbao-Jayo and A. Almeida, "Automatic political discourse analysis with multi-scale convolutional neural networks and contextual data," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 11, 2018, Art. no. 1550147718811827.

[9] J. Mellon and C. Prosser, "Twitter and Facebook are not representative of the general population: Political attitudes and demographics of British social media users," *Res. Politics*, vol. 4, no. 3, 2017, Art. no. 2053168017720008.

[10] J. Stromer-Galley, *Presidential Campaigning in the Internet Age*. London, U.K.: Oxford Univ. Press, 2019.

[11] M. Ramos-Serrano, J. D. Fernández Gómez, and A. Pineda, "Follow the closing of the campaign on streaming': The use of Twitter by Spanish political parties during the 2014 European elections," *New Media Soc.*, vol. 20, no. 1, pp. 122–140, Jan. 2018.

[12] G. López-García, "'New' vs 'old' leaderships: The campaign of Spanish general elections 2015 on Twitter," *Commun. Soc.*, vol. 29, no. 3, pp. 149–168, 2016.

[13] A. Casero-Ripollés, M. Sintes-Olivella, and P. Franch, "The populist political communication style in action: Podemos's issues and functions on Twitter during the 2016 Spanish general election," *Amer. Behav. Scientist*, vol. 61, no. 9, pp. 986–1001, Aug. 2017.

[14] A. Lopez-Meri, S. Marcos-Garcia, and A. Casero-Ripolles, "What do politicians do on Twitter? Functions and communication strategies in the Spanish electoral campaign of 2016," *El Profesional Información*, vol. 26, no. 5, pp. 795–804, Sep. 2017.

[15] L. Alonso-Muñoz and A. Casero-Ripollés, "Political agenda on Twitter during the 2016 Spanish elections: Issues, strategies, and users' responses," *Commun. Soc.*, vol. 31, no. 3, pp. 7–23, 2018.

[16] A. Russell, "U.S. senators on Twitter: Asymmetric party rhetoric in 140 characters," *Amer. Politics Res.*, vol. 46, no. 4, pp. 695–723, Jul. 2018.

[17] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, "Political ideology detection using recursive neural networks," in *Proc. Assoc. Comput. Linguistics*, 2014, pp. 1113–1122.

[18] A. Rao and N. Spasojevic, "Actionable and political text classification using word embeddings and LSTM," 2016, *arXiv:1607.02501*. [Online]. Available: http://arxiv.org/abs/1607.02501

[19] J. Ratkiewicz, M. Conover, M. R. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media," in *Proc. ICWSM*, vol. 11, 2011, pp. 297–304.

[20] A. Zubiaga, M. Liakata, R. Procter, G. W. S. Hoi, and P. Tolmie, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," *PLoS ONE*, vol. 11, no. 3, Mar. 2016, Art. no. e0150989.

[21] S. Stier, A. Bleier, H. Lietz, and M. Strohmaier, "Election campaigning on social media: Politicians, audiences, and the mediation of political communication on Facebook and Twitter," *Political Commun.*, vol. 35, no. 1, pp. 50–74, Jan. 2018.

[22] R. Schmitt-Beck, E. Bytzek, H. Rattinger, S. Roßteutscher, and B. Weßels, "The German longitudinal election study (gles)," *Vortrag im Rahmen der Jahrestagung der Int. Commun. Assoc. (ICA), Chicago*, vol. 21, p. 25, May 2009.

[23] U. Yaqub, S. A. Chun, V. Atluri, and J. Vaidya, "Analysis of political discourse on Twitter in the context of the 2016 US presidential elections," *Government Inf. Quart.*, vol. 34, no. 4, pp. 613–626, Dec. 2017.

[5] https://github.com/AritzBi/tweets_manifestos_methodology_2016_usa

[6] https://github.com/AritzBi/manifestos_context_classifier

[24] F. Nanni, C. Zirn, G. Glavaš, J. Eichorst, and S. P. Ponzetto, "TopFish: Topic-based analysis of political position in US electoral campaigns," in *Proc. Int. Conf. Adv. Comput. Anal. Political Text (PolText), Eur. Social Fund, Oper. Programme Efficient Hum. Resour.*, D. Širinić, Ed. Zagreb, Croatia: Univ. Zagreb, 2016. [Online]. Available: https://madoc.bib.uni-mannheim.de/41550/

[25] A. B. Jayo and A. Almeida, "Political discourse classification in social networks using context sensitive convolutional neural networks," in *Proc. 6th Int. Workshop Natural Lang. Process. Social Media*, Melbourne, VIC, Australia, Jul. 2018, pp. 76–85. [Online]. Available: http://www.aclweb.org/anthology/W18-3513

[26] F. Nanni *et al.*, "Findings from the hackathon on understanding euroscepticism through the lens of textual data," in *Proc. LREC Workshop ParlaCLARIN, Creating Using Parliamentary Corpora*, D. Fišer, Ed. Miyazaki, Japan, May 2018, pp. 59–66. [Online]. Available: https://madoc.bib.uni-mannheim.de/44172/

[27] I. Budge, *Mapping Policy Preferences: Estimates for Parties, Electors, and Governments*, vol. 1. London, U.K.: Oxford Univ. Press, 2001, pp. 1945–1998.

[28] A. Volkens. (2002). *Manifesto Coding Instructions*. [Online]. Available: https://www.poltext.org/sites/poltext.org/files/iii02-201.pdf

[29] A. Volkens, W. Krause, P. Lehmann, T. Matthieß, N. Merz, S. Regel, and B. WeßYels. (2019). *The Manifesto Data Collection. Manifesto Project (MRG/CMP/MARPOR)*. Version 2019a. Berlin, Germany. [Online]. Available: https://doi.org/10.25522/manifesto.mpds.2019a

[30] W. Nordsieck, *Parties and Elections in Europe*, 3rd ed. Norderstedt, Germany: Books on Demand, Jun. 2019.

[31] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[32] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882

[33] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2539–2544.

[34] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[36] Y. LeCun. (2015). *LeNet-5, Convolutional Neural Networks*. [Online]. Available: http://yann.lecun.com/exdb/lenet

[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[38] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*. [Online]. Available: http://arxiv.org/abs/1510.03820

[39] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[40] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 111–118.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[44] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding With Unsupervised Learning*. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/ language_understanding_paper.pdf

[45] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[47] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.

[48] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*. [Online]. Available: http://arxiv.org/abs/1609.08144

[49] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks Trade*. Berlin, Germany: Springer, 1998, pp. 55–69.

[50] J. Littman, L. Wrubel, and D. Kerchner. (2016). *2016 United States Presidential Election Tweet IDS*. [Online]. Available: https://doi.org/10.7910/DVN/PDI7IN

ARITZ BILBAO-JAYO received the bachelor's, master's, and Ph.D. degrees in computer science from the University of Deusto, in 2014, 2016, and 2020, respectively. He is currently a Research Associate with the Faculty of Engineering, DeustoTech Institute, University of Deusto. He has worked on European Research Projects, such as SONOPA(AAL) or MoveSmart (FP7). His research interests include the application of deep learning and natural language processing techniques on political discourse analysis on social networks, and user behaviour analysis within the BD4QoL H2020 project.

AITOR ALMEIDA received the Ph.D. degree in computer science from the University of Deusto. He is currently a Researcher and the Project Manager of the Faculty of Engineering, DeustoTech Institute, University of Deusto. His research interests include the analysis of the behaviour of the users in intelligent environments, the application of artificial intelligence for smart health, and the study of the users' activity and discourse on social networks.