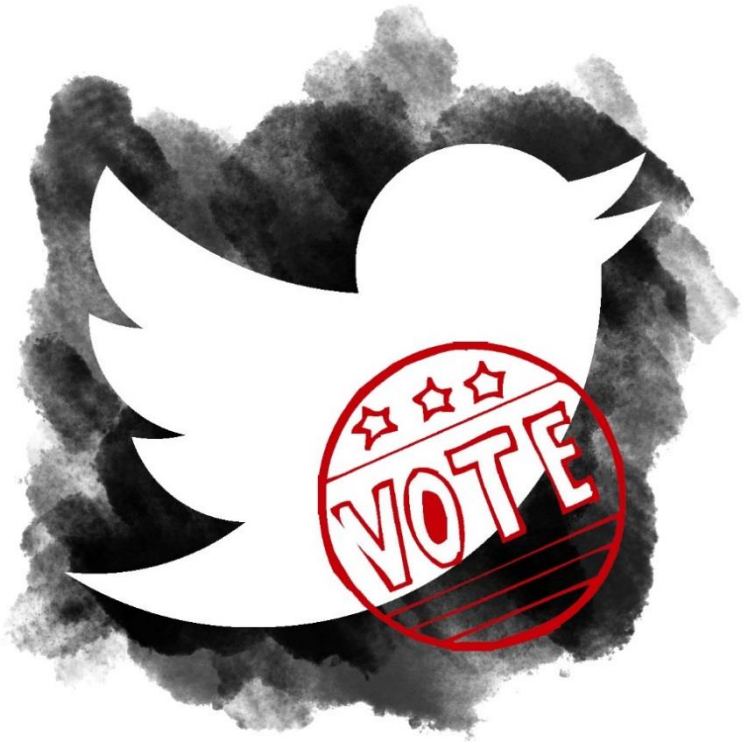
DIPLOMA THESIS

Election Analysis and Prediction of Election Results with Twitter

Virginia Tsintzou

Advisor: Associate Professor Panayiotis Tsaparas



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

October 2016

Acknowledgements

First I would like to thank Professors Evaggelia Pitoura, Nikolaos Mamoulis for being members of my thesis committee and for their invaluable comments on my work and its perspectives.

Special thanks to my advisor and also member of the committee Professor Panayiotis Tsaparas for his useful guidance and support. Most importantly, I have to acknowledge and thank him for his open mind to any ideas from his students and patience.

Last but not least, I have to thank my parents Emilios and Nausika, my sister Iro, my friend Yiannos and all of my friends for their continuous support.

Table Of Contents

1 Introduction

1.1 Motivation

Elections are the means to people's choice of representation. Due to their important role in politics, there always has been a big interest in predicting an election outcome. Lately, it is observed that traditional polls may fail to make an accurate prediction. The scientific community has turned its interest in analyzing web data, such as blog posts or social networks' users' activity as an alternative way to predict election outcomes, hopefully more accurate. Furthermore, traditional polls are too costly, while online information is easy to obtain and freely available. This is an interesting research area that combines politics and social media which both concern today's society. It is interesting to employ technology to solve modern-day challenges.

One of the most popular online social networking services nowadays is Twitter. It enables users to send and read short 140-character messages called "tweets". Twitter's network of users is created by following other users, in order to view their tweets in real time or to be notified when someone posts a new tweet. Following other users indicates interest in what they have to say, so the action of following can be interpreted as a vote of confidence. As the number of followers is a measure of popularity, popular users appear as recommendations to other users, expanding the network more quickly.

Twitter captures society's "pulse". User's tweets are indicators of what people talk about or how they feel. It's important that we can use tweets to understand the public's opinion on different news and current topics. Topics about politics and elections often become subjects of discussion in Twitter. Most recently, the Irish general election of 2016 and the United Kingdom European Union membership referendum appeared on Twitter's posts.

This thesis, will analyze data from Twitter regarding the Irish elections and the UK referendum and examine whether an election outcome could be predicted from this data.

1.2 Overview

We will consider two political events in this thesis: the Irish General Election and the United Kingdom European Union membership referendum.

The Irish General Election took place on February 26, 2016. A total of 551 candidates contested the election. In total 18 parties and alliances took part, while some politicians exercised their candidacy as independent.

The United Kingdom European Union membership referendum, known within the United Kingdom as the EU referendum, or the Brexit referendum, was a referendum that took place on June 23, 2016 in the United Kingdom and Gibraltar to gauge support for the country's membership in the European Union. People were asked to vote “remain” or “leave” as to whether UK should remain a member of the European Union or not.

The goal of this thesis is to use Twitter data to predict the outcome of these two elections. To this end, we will use both text from tweets and the network of followers to examine if they indicate similar behavior. Since the related literature mainly concentrates on processing users’ content rather than their connections, it would be interesting to explore if useful conclusions on users’ political opinion can be produced by analyzing the network.

For the Irish elections, the network we will consider is created by the parties and their followers. We will use Link Analysis Techniques (PageRank, HITS and Random Walks) to estimate users’ intention to vote for the parties, or the support of the Twitter users to the different parties.

We will also estimate the interest of voters for parties, using the text of their tweets. Using a large sample of tweets posted during a month before the Irish elections, we will measure the mentions of parties or parties’ leaders and the text’s sentiment. This will indicate how much discussion there is about the elections and how people feel about the candidates. People’s intention to vote “remain” or “leave” on the European Union membership referendum will be similarly estimated from tweets containing keywords like “Brexit” or “Bremain” (which are popular references on the referendum) and applying sentiment analysis on those tweets as well.

In both cases, we will define metrics that measure the popularity of a party in the network or the tweets. We will use these metrics for prediction and compare with the actual results.

1.3 Roadmap

The work in this thesis is structured as follows:

In chapter 2, “Related Work”, we review past research on the subject of predicting political opinion through social media is reviewed.

Chapter 3, “Data”, describes the data collected from Twitter about Ireland’s elections and the UK’s referendum. This chapter also describes how the Twitter API tool is used to retrieve the data and gives a summary of data’s characteristics. The Irish Election dataset consists of the network of users, and the collection of tweets on the election. UK’s referendum dataset consists of a collection of tweets on the referendum.

Chapter 4, “Algorithmic techniques”, describes the algorithms and the metrics that we will use for the prediction task. Those techniques include the algorithms Absorbing Random Walks, PageRank and HITS that operate on the network. The text-based approaches will use sentiment analysis.

In chapter 5, “Experimental Results”, we perform predictions for the test cases and we evaluate our predictions by comparing them to the actual results.

Chapter 6, “Conclusions”, summarizes the thesis and discusses the techniques’ success and conclusions from the experiments.

2 Related Work

Even though social media is a relatively new form of communication, analyzing web data and casting predictions based on that data, is a popular subject for research. On the past few years there has been an impressive amount of work on extracting public political opinion from social media and specifically Twitter. Gayo-Avello's work [1] and Prasetyo's & Hauff 's work [2] are the main influences of this thesis. In this chapter we will provide a comprehensive survey of the field.

Twitter-based election polling is a cheap alternative to traditional polls, but data selection and data pre-processing have a considerable influence on the prediction accuracy. In literature [1],[2] so far, a Twitter-based prediction follows five steps: data collection, data filtering, de-biasing the data, prediction and evaluation of the prediction.

1. Data collection involves three important factors. The way data are accessed can vary between collecting a stream of random tweets and using a search approach with keywords or users as criteria. Which keywords or phrases are used in order to extract relevant tweets is the most important aspect of data collection. Finally, the duration of the data collection influences the accuracy of the prediction.
2. Data Filtering is used to reduce the noise in the dataset and exclude spam users or non-human users (organizations, businesses, candidates' bots, etc.). Also, users whose tweets do not originate from the election country need to be ignored.
3. De-biasing the data results in a dataset that approximates the elections voters' characteristics. Users of social networks are not necessarily representative of the overall country's population. Groups of people with different gender, age, location, education, income, etc. are underrepresented, so external sources can be used to reduce bias.
4. Prediction and accuracy of the results depend on the selected methodology. Election tweets or users mentioning a candidate or party's name can correspond to "votes". Vote per user is probably a more realistic approach than vote per tweet. For more accuracy, sentiment analysis, machine learning and crowdsourcing can be used to determine whether a user is in favor of a candidate.
5. Evaluation of the prediction is necessary to assess the success of the methodology followed. The prediction can be evaluated by computing the Mean Absolute Error to compare forecasting results to the election outcome and polls' results.

Bollen et al. (2009) [4] applied sentiment analysis to Twitter data to extract different mood states and concluded that popular events (social, political, etc.) have an effect on the public mood. They argued that Twitter data could be used for predictive purposes but didn't describe any predictive method. Sentiment analysis was also used by O'Connor et al. (2010) [5] who showed that a sentiment score produced by counting positive and negative tweets regarding a given topic, correlates with consumer confidence and presidential approval polls. However, no correlation was found between Twitter sentiment and electoral polls. On the other hand, Tumasjan et al. (2010) [6] claimed that a simple count of tweets that mention a party or a candidate's name reflects on election results. Jurgens ets al. (2011) [7] responded to that claim by pointing out that Tumasjan et al. didn't take into account all the parties but just those represented in congress and that results varied depending on the time windows used. They finally argued that a count of tweets allows no prediction of election results. Sang et al. (2012) [8] also concluded that tweet counting is not a good predictor. They found that applying

sentiment analysis improves performance and used polling data to correct demographic differences in the data showing that Twitter data on their own are not enough.

After a thorough survey, Gayo-Avello (2012) [1] summarized this research related to electoral prediction using Twitter data. He claimed that it is not possible to predict elections with Twitter and presented the flaws of past research. He interestingly points out that incumbency tends to play a major role in most of the elections and if someone attempts to predict elections with Twitter, it needs to be taken into account. He also recommends that simplistic sentiment analysis methods should be avoided because of the difficulty in detecting humor and sarcasm.

Also, silent majority is a problem in using social media to build predictive models. Mustafaraj et al. (2011) [3] showed that users engage in different ways with Twitter. Most users belong to the silent majority and aren't as active as users that post tweets more often who are referred to as the vocal minority. These two groups show different behaviors, as the silent majority users post tweets rarely and when they do, they tend to be more subjective and express personal opinions. Vocal users produce the greatest part of Twitter's content and they tend to link more to outside content. As for their retweeting behavior, silent users retweet famous people and news organizations, but don't retweet users from the vocal group. On the other hand, the vocal minority retweets other members of their community with whom they agree politically. Finally, the vocal minority distorts the representation of the public opinion in Twitter. In conclusion, Mustafaraj et al. warn researchers that all content on the web is not equal and that they should be aware of aggregating data and building predictive models upon them.

Dwi Prasetyo and Hauff (2015) [2] showed that a prediction based on Twitter data can be more accurate than traditional polls, due to Twitter's data availability on a much larger sample of users. On their use case of Indonesia's presidential elections 2014, they defined different predictors based on their hypotheses. Assigning a vote per user combined with sentiment analysis, stood out of all methods. Also, counting vote per user is usually more accurate than counting vote per tweet. The attempt of removing spam users, non-human users and slacktivists[1], as well as weighting the contribution of tweets/users to votes depending on the users' demographic information, failed to improve the forecast accuracy. In conclusion, the most basic Twitter-based predictor led to more accurate predictions than the majority of offline polls and the best performing predictors outperformed all available offline polls on the national level of Indonesia.

There are also efforts beyond the research and academic community in using Social Media for predicting election outcomes. Sensei (2013-2016) [9],[10] is a project coordinated by the University of Trento and backed by the European Commission. Partners of the project include European universities and companies (Université d'Aix Marseille, University of Sheffield, University of Essex, Teleperformance, Websays). It is complex and multileveled project that uses syntactic and semantic parsing to analyze text and speech, detects sentiment and topic of conversation.

[1] There are many types of slacktivists. Here, we refer to those users who have the goal of gaining support for a campaign, on a party's account.

Sensei's purpose is analyzing millions of online conversations to predict events, such as Brexit: the UK vote on whether to retain its EU membership. The project's hope is that sentiment analysis will prove more accurate than traditional polling systems. In the test case of the UK's referendum on 2016, the analysis of data collected from many multilingual social and news media sources, concluded in an accurate prediction. Sensei's prediction was 51.79% for LEAVE and 48.21% for STAY while the actual results where 51.90% LEAVE and 48.10% STAY.

3 Data Collection

Data collected for this thesis are divided to those regarding the Irish General Election and those regarding the UK referendum. For the Irish elections we collected the network of 17 parties, the parties' followers, the followers of followers and the friends of followers. Due to technical limitations, we used the network of the 4 most popular parties for the test case of Ireland's elections. We also collected tweets that were posted from Ireland during a month before the elections' date. Finally, we collected tweets that were posted from the United Kingdom during two weeks before the UK's referendum.

To collect the network for the Irish elections, we used Twitter's Search API. The Search API allows queries against tweets and users. Specifically, for every party we collected its followers' unique identifiers. For every user in the dataset of parties' followers, we collected the user's followers and the user's friends, who are those that the user follows.

Twitter's Streaming API returns a small random sample of all public statuses flowing through Twitter. We used the Streaming API to collect the tweets datasets for both the Irish elections and the UK's referendum. To retrieve tweets from the stream that were posted from Ireland for the Irish elections and from the UK for the Brexit referendum, we used some criteria like the user's time zone. We will describe those criteria later.

To access Twitter's Search API and Streaming API, we used the "twitter4j" library, which is a Java library designed to provide easy access to the Twitter's APIs. Twitter4j, contains functions that return a given user's followers or friends. We used those functions to build a Java program to collect the network for Ireland's parties. The twitter4j library also provides access to Twitter's sample stream of tweets. We collected tweets from the stream in real time for both Ireland's elections and UK's referendum.

More information on Twitter's APIs and examples of the related necessary code to retrieve data from Twitter are available in the Appendix.

An overview of the amount of data we collected:

Data	Collection Period	Dates	File size	Dataset size
Twitter network (17 Irish parties)	3 days	February 2016	7Gb	nodes: 126.656.135 edges: 368.411.708
Tweets Ireland (Irish elections)	27 days	4/2-1/3/2016	900Mb	tweets: 1.566.000
Tweets UK (Brexit referendum)	13 days	17/6-30/6/2016	3,3Gb	tweets: 839.756

Table 1: Datasets collected and characteristics

3.1 Ireland

Regarding the Irish elections, we concentrate on the four most popular Irish parties. It is interesting to state some facts on those parties, such as their political position, which will help explain each party's popularity on Twitter.

Fine Gael is a center-right party and its ideology is defined by liberal conservatism, Christian democracy, liberalism and pro-Europeanism. Since Fine Gael's establishment in 1933, it has always taken the second place in general elections and participated in the Irish parliament as the opposition party. In some cases it participated in the government by forming coalitions with the Labour party. In the Irish elections of 2011, Fine Gael took the first place. However, its

popularity has decreased and in 2016 received only the 25,5% of the votes. That led to forming a minority government supported by Fianna Fail which took the second place with 24,3%.

Fianna Fail is a center-left party which ideologically lies to Irish republicanism, conservatism, populism and pro-Europeanism. Since 1926, it often took the first place in the elections and participated in the parliament either as the government or as the primary opposition. Fianna Fail's popularity decreased in 2011, when it fell to second place in the elections. From 41,6% in 2007, its percentage of votes dropped to 17,5% in 2011.

Sinn Fein, a left party supporting Irish republicanism, left-wing nationalism, democratic socialism and Euroscepticism, has historically been associated with the Provisional Irish Republican Army (IRA) and took its current form in 1970. Since then, Sinn Fein gradually rises and finally took the third place in the 2016 elections with 13,8%.

The Labour party is a center-left party supporting social democracy and pro-Europeansim. Since 1912, it often took the third place in Irish general elections and was often the Opposition of the Irish government. Recently, in 2011, the Labour party came second in the general elections with 19,5%. However, in 2016 it received only the 6,6% of the votes.

We will examine two datasets for the Irish elections: users' network and tweets

3.1.1 Network

We collected the network of twitter users, for 17 Irish parties' twitter accounts. Its file's size is approximately 7Gb. After specifying the parties' accounts, their **followers** were collected. Afterwards, we collected the followers of the followers and the friends of the followers. We will refer to those users as **ffollowers** and **ffriends** respectively. The network consists of 126m nodes and 368m edges.

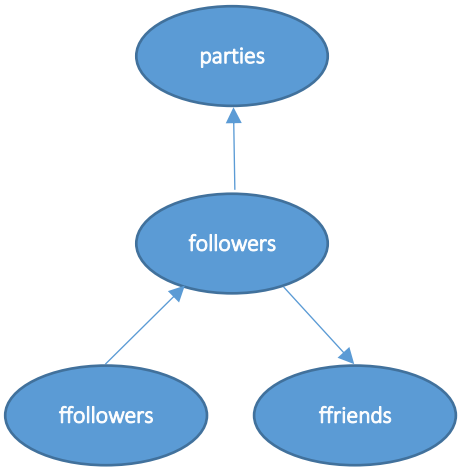


Figure 1: Network's subsets and their connections

To collect the network for the Irish parties, we used the twitter4j Java library. The twitter4j library provides access to Twitter's API and allows us to retrieve specific data, such as a user's followers and friends. Every twitter user has a unique identifier. We collected all parties' ids by searching for the parties' names on the online API Console Tool[2] provided by twitter. A list of the parties' ids is provided in the Appendix.

[2] https://dev.twitter.com/rest/tools/console

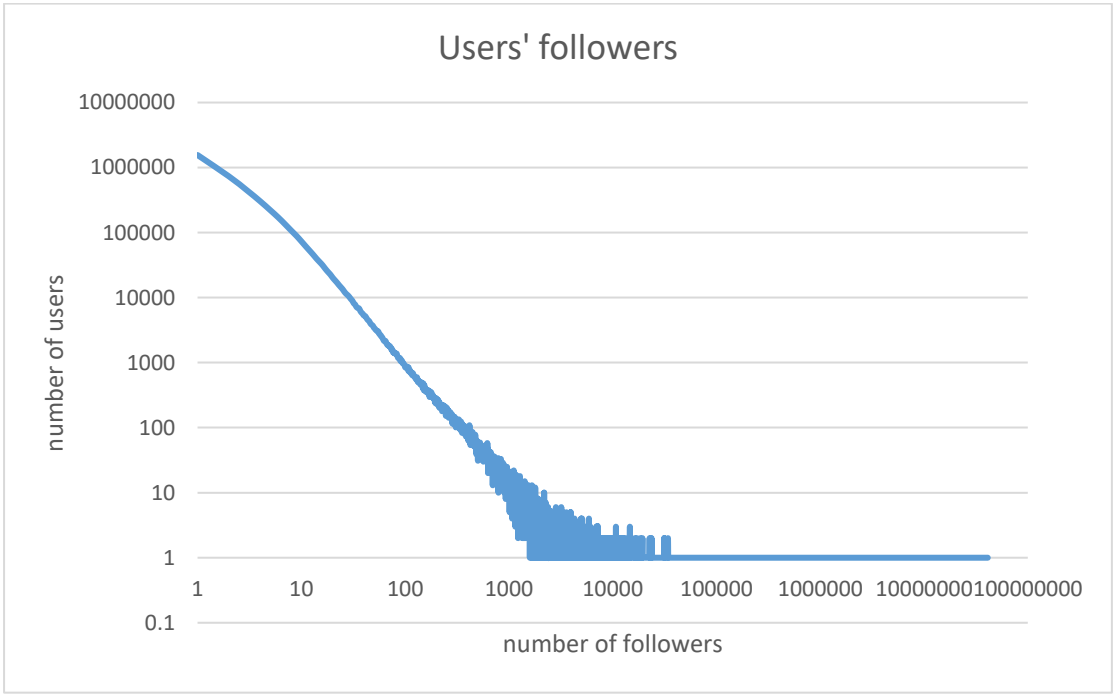
For every party's twitter id, we collected its followers' ids using the getFollowersIDs function from the library. Afterwards we used the getFollowersIDs and getFriendsIDs functions to collect the ffollowers and ffriends.

We will try to predict the election outcome only for the four most popular Irish parties. They are Fine Gael, Fianna Fail, Sinn Fein and the Labour Party. Specifically, on the day of the Irish General Election 2016, February 26th, their twitter accounts had the following numbers of followers:

Party	# followers
FG (Fine Gael)	21.087
FF (Fianna Fail)	22.135
SF (Sinn Fein)	44.970
LP (Labour Party)	29.941
Sum total	118.133

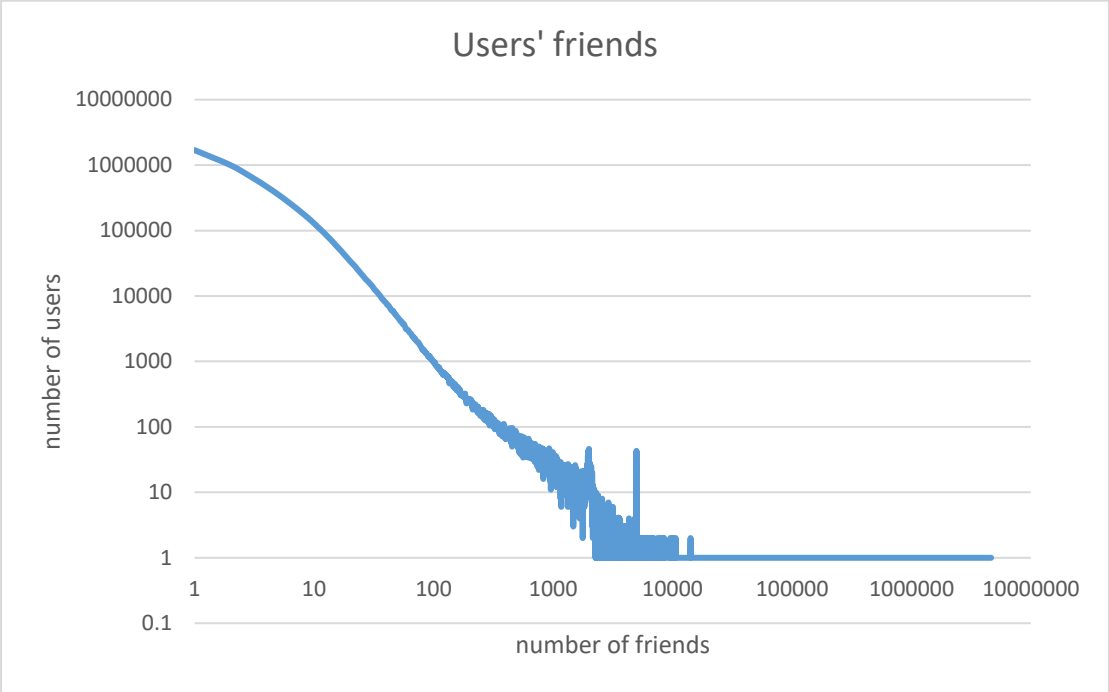
Table 2: Four popular parties' number of followers

On the graphs below, we can see the number of followers and friends users have in the network.



Graph 1: Users' number of followers

As expected, users tend to have more friends than followers, since Twitter is a social network were users often follow news media and famous people.



Graph 2: Users' number of friends

3.1.2 Tweets

The Tweets dataset contains tweets from Twitter's Sample Stream, dated from February 4th to March 1st, 2016. The final size of this dataset is 900Mb and the total number of collected tweets were 1.566.000. In order to locate those tweets posted probably from Ireland, we filtered the tweets using the following criteria:

- The tweet's coordinates represent Ireland's location. Ireland's latitude is between 51.5 and 55.5 and its longitude is between -10.7 and -5.9.
- The user who posted the tweet is probably living in Ireland and at the same time the tweet's language is either "en" for English or "und" if no language could be detected. We consider that a user is living in Ireland if the user's time zone is "Dublin", or the user's coordinated universal time (utc) offset is 0.

To collect tweets for the Irish elections, we used the twitter4j Java library. The twitter4j library provides access to Twitter's Public Stream of tweets and allows us to retrieve those random tweets. Using the library's function "sample" we receive all tweets on the stream. We created a listener function that applied the criteria mentioned above on the stream and we collected tweets from Ireland.

From the set of tweets we collected, we will use those tweets that have time zone "Dublin" or user's location contains "Ireland" and the text of tweets contains either one of the following keywords:

Keywords for Irish elections			
for Fine Gael	for Fianna Fail	for Sinn Fein	for Labour Party
fine gael	fianna fail	sinn fein	labour party
finegael	fiannafail	sinnfein	#labour
@finegael	@fiannafailparty	@sinnfeinireland	@labour
enda kenny	micheal martin	gerry adams	joan burton
endakenny	michealmartin	gerryadams	joanburton
endakennytd	michealmartintd	gerryadamssf	@joanburton
@endakennytd	@michealmartintd	@gerryadamssf	

Table 3: Keywords for Ireland's set of tweets

Keywords include parties' names and their leaders' names. We will test five sets of tweets: Tweets that mention the partys' Twitter accounts (name starts with @), tweets that mention the leaders' accounts, tweets with text that includes a party's name in any form (that automatically also includes mentions @ and hashtags #), tweets with text that includes a leader's name in any form and tweets that contain any of the keywords related to a party, as displayed on Table 3.

3.2 United Kingdom

The United Kingdom European Union membership referendum, took place on Thursday, June 23, 2016. People were asked to vote "remain" or "leave" as to whether UK should remain a member of the European Union or not. On June 2016, we collected tweets from the United Kingdom. Our predictors on the referendum are based on content and sentiment analysis.

The UK Tweets dataset contains tweets from Twitter's Sample Stream, dated from June 17th to June 30th, 2016. The final size of this dataset is 3,3Gb and the total number of collected tweets were 839.756. In order to locate those tweets posted probably from the United Kingdom, we filtered the tweets using the following criteria:

- The tweet's coordinates represent Ireland's location. Ireland's latitude is between 49.8 and 61.3 and its longitude is between -8.8 and 2.1.
- The user who posted the tweet is probably living in Ireland and at the same time the tweet's language is either "en" for English or "und" if no language could be detected. We consider that a user is living in Ireland if the user's time zone is "London", or the user's coordinated universal time (utc) offset is 3600 (summer time in UK).

We collected tweets from the UK by using the twitter4j library function "sample", the same way we collected tweets for the Irish elections.

From the set of tweets we collected, we will use those tweets that have time zone "London" or user's location contains "UK" and the text of tweets contains either one of the following keywords:

Keywords for Brexit	Keywords for Bremain
#brexit	#bremain
#leave	#voteremain
#voteleave	#remain
#leaveeu	#remaineu
#takecontrol	bremain
brexit	stay
leave && referendum	remain && referendum
leaveeu	remaineu
voteleave	voteremain
takecontrol	

Table 4: Keywords for the UK's set of tweets

4 Algorithmic Techniques

In this chapter, we will describe the algorithmic techniques that we will use. We will also define the metrics that we will use as predictors for the election and referendum outcomes. We will use E to denote the actual results. $E(p)$ will denote the percentage of party p in the elections. The normalized $\overline{E}$, is defined in order to compare actual results with our predictors and evaluate each technique:

$$\overline{E}(p) = \frac{E(p)}{\sum_{x \in parties} E(x)}$$

4.1 Network

To approximate a prediction of each party's percentage of votes, four algorithms are applied on the network to produce a metric: the calculation of the percentage of followers for each party, Absorbing Random Walks, Page Rank and HITS algorithm.

The network is composed by the following subsets: **parties**, **followers**, **ffollowers** and **ffriends**. As we described earlier, parties don't have outgoing edges, meaning that a node may belong to the parties set and the ffriends set at the same time. Any other node may belong to more than one of the sets of followers, ffollowers and ffriends. Nodes that belong to followers, necessarily have outgoing edges that point to parties. Nodes from the ffollowers set point to followers and followers point to ffriends. Nodes that have at least one outgoing edge are considered to be **voters**, so voters = followers ∪ ffollowers. We will refer to nodes that belong to those sets as **p** for parties, **f** for followers, **ff** for ffollowers and **fr** for ffriends. That means that $p \in parties$, $f \in followers$, $ff \in ffollowers$, $fr \in ffriends$, $v \in voters$.

4.1.1 Followers percentage

The number of a party's followers is another popularity index. We assume that the party with the most followers will perform better on the elections. We define the metric V_f:

$$V_f(p) = \sum_{f \in followers} \mathbb{I}_f(p, f),$$

where $\mathbb{I}_f$ is the indicator function: $$\mathbb{I}_f(p, f) = \begin{cases} 1, if\ f \to p \\ 0, \ else \end{cases}$$

The normalized metric $\overline{V}_f$ gives us a prediction of the percentage of a party:

$$\overline{V}_f(p) = \frac{V_f(p)}{\sum_{x \in parties} V_f(x)} \qquad (1)$$

4.1.2 Absorbing Random Walk Metric

When we take a walk in a graph, if we reach a node without any outgoing edges the random walk will stay at that node. We call such nodes absorbing nodes. For every absorbing node z in a graph, and every non-absorbing node v, we can compute the probability $P(z|v)$ to end up in that absorbing node if we start our walk from node v.

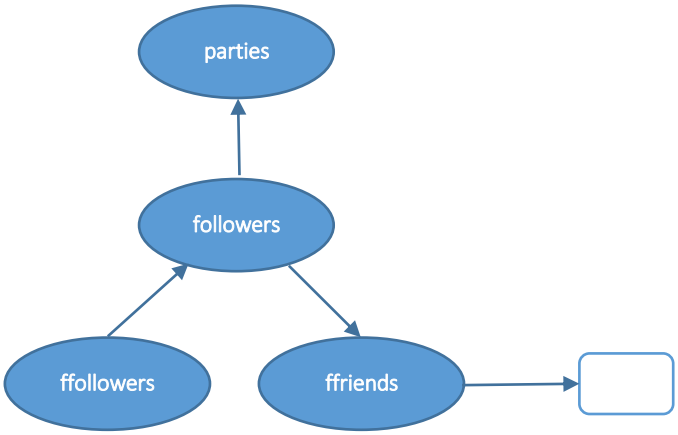


Figure 2: Network's subsets and mental absorbing nodes

We suppose that a user who follows a party, is probably supporting this party and also that users' connections in Twitter generally show that they approve of their friends' opinions. We consider that the probability of a user to end up to a party by taking a walk on the users' network represents the probability to vote this party.

In our network, parties are absorbing nodes and every node with no outgoing edges is considered to have an edge to a virtual absorbing node (Figure 2). In order to exclude inactive users and users that are very popular, we will create five networks for the four popular parties, by setting different min and max limits to the parties' followers. For every network we will include every follower f that its number of followers from the ffollowers set is between those limits. We will also include every ff ffollower that follows f and every fr ffriend followed by f. We will use this algorithm to test each of those five networks.

For every non-absorbing node x, there is a probability that a random walk will lead to a specific absorbing node. If node x has d outgoing edges that point to nodes $y_1, y_2, \dots, y_d$, then a random walk from x will end to an absorbing node(party) with probability $P(p|x)$:

$$P(p|x) = \frac{1}{d}\sum_{i=1}^{d} P(p|y_i)$$

For every party p, it's $P(p|p) = 1$ and for every other party p' it's $P(p|p') = 0$. We initialize the probability $P(p|x) = 0.25$, for every party p and for every non-absorbing node x, meaning that at first we consider that every node x, is equally absorbed by each party. The Absorbing Random Walks algorithm computes the probability $P(p|x)$ for every node x in the graph, depending on the node's outgoing edges as we described. Probabilities for all nodes and for every party are updated until convergence.

Nodes $v \in voters$ are non-absorbing nodes that have probability $P(p|v) > 0$ for at least one party. This means that voters will be absorbed by the parties.

We define the metric V_a:

$$V_a(p) = \frac{1}{n}\sum_{v \in voters} P(p|v)\,,\ where\ n\ is\ the\ number\ of\ voters.$$

The value $V_a(p)$, of the metric V_a, will be used as a predictor for the election percentage of the party p.To evaluate the prediction, we will compute the normalized metric:

$$\overline{V}_a(p) = \frac{V_a(p)}{\sum_{x \in parties} V_a(x)} \qquad (2)$$

In every election, voters have only one vote. We suppose that Twitter users who belong to the voters set of the network must choose to vote for only one party. We consider that, a voter v will vote for the party p for which it has the greatest probability $P(p|v)$. So, every party p collects those voters v that are more likely to be absorbed by the party p. If a voter v has equal probabilities $P(p|v)$ for two or more parties, then the voter is considered to be undecided or absentee and doesn't vote.

We define the metric V_i for parties p as another predictor for parties' percentages:

$$V_i(p) = \sum_{v \in voters} \mathbb{I}(p, v),$$

where $\mathbb{I}$ is the indicator function:

$$\mathbb{I}(p, v) = \begin{cases} 1, if\ p = \arg\max\limits_{x \in parties} P(x|v) \\ 0,\ \ else \end{cases}$$

The value $V_i(p)$ for a party p, represents the party's total number of votes. The normalized metric $\overline{V}_i$ will be used to predict percentages of the election outcome and compare with the actual results:

$$\overline{V}_i(p) = \frac{V_i(p)}{\sum_{x \in parties} V_i(x)} \qquad (3)$$

4.1.3 PageRank

PageRank algorithm's principle is that "good authorities should be pointed by good authorities". If popular people support a party, then the party is popular. Users are popular not only if many users follow them, but also if many other popular users follow them. We consider that a party's performance depends on its followers' popularity. By ranking users according to their popularity in a network, we find which are the most popular and how they affect the parties they follow.

We will set different min and max limits to the parties' followers and create five networks for the four most popular parties. Every follower f that has number of followers between those limits, will be included in the network. All five networks will include all nodes p from the parties set, the selected nodes f from the followers set, the ff followers of the followers and fr friends of the followers, and the edges among them. We will use this algorithm to test each of the five networks. We will also test the five networks that include users only from the set of parties and the set of followers.

For every node x we compute a weight w_x which shows the node's ranking in the graph. Every node's weight is initialized to 1/n, where n is the total number of nodes. Each node then distributes the authority value they have to their neighbors. The authority value of each node is the sum of the authority fractions it collects from its neighbors. With probability a = 0.15, a node/user follows one of its connections/friends. Every node x has weight w_x:

$$w_x = (1-a)\sum_{u \to x} \frac{1}{d_{out}(u)} w_u + a\frac{1}{n} \quad , \quad \text{where } a = 0.15$$

The PageRank algorithm computes weights for all nodes until convergence.

The algorithm will be applied on the complete network and also on the network among parties' followers. Every follower f has a weight w_f. To forecast a party's performance on the elections, on both cases, we will use the metric V_p :

$$V_p(p) = \sum_{f \in followers} w_f$$

Each party's p ranking $V_p(p)$, is the sum of its followers' weight. And the prediction of the party's percentage is given from the normalized metric $\overline{V}_p$:

$$\overline{V}_p(p) = \frac{V_p(p)}{\sum_{x \in parties} V_p(x)} \qquad (4)$$

After applying the PageRank algorithm on the network of followers, we will measure the average weight of the followers of each party. We define the metric V_g:

$$V_g(p) = \frac{1}{n} \sum_{\substack{f=1 \\ f \in followers}}^{n} w_f \quad , where\ n\ is\ the\ number\ of\ followers$$

Each party's p ranking $V_g(p)$, is the average of its followers' weight. And the prediction of the party's percentage is given from the normalized metric $\overline{V}_g$:

$$\overline{V}_g(p) = \frac{V_g(p)}{\sum_{x \in parties} V_g(x)} \qquad (5)$$

4.1.4 HITS

We consider that if we have users who follow popular users, then every user they follow is popular. If a user follows many popular users, then he also follows a popular party.

HITS algorithm's principle is that "authority is not necessarily transferred directly between authorities". Nodes have two identities and are called hubs and authorities. Good hubs point to good authorities and good authorities are pointed by good hubs. We can compute hub and authority weights for nodes in a graph and determine which are the best hubs and the best authorities.

The hub weight is the sum of the authority weights of the authorities pointed to by the hub:

$$h_x = \sum_{y:x\to y} a_y$$

The authority weight is the sum of the hub weights that point to this authority:

$$a_x = \sum_{y:y\to x} h_y$$

The HITS algorithm computes hub and authority weights. At first, all weights are initialized to 1. To normalize weights, all authorities are divided by the max authority weight in every round. Hub and authority weights are calculated in turns, until convergence. The algorithm is applied on the five networks among all users and on the five networks among parties' followers, as described for the PageRank Algorithm.

For parties p, the metric V_h:

$$V_h(p) = a_p, where\ a_p = \sum_{\substack{f:f\to p \\ f\in followers}} h_f$$

To compute $V_h(p)$ values, all nodes of the network are authorities and hubs. The normalized metric is:

$$\overline{V}_h(p) = \frac{V_h(p)}{\sum_{x\in parties} V_h(x)} \qquad (6)$$

We will also use the authority weight of followers of every party to measure each party's importance. For the five networks that include all users and the five networks among parties' followers and the parties, we will use the metric V_s:

$$V_s(p) = \sum_{\substack{f:f\to p \\ f\in followers}} a_f$$

The normalized metric is:

$$\overline{V}_s(p) = \frac{V_s(p)}{\sum_{x\in parties} V_s(x)} \qquad (7)$$

After we apply HITS algorithm on the complete network, we will compute the average authority weight of the followers for every party. We define the metric V_m:

$$V_m(p) = \frac{1}{n} \sum_{\substack{f=1 \\ f:f\to p \\ f\in followers}}^{n} a_f\ , \qquad where\ n\ is\ the\ number\ of\ followers$$

We will normalize the metric as follows:

$$\overline{V}_m(p) = \frac{V_m(p)}{\sum_{x\in parties} V_m(x)} \qquad (8)$$

4.2 Tweets

The methodology applied on tweets, aims to analyze the text of the tweets and find indications of support for the different parties. We will use Stanford's Sentiment Analysis tool "Stanford CoreNLP API" for sentiment analysis to determine the polarity of the tweets. It is known that sentiment analysis is not very accurate but we want to experiment and see if it can help in the prediction cast.

We define the metric T_m for a keyword $k \in keywords$:

$$T_m(k) = \sum_{t \in tweets} \mathbb{I}_m(k,t)$$

where $\mathbb{I}_m$ is the indicator function:

$$\mathbb{I}_m(k,t) = \begin{cases} 1, if\ k \subseteq \text{t} \\ 0, \quad else \end{cases}$$

The metric T_m is actually a counter. It counts how many tweets contain at least one keyword from the set we will define. This is a vote per tweet approach. For a keyword k, the value $T_m(k)$ will be used as a predictor for the Irish elections and for the UK referendum.

To normalize the metric T_m, for party or referendum option p:

$$\overline{T}_m(p) = \frac{T_m(k \in keywords\ of\ p)}{\sum_{x \in parties} T_m(w \in keywords\ of\ x)} \qquad (9)$$

As metric T_m is actually a counter of tweets, we will also use this metric to every set of tweets that includes tweets from only one user. The party p that will have the max value $\overline{T}_m(p)$, will be given one vote from that user. We will again use the metric $\overline{T}_m(p)$, with a vote per user approach.

After, we will apply sentiment analysis on tweets that contain relative keywords. Every tweet is valued according to its sentiment provided by Stanford CoreNLP API, as very positive, positive, neutral, negative or very negative. At first we will ignore how positive or negative a tweet scores, meaning that very positive and positive tweets will be considered as positive. For negative tweets respectively.

We compute the metric T_s:

$$T_s(p) = \frac{positive}{negative}$$

For party or referendum option p, the predictor $\overline{T}_s(p)$, gives the percentage of p on the election:

$$\overline{T}_s(p) = \frac{T_s(p)}{\sum_{x \in parties} T_s(x)} \qquad (10)$$

Then, we will use the sentiment score of every tweet to test the tweets. We compute the metric T_o:

$$T_o(p) = \frac{positive\ score}{negative\ score}$$

For party or referendum option p, the predictor $\overline{T}_o(p)$, gives the percentage of p on the election:

$$\overline{T}_o(p) = \frac{T_o(p)}{\sum_{x \in parties} T_o(x)} \qquad (11)$$

Those two metrics will be used on two sets of tweets. Text of tweets contains at least a word from a set of keywords for each party or referendum option as we described previously. For the Irish General Election we will use the tweets we collected from Ireland (tweets have time zone "Dublin" or user's location contains "Ireland") a month before the election and for the UK referendum we will use the set of tweets from the UK (tweets have time zone "London" or user's location contains "UK") we collected two weeks before the referendum.

5 Experimental Results

5.1 Ireland

5.1.1 Network

To test our algorithms, we created five networks for the four most popular Irish parties, by setting different min and max limits to the parties' followers. We include nodes that are parties' followers, only if the number of followers they have is between the limits we define for each network. Then, we include the ffollowers and ffriends for the selected followers and we include the parties as well. Those limits are meant to exclude inactive users and users that are very popular. If a popular user has many followers and follows a party, the "ffollowers" set will have users that are irrelevant. For example, Barack Obama follows the Labour Party and has 40 million followers that probably aren't potential voters. So users like Obama need to be excluded from the "followers" set.

To avoid inactive users, we always use the minimum 5 limit of ffollowers for every follower. We also test different maximum limits to the number of followers that a party's follower has, from 100 to 2000. In network #1 we include users that probably aren't famous and in the following networks we include more and more popular users.

Network	min followers	max followers	edges	nodes
1	5	100	3.891.563	757.151
2	5	200	8.621.758	1.527.954
3	5	500	21.735.320	3.305.023
4	5	1000	35.030.093	4.940.028
5	5	2000	46.649.047	6.389.524

Table 5: The Five Networks characteristics

As Figure 3 shows, every network consists of four sets: parties, parties' followers, the followers of the parties' followers which are called ffollowers, and the friends of the parties' followers which are called ffriends. All networks include only the four popular parties but they don't include any outgoing edges for parties' nodes. Users that have at least one friend, including the parties, are considered to be voters: voters = followers ∪ ffollowers.

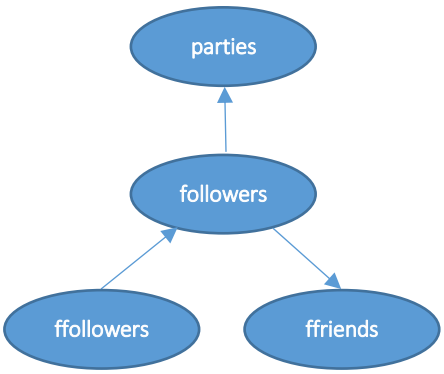


Figure 3: Networks' form of subsets

Statistics on the sets described above are presented next:

Network:	#1	#2	#3	#4	#5
nodes	757.151	1.527.954	3.305.023	4.940.028	6.389.524
parties	4	4	4	4	4
followers	33.753	44.725	59.614	67.456	71.688
ffollowers	733.581	1.504.704	3.285.470	4.924.035	6.376.587
ffriends	441.556	942.611	2.090.797	3.094.025	3.863.584
voters	757.147	1.527.950	3.305.019	4.940.024	6.389.520

Table 6: Networks subsets

We observe that the followers set in network #1 is half the size of the followers set in network #5, even though network #5 has almost ten times the number of nodes in network #1. This means that most users have less than 100 followers, but every popular user we include in the network makes it much bigger. Popular people appear to be more influential to the network than others, if a popular user's followers are influenced by his political opinions.

Followers that follow only one party, are considered to be the "faithful" followers. The tables below, contain the numbers of parties' common followers and on the diagonal, there are the "faithful followers" of each party.

Network #1: 1.139 followers are common for all four parties.

Party	#followers	common followers between parties and faithful followers			
		FG	FF	SF	LP
FG (Fine Gael)	7.476	3.201 (42,82%)	3456	1739	3004
FF (Fianna Fail)	8.162	3456	4.663 (42,84%)	2152	2873
SF (Sinn Fein)	17.563	1739	2152	19.644 (84 %)	1677
LP (Labour Party)	10.234	3004	2873	1677	8.562 (61,72%)

Table 6: Network #1 common & faithful followers

Network #2: 1.630 followers are common for all four parties.

Party	#followers	common followers between parties and faithful followers			
		FG	FF	SF	LP
FG (Fine Gael)	10.051	4.127 (41,06%)	4800	2458	4204
FF (Fianna Fail)	10.885	4800	4.663 (42,84%)	2989	4039
SF (Sinn Fein)	23.398	2458	2989	19.644 (84 %)	2375
LP (Labour Party)	13.873	4204	4039	2375	8.562 (61,72%)

Table 7: Network #2 common & faithful followers

Network #3: 2.402 followers are common for all four parties.

Party	#followers	common followers between parties and faithful followers			
		FG	FF	SF	LP
FG (Fine Gael)	13.674	5.295 (38,7%)	6818	3564	6054
FF (Fianna Fail)	14.771	6818	5.994 (40,5%)	4280	5840
SF (Sinn Fein)	31.211	3564	4280	25.817 (82,7%)	3509
LP (Labour Party)	19.247	6054	5840	3509	11.593 (60,2%)

Table 8: Network #3 common & faithful followers

Network #4: 2.937 followers are common for all four parties

Party	#followers	common followers between parties and faithful followers			
		FG	FF	SF	LP
FG (Fine Gael)	15.776	5.860 (37,1%)	8084	4309	7230
FF (Fianna Fail)	16.999	8084	6.604 (38,8%)	5141	7002
SF (Sinn Fein)	35.285	4309	5141	28.802 (81,6%)	4280
LP (Labour Party)	22.413	7230	7002	4280	13.265 (59,2%)

Table 9: Network #4 common & faithful followers

Network #5: 3.258 followers are common for all four parties.

Party	#followers	common followers between parties and faithful followers			
		FG	FF	SF	LP
FG (Fine Gael)	16.950	6.121 (36,1%)	8850	4728	7944
FF (Fianna Fail)	18.275	8850	6.906 (37,8%)	5624	7730
SF (Sinn Fein)	37.445	4728	5624	30.352 (81%)	4744
LP (Labour Party)	24.260	7944	7730	4744	14.187 (58,5%)

Table 10: Network #5 common & faithful followers

We observe that Fine Gael and Fianna Fail have many common followers. An explanation to this is that those are the most popular parties and they received the higher number of votes in Ireland in past elections. On the other hand, Sinn Fein has the most "faithful" followers and has the less common followers with other parties, even though it is the party with the most followers. We assume that it is an indication of the ideological differences between Sinn Fein and the other parties, since it is the only Eurosceptic party.

In order to compare results with the actual results of the election, parties' percentages are normalized:

Party	Actual Result $\boldsymbol{E(p)}$	Normalized result $\overline{\boldsymbol{E}}\boldsymbol{(p)}$
Fine Gael (FG)	25,5 %	36,3 %
Fianna Fail (FF)	24,3 %	34,6 %
Sinn Fein (SF)	13,8 %	19,7 %
Labour Party (LP)	6,6 %	9,4 %
	70,2 %	**100 %**

Table 11: Parties' actual results on the Irish General Elections 2016

First, we compute the metric $\overline{V}_f$, which is the percentage of followers for each party:

Network	parties' followers				$\overline{\boldsymbol{V}}_{\boldsymbol{f}}\boldsymbol{(p)}$ parties' followers percentage				MSE
	FG	FF	SF	LP	FG **36,3%**	FF **34,6%**	SF **19,7%**	LP **9,4%**	
1	7.476	8.162	17.563	10.234	17,21	18,79	40,43	23,56	35,28126
2	10.051	10.885	23.398	13.873	17,26	18,70	40,19	23,83	35,26365
3	13.674	14.771	31.211	19.247	17,33	18,72	39,55	24,39	35,08675
4	15.776	16.999	35.285	22.413	17,43	18,78	39,00	24,77	34,85163
5	16.950	18.275	37.445	24.260	17,48	18,85	38,63	25,02	34,70598

Table 12: Parties' followers percentage, actual results and Mean Squared Error

We observe that the followers' percentage almost doesn't change for the different networks. Sinn Fein appears to receive the highest values, since it has the most followers. The percentage of followers doesn't seem to be an effective metric to predict the elections' results.

Absorbing Random Walks

Network	$\overline{V}_a$				MSE	$\overline{V}_i$ (users have one vote)				MSE
	FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%		FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%	
1	14,1	15,41	41,33	29,05	41,41	11,82	13,14	48,64	26,4	46,75
2	13,45	14,72	42,32	29,36	42,74	11,13	12,41	48,95	27,5	48,05
3	12,35	13,0	40,92	33,5	45,51	10,38	10,88	48,33	30,39	49,94
4	11,83	12,97	41,98	33,07	46,07	9,63	10,81	47,74	31,79	50,64
5	12,11	13,4	41,6	32,78	45,39	8,69	10,07	47,29	33,93	52,22

Table 13: Absorbing Random Walks results, actual results and evaluation

Results seem to correlate with a party's followers. Sinn Fein party, which was elected third, appears to be the most popular. Fine Gael and Fianna Fail are close as they were in actual elections results, but we need to keep in mind that they have a close number of followers in Twitter. The Labour party's actual result is far from the predicted outcome.

An interesting fact is that Network #1 seems to be the best regarding results. Not including popular users helps average citizens to be represented. However, more de-biasing of the data could help to better approach the actual results of the elections.

PageRank

Using PageRank algorithm for the network of party's followers and for the network of all users, weights of every party's followers sum up to the following values:

Network	$\overline{V}_p$ on networks of followers				MSE	$\overline{V}_p$ on networks of all users				MSE
	FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%		FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%	
1	17,7	19,9	42	20,4	34,35	17,4	19,3	37,9	25,4	34,32
2	19,1	20,5	40	20,4	32,05	17,3	18,7	40,2	23,8	35,23
3	20,5	21,6	35,3	22,6	28,91	17,3	18,7	39,6	24,4	35,14
4	20,9	22,1	33,1	23,9	27,98	17,4	18,8	39	24,8	34,87
5	20,8	22,6	31,5	25	27,69	17,5	18,9	38,6	25	34,64

Table 14: PageRank results, actual results and evaluation

PageRank algorithm performs better than Absorbing Random Walks. However, results don't reflect the actual percentages of parties' votes in the elections 2016. On the networks of followers, we observe that as the network gets bigger, Fianna Fail takes the third place while at first it took the second place with Labour party having an equal percentage. On the networks of all users, Fianna Fail is always the second party. In this case, values don't seem to be affected by the size of the network and the Mean Squared Error remains higher comparing to the networks of followers.

It is interesting that for bigger networks when operating on networks of followers, PageRank's results seem to approach more to the actual results as Sinn Fein's percentage drops and the Mean Squared Error reduces.

We also measured the average weight of the followers of every party after applying PageRank on the network of followers.

Network	$\overline{V}_g$ on networks of followers				MSE
	FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%	
1	25,78	26,50	26	21,72	19,17
2	27,33	27,09	24,5	21,08	17,21
3	28,41	27,79	21,52	22,28	16,66
4	28,68	28,07	20,24	23,01	16,91
5	28,32	28,48	19,42	23,78	17,54

Table 15: PageRank average results of followers

This technique performs better than others using PageRank algorithm. Fine Gael and Fianna Fail are close and take the first and second place and MSE reduces. While in Network #2 results seem to correlate with the actual results regarding the ranking of parties, however, the Labour Party generally appears to have a percentage significantly greater than the actual percentage it received.

HITS

HITS algorithm converged to authority and hub values. If results are normalized with the metric $\overline{V}_h$ for the networks of followers and the networks that include all users:

Networks	$\overline{V}_h$ on networks of followers				MSE	$\overline{V}_h$ on networks of all users				MSE
	FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%		FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%	
1	16,40	18,27	46,39	18,94	38,28	23,90	25,79	28,82	21,47	21,45
2	17,01	18,69	44,52	19,78	36,72	24,04	25,67	28,26	22,01	21,50
3	17,99	19,53	41,21	21,27	34,14	23,77	25,36	27,93	22,92	22,20
4	18,87	20,33	38,71	22,09	32,09	23,88	25,25	27,24	23,61	22,37
5	20,07	21,67	35,56	22,70	29,30	23,86	25,23	26,71	24,2	22,59

Table 16: HITS results, actual results and evaluation

HITS' results also show that Sinn Fein is the most popular party. It is interesting that for bigger networks of followers, the Mean Squared Error reduces and that even though the Labour party's percentage is the second highest, its distance from Fine Gael and Fianna Fail is less than it was in some of the previous metrics.

Applying HITS algorithm on networks of all users and using the metric $\overline{V}_h$, unlike PageRank ($\overline{V}_p$), performs better and reduces the Mean Squared Error. However, we don't observe any changes in the ranking of parties.

To examine the sum of authority values of parties' followers on networks of followers and networks of all users, we use the metric $\overline{V}_s$. We observe that in this case HITS algorithm performs better on the networks of all users as well.

The results of HITS algorithm using the metric $\overline{V}_s$:

Networks	$\overline{V}_s$ on networks of followers				MSE	$\overline{V}_s$ on networks of all users				MSE
	FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%		FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%	
1	16,88	19,30	47,05	16,75	37,59	22,71	24,81	33,19	19,27	23,66
2	19	20,05	43,17	17,78	33,64	24,22	24,59	30,66	20,53	22,13
3	22	22,79	34,1	21,11	26,23	25,19	25,67	25,94	23,19	20,79
4	21,91	22,90	33,70	21,47	26,18	24,82	25,42	25,75	23,98	21,56
5	21,55	23,41	32,44	22,6	26,06	24,32	25,76	24,69	25,23	22,29

The metrics $\overline{V}_h$ and $\overline{V}_s$ on the networks of followers, present similar results. The Mean Squared Error reduces as the network gets bigger but Sinn Fein party that has the most followers, remains first.

The values of the metric $\overline{V}_s$ on the networks of all users are more balanced. However, results are better when we apply HITS and calculate the average authority value for the followers of every party, using the metric $\overline{V}_m$:

Networks	$\overline{V}_m$ on networks of all users				MSE
	FG 36,3%	FF 34,6%	SF 19,7%	LP 9,4%	
1	32,3	30,29	17,57	19,84	12,16
2	30,84	30,86	19,19	19,11	11,76
3	32,8	30,94	14,8	21,46	13,96
4	32,31	30,71	15	21,98	14,53
5	31,58	31,02	14,51	22,89	15,62

We observe that the MSE is lower and that this technique provides better results, closer to the actual percentages of the election. The labour party appears as the third party with a rather high value of the metric. For the first two parties, Fine Gael and Fianna Fail, the values are very close to the actual results and on most Networks the prediction of ranking is correct.

It is obvious that HITS performed better than all the other algorithms. HITS algorithm shows a better balance for the parties' percentages and has the smallest MSE. PageRank was the second best algorithm and as the network gets bigger, PageRank improves. On the other hand, Absorbing Random Walks has the greater MSE and rather unexpectedly gets worse as the network gets big. An explanation to this is that Absorbing Random Walks reinforces parties with followers that are popular users with many followers and Fine Gael and Fianna Fail are the parties with the less followers. So adding more popular users, strengthens the parties Sinn Fein and the Labour party. A prediction using the follower's percentage is equal on almost all networks because the parties' followers that are popular and are added in bigger networks is proportional to the number of followers each party has.

It is safe to say that we cannot predict the elections using just the network of parties. The main conclusion from these experiments is that a party's number of followers plays the most important role in network metrics, and since it's not always proportional to the actual popularity of the party, we need more data to make a forecast for an election.

5.1.2 Tweets

Statistics on tweets:

Tweets in total	1.566.000
Tweets with user's time zone "Dublin"	130.724
Tweets with time zone "Dublin" or location "Ireland"	145.506
Tweets with #ge2016	46
Tweets with #ge16	938
hashtags containing word "election"	40
tweet's text containing word "election"	761

Table 17: Statistics on Ireland's tweets

We observe that there are only a few tweets that discuss the elections. We will see that even less tweets talk about the parties and their leaders.

We remind the keywords that were used to select tweets for each party:

Keywords for Irish elections			
for Fine Gael	for Fianna Fail	for Sinn Fein	for Labour Party
fine gael	fianna fail	sinn fein	labour party
finegael	fiannafail	sinnfein	#labour
@finegael	@fiannafailparty	@sinnfeinireland	@labour
enda kenny	micheal martin	gerry adams	joan burton
endakenny	michealmartin	gerryadams	joanburton
endakennytd	michealmartintd	gerryadamssf	@joanburton
@endakennytd	@michealmartintd	@gerryadamssf	

Table 18: Keywords for Ireland's set of tweets

Keywords include parties' names and their leaders' names. We will test five sets of tweets: Tweets that mention the partys' Twitter accounts (name starts with @), tweets that mention the leaders' accounts, tweets with text that includes a party's name in any form (that automatically also includes mentions @ and hashtags #), tweets with text that includes a leader's name in any form and tweets with text that includes any keyword related to a party.

Mentions Parties' Names

We test the set of tweets that mention parties' names:

	T_m	$\overline{T}_m$	Positive	Negative	$\overline{T}_s$	Actual Results
@FineGael	98	25,38	4	51	13,59	36,3
@fiannafailparty	60	15,54	3	31	16,76	34,6
@sinnfeinireland	68	17,61	7	30	40,43	19,7
@labour	160	41,45	15	89	29,205	9,4
Total sum	386	100	29	201	100	100
MSE		38,90			40,69	

Table 19: Mentions of parties' names and count, sentiment analysis metrics

We observe that the first metric's ($\overline{T}_m$) Mean Squared Error is smaller than those for sentiment analysis techniques. Sentiment analysis is not always reliable since we don't detect irony and sarcasm. Counting tweets with mentions to parties, the Labour party comes first with great

distance from its actual percentage but the other three parties appear to be better balanced. Fine Gael which actually took the first place in the elections, is second to mentions.

	Positive score	Negative score	$\overline{T}_o$	Actual Results
@FineGael	4	54	13,41	36,3
@fiannafailparty	3	33	16,46	34,6
@sinnfeinireland	7	31	40,90	19,7
@labour	15	93	29,21	9,4
Total sum	29	211	100	100
MSE			41,16	

Table 20: Mentions of parties' names and sentiment analysis with score metric

The difference from the network techniques that we tested is that Sinn Fein doesn't have the most mentions. Still, sentiment analysis results prove Sinn Fein to be the most popular.

Mentions Leaders'Names

	T_m	$\overline{T}_m$	Positive	Negative	$\overline{T}_s$	Actual Results
Fine Gael leader @EndaKennyTD	49	22,37	9	21	33,89	36,3
Fianna Fail leader @MichealMartinTD	34	15,52	3	17	13,95	34,6
Sinn Fein leader @GerryAdamsSF	84	38,35	15	32	37,07	19,7
Labour leader @joanburton	52	23,74	4	21	15,06	9,4
Total sum	219	100	31	91	100	100
MSE		33,34			27,67	

Table 21: Mentions of leaders' names and count, sentiment analysis metrics

Testing tweets that mention leaders' names also shows Sinn Fein to be first. It is interesting though that sentiment analysis shows that the leader of Fine Gael is popular. We need to point out that sometimes voters choose a leader and not a party, meaning that they vote for the leader they trust the most even if they have some ideological differences with the party. Sentiment analysis on these tweets leads to a MSE comparable to that of PageRank, and examining tweets that mention leaders' names proves more effective than those with parties' names. We assume that Twitter users discuss more specific events and talk about how they feel for politicians rather than for parties.

	Positive score	Negative score	$\overline{T}_o$	Actual Results
Fine Gael leader @EndaKennyTD	9	21	34,28	36,3
Fianna Fail leader @MichealMartinTD	3	17	14,12	34,6
Sinn Fein leader @GerryAdamsSF	15	33	36,36	19,7
Labour leader @joanburton	4	21	15,24	9,4
Total sum	31	92	100	100
MSE			27,11	

Table 22: Mentions of leaders' names and sentiment analysis with score metric

Using sentiment analysis with score doesn't change results that much since tweets rarely appear to be very positive or very negative.

Keywords with Parties' Names

	T_m	$\overline{T}_m$	Positive	Negative	$\overline{T}_s$	Actual Results
FG keywords	303	33,26	21	187	22,63	36,3
FF keywords	133	14,59	8	86	18,74	34,6
SF keywords	204	22,39	12	126	19,19	19,7
LP keywords	271	29,74	27	138	39,42	9,4
Total sum	911	100	68	537	100	100
MSE		28,81			36,60	

Table 23: Keywords for parties' names and count, sentiment analysis metrics

Counting tweets that contain text with at least one keyword that indicates a party's name, Fine Gael approximates its actual percentage and takes the first place in the metric's $\underline{T}_m$ results as in the Irish elections. However, the Labour party receives a high percentage that doesn't reflect the actual result.

	Positive score	Negative score	$\overline{T}_o$	Actual Results
FG keywords	22	193	23,62	36,3
FF keywords	8	90	18,41	34,6
SF keywords	12	130	19,12	19,7
LP keywords	27	144	38,85	9,4
Total sum	69	557	100	100
MSE			35,92	

Table 24: Keywords for parties' names and sentiment analysis with score metric

Sentiment analysis shows the Labour party to be popular even though results for the other parties are relatively good. The Mean Squared Error in this case is also high.

Keywords with Leaders' Names

	T_m	$\overline{T}_m$	Positive	Negative	$\overline{T}_s$	Actual Results
Fine Gael leader @EndaKennyTD	181	30,57	15	103	21,91	36,3
Fianna Fail leader @MichealMartinTD	62	10,47	4	29	20,76	34,6
Sinn Fein leader @GerryAdamsSF	213	35,97	26	118	33,16	19,7
Labour leader @joanburton	136	22,97	13	81	24,15	9,4
Total sum	592	100	58	331	100	100
MSE		32,62			28,23	

Table 25: Keywords for leaders' names and count, sentiment analysis metrics

In the experiment of testing tweets that contain keywords for parties' leaders, Sinn Fein is once again popular. Even though only a few Twitter users talk about Fianna Fail's leader, sentiment analysis shows that those tweets are positive.

It is also interesting that even though the experimental results don't rank leaders' popularities similarly to the election results, the scored sentiment analysis technique produces the smallest MSE of all algorithmic techniques we used to test data on the Irish General Election.

	Positive score	Negative score	$\overline{T}_o$	Actual Results
Fine Gael leader @EndaKennyTD	15	104	14,93	36,3
Fianna Fail leader @MichealMartinTD	13	29	46,4	34,6
Sinn Fein leader @GerryAdamsSF	26	122	22,06	19,7
Labour leader @joanburton	13	81	16,61	9,4
Total sum	67	336	100	100
MSE			25,56	

Table 26: Keywords for leaders' names and sentiment analysis with score metric

All Keywords (Parties' Names & Leaders' Names): Vote per Tweet

Using all keywords related to a party, either party names or leader names, results have lower MSE for the metrics $\underline{T}_m$ and $\underline{T}_o$. For sentiment analysis without using the sentiment score, the Labour Party takes the first place since it has less negative tweets than others, so the MSE increases.

	T_m	$\overline{T}_m$	Positive	Negative	$\overline{T}_s$	Actual Results
FG keywords	484	32,20	36	290	21,89	36,3
FF keywords	195	12,97	12	115	18,40	34,6
SF keywords	417	27,75	38	244	27,47	19,7
LP keywords	407	27,08	40	219	32,22	9,4
Total sum	1503	100	126	868	100	100
MSE		29,36			32,44	

Table 27: All keywords count and sentiment analysis

In general, it is more effective to search for keywords in the text of tweets than applying techniques that use tweets with mentions (@). In "vote per tweet" methods sentiment analysis with score proves better than simplistic sentiment analysis. However, we need more data and we cannot rely on those methods alone.

	Positive score	Negative score	$\overline{T}_o$	Actual Results
FG keywords	37	297	17,66	36,3
FF keywords	30	119	35,74	34,6
SF keywords	38	252	21,38	19,7
LP keywords	40	225	25,20	9,4
Total sum	145	893	100	100
MSE			24,51	

Table 28: All keywords sentiment analysis score

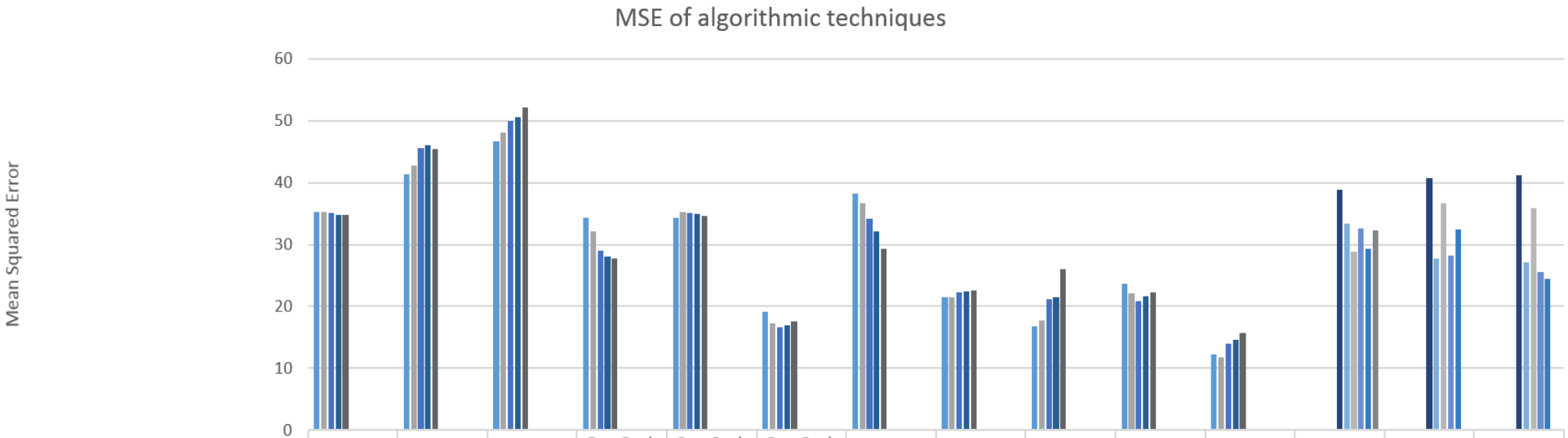
All Keywords (Parties' Names & Leaders' Names): Vote per User

Another approach is the "vote per user" method were we search for tweets with text that includes any keyword related to a party and for every user we find which is the party that the user mentions the most. We consider that the user will vote for this party. In this dataset, there were 756 users considered to be voters and the results show that Fine Gael and Sinn Fein are the parties that users discuss the most.

	$\overline{T}_m$	Actual Results
FG keywords	31,48	36,3
FF keywords	10,58	34,6
SF keywords	30,69	19,7
LP keywords	27,25	9,4
Total sum	100	100
MSE	32,24	

This technique doesn't prove better than "vote per tweet" but there is room for future work. The volume of data is an important factor and a dataset that includes more tweets per user could lead to a better analysis of the users behavior.

To review all techniques applied on the five networks and the set of tweets, Mean Squared Error is illustrated on the following graph, for every algorithmic technique:



	Followers' Percentage Vf	Absorbing Random Walks Va	ARW: Votes Vi	PageRank Vp (followers network)	PageRank Vp (complete network)	PageRank Vg (followers network)	HITS Vh (followers network)	HITS Vh (complete network)	HITS Vs (followers network)	HITS Vs (complete network)	HITS Vm (complete network)	Count Tm	Sentiment Ts	Sentiment score To
Network1	35,28126	41,41	46,75	34,35	34,32	19,17	38,28	21,45	16,75	23,66	12,16			
Network2	35,26365	42,74	48,05	32,05	35,23	17,21	36,72	21,5	17,78	22,13	11,76			
Network3	35,08675	45,51	49,94	28,91	35,14	16,66	34,14	22,2	21,11	20,79	13,96			
Network4	34,85163	46,07	50,64	27,98	34,87	16,91	32,09	22,37	21,47	21,56	14,53			
Network5	34,70598	45,39	52,22	27,69	34,64	17,54	29,3	22,59	26,06	22,29	15,62			
Mentions: vote per tweet (party name)												38,9	40,69	41,16
Mentions: vote per tweet (candidate)												33,34	27,67	27,11
Any text: vote per tweet (party name)												28,81	36,6	35,92
Any text: vote per tweet (candidate)												32,62	28,23	25,56
Any text: vote per tweet (any keyword)												29,36	32,44	24,51
Any text: vote per user (any keyword)												32,24		

Algorithmic Techniques

5.2 United Kingdom

5.2.1 Tweets

For the UK referendum we collected a dataset of tweets:

Tweets	839.756
Tweets with time zone “London”	263.599
Tweets with time zone “London” or location “United Kingdom”	264.930

Table 29: Statistics on tweets

We parsed 264.930 tweets that were posted probably from the United Kingdom. We used two sets of keywords, one for “Brexit” and one for “Bremain”. Those sets include the following words and hashtags:

Keywords for Brexit	Keywords for Bremain
#brexit	#bremain
#leave	#voteremain
#voteleave	#remain
#leaveeu	#remaineu
#takecontrol	bremain
brexit	stay
leave && referendum	remain && referendum
leaveeu	remaineu
voteleave	voteremain
takecontrol	

Table 30: Keywords for the UK's set of tweets

Sentiment analysis on the set of tweets we examined, shows that people have more negative feelings about “Brexit” while for “Bremain” the positive score is closer to the negative score:

	Brexit	Bremain
Count of tweets with keywords T_m	6194	2112
Positive tweets	608	311
Negative tweets	3550	1082
Positive score	627	324
Negative score	3702	1197

Table 31: Sentiment analysis results on the UK’s tweets

We observe that there are much more tweets that talk about “Brexit” than about “Bremain”. Brexit was actually the catchphrase of the referendum, most commonly used by the media. It is reasonable that most people tweet using that catchphrase.

Results on the data analysis of tweets that have text containing at least one of the previous keywords for the referendum follow:

	T_m (vote per tweet)	$\overline{T}_m$	MSE	$\overline{T}_s$	MSE	$\overline{T}_o$	MSE	Actual results
Brexit	6194	74,57	32,06	37,34	20,59	38,49	18,96	51,90
Bremain	2112	25,42		62,66		61,51		48,10

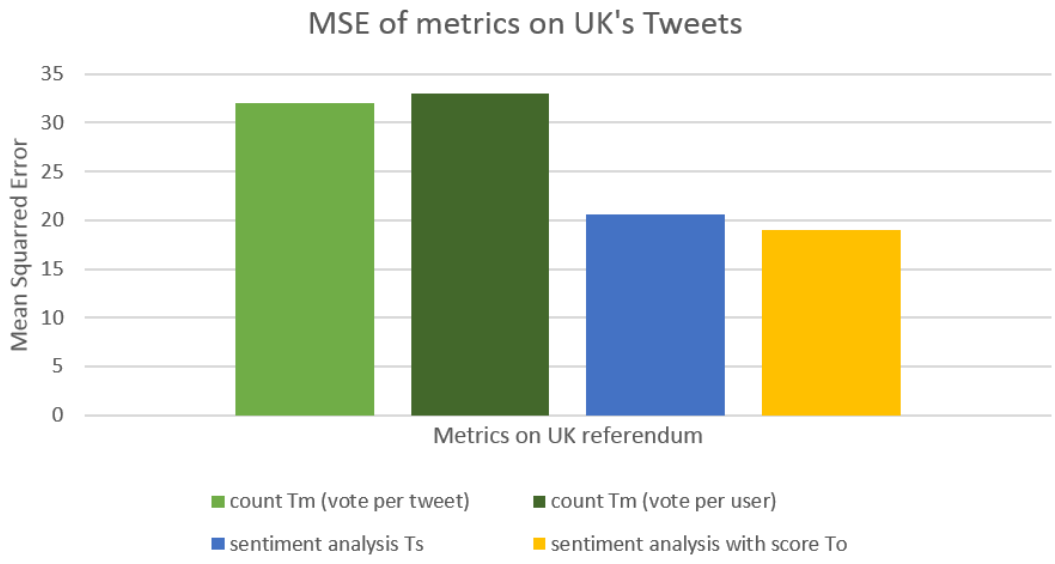
Table 32: Results of algorithmic techniques on the referendum

Bremain comes first according to the sentiment analysis metrics with smaller Mean Squared Error than those in the experiments on the Irish elections. In the actual referendum, Brexit received more votes. Counting tweets that contain keywords for Bremain or Brexit has a bigger MSE, even though it predicts the winner. The metric's $\overline{T}_m$ success in predicting the result of the referendum could be explained by the fact that Brexit was a common word, but we can assume that there is no bad publicity and as Brexit was the popular subject of discussion, the vote to UK leaving the EU, finally gained more supporters.

"Vote per user" approach is similar to "vote per tweet" in this case as Brexit is discussed by more users:

	T_m (vote per user)	$\overline{T}_m$	MSE
Brexit	3936	75,23	32,99
Bremain	1296	24,77	

Again, sentiment analysis with score is the technique with the lower MSE. In the following graph, we review all techniques applied on the test case of the UK referendum:



Graph 3: Evaluation of the referendum results

6 Conclusions

We collected data from Twitter for the Irish General Election 2016 and the UK EU membership referendum. Data included random tweets from a stream of tweets provided by the Twitter API and the network of followers of the four most popular parties of the Irish elections. We used the algorithms Absorbing Random Walks, PageRank and HITS on the network of users, and the percentages of the followers of every party to estimate the values of different metrics as predictors for the results of the Irish elections. For both test cases of Ireland and the UK, we counted tweets that contained specific keywords and also applied Sentiment Analysis to measure Twitter users' preferences.

Our techniques on these data didn't manage to predict results with accuracy but we acquired knowledge on the behavior of the Algorithms. HITS gave the best results and the Absorbing Random Walks performed poorly. PageRank performed well in general. Finally, even though sentiment analysis is not that accurate, also performed better than the Absorbing Random Walks algorithm.

We realize that when we examine a network, the most definitive factor is the number of followers a party has. For future work, we could try different filtering of the data to reduce bias. For example, the Sinn Fein party is possibly popular among younger people whose presence in the social media is stronger, so young people were overrepresented in our datasets and influenced our predictions. Also, groups of older voters are possibly underrepresented in social media. It is logical that older people choose incumbents and rely on familiar political faces. On the other hand, younger people are usually more open to change. By using demographics to de-bias the data we could expect better results.

It is possible that people are not actually satisfied with Fine Gael and Fianna Fail party, but the fact that they are the more powerful parties and carry a century of authority positions, makes people feel obliged to choose one over the other. Some people vote for a powerful party, in order to avert another from rising to power. This would mean that voters are not interested in their everyday lives to follow the party they voted for or comment on Twitter.

The Labour party's actual result is far from the predicted outcome. A possible explanation is that since it used to be much more popular, users haven't unfollowed the party's Twitter account, even though they didn't vote for LP.

Our prediction on Brexit was also unsuccessful when we applied Sentiment Analysis, but with a smaller Mean Squared Error. More thorough pre-processing of tweets could improve the content predictions. The volume of tweets talking about Brexit showed that Brexit will win but it is possible that this happened because "brexit" was the most popular catchphrase.

In conclusion, there is still much room for improvement in predictions based on the Social Media, but big steps have already been made by the scientific community and we can hope they will soon overcome today's problems. We can say that the problem of predicting election results with Twitter is very hard. It is necessary to collect enough data and reduce bias if possible.

7 References

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

[2] Dwi Prasetyo, N., & Hauff, C. (2015, August). Twitter-based Election Prediction in the Developing World. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media* (pp. 149-158). ACM.

[3] Mustafaraj, E., Finn, S., Whitlock, C., & Metaxas, P. T. (2011, October). Vocal minority versus silent majority: Discovering the opinions of the long tail. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on* (pp. 103-110). IEEE.

[4] Bollen, J., Mao, H., & Pepe, A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *ICWSM*, *11*, 450-453.

[5] O'Connor, B., Balasubramanyan, R., Routledge, B. R., & Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series.*ICWSM*, *11*(122-129), 1-2.

[6] Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *ICWSM*, *10*, 178-185.

[7] Jungherr, A., Jürgens, P., & Schoen, H. (2012). Why the pirate party won the german election of 2009 or the trouble with predictions: A response to tumasjan, a., sprenger, to, sander, pg, & welpe, im "predicting elections with twitter: What 140 characters reveal about political sentiment". *Social science computer review*, *30*(2), 229-234.

[8] Sang, E. T. K., & Bos, J. (2012, April). Predicting the 2011 dutch senate election results with twitter. In *Proceedings of the Workshop on Semantic Analysis in Social Media* (pp. 53-60). Association for Computational Linguistics.

[9] http://www.sensei-conversation.eu/

[10] http://www.zdnet.com/article/can-the-eus-sensei-project-predict-brexit-by-data-mining-social-media-chatter/

Appendix

A. Network (Ireland)

Screen names and ids on Twitter for the parties in dataset "network"

Party Name	Screen name	Id
Anti Austerity Alliance	AAA_IRE	1910508002
An Chomhdháil Phobail | The People's Convention (CPPC)	cppc_ie	219684564
Direct Democracy Ireland	ddi	619318353
Fianna Fáil	fiannafailparty	19594736
Fine Gael	FineGael	19530527
Green Party	greenparty_ie	19652551
Irish Democratic Party	IDPirl	3691107977
Communist Party of Ireland	irelandcp	154143068
Labour Party	labour	6751502
The National Citizens Movement	ncmirl	3392077097
People Before Profit	pb4p	237487471
Renua Ireland	RENUAIreland	3076115325
Right2Change	Right2ChangeIrl	3502590143
Workers and Unemployed Action	SeamusHealyTD	392883494
Sinn Féin	sinnfeinireland	22628924
Social Democrats	SocDems	3092798375
Workers' Party of Ireland	workersparty	5956402

B. Tweets (Ireland)

These data follow the json format and include information such as text, user's screen name and id, tweet's id, user's time zone, etc:

{"**screenname**": ""(user's unique name), "**user_id**": ""(user's unique id), "**tweet_id**": ""(tweet's unique id), "**is_rt**": (1 if it is a retweet or 0 if it's not), "**text**": ""(tweet's text), "**lang**": "en" for english or "und" if no language was detected, "**quoted_id**": "" (-1 if it doesn't quote another tweet), "**quoted_text**": ""(text of tweet quoted), "**mentions_length**": (number), "**mentions**": [{"**m_id**": ""(id of user mentioned, "**m_screenname**": ""(screen name of user mentioned in text using @ before screen name)}], "**hashtags_length**": (number), "**hashtags**": [{"**h_text**": ""(text following a hashtag #)}], "**media_length**": (number), "**media**": [{"**media_text**": ""(links in tweet's text)}], "**user_location**": ""(location described by user with no restrictions), "**geolocation**": {"**latitude**": "", "**longitude**": ""}(user's coordinates if available), "**date**": (date as a string, e.g."Thu Feb 04 05:15:42 EET 2016"), "**time_zone**": "" (city to define user's time zone), "**utc**": "" (number in seconds showing time difference from coordinated universal time or -1 for no utc offset available)}

C. Code

1. Collect Tweets(example from collecting tweets from twitter's sample stream possibly coming from the UK)
   Data with text content were collected using the twitter4j library to access the Public stream from Twitter's Streaming APIs. GET statuses/sample is a part of the Public stream that returns a small random sample of all public statuses flowing through Twitter.

//necessary libraries

```
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import twitter4j.GeoLocation;
import twitter4j.StallWarning;
import twitter4j.Status;
import twitter4j.StatusDeletionNotice;
import twitter4j.StatusListener;
import twitter4j.TwitterException;
import twitter4j.TwitterObjectFactory;
import twitter4j.TwitterStream;
import twitter4j.TwitterStreamFactory;
import twitter4j.User;
import twitter4j.conf.ConfigurationBuilder;
```

//connect to Twitter

In order to make authorized calls to Twitter's APIs, your application must first obtain an OAuth access token on behalf of a Twitter user. (https://dev.twitter.com/oauth/overview)

```
ConfigurationBuilder cb = new ConfigurationBuilder();
cb.setDebugEnabled(true);
cb.setOAuthConsumerKey("XXX");
cb.setOAuthConsumerSecret("XXX");
cb.setOAuthAccessToken("XXX ");
cb.setOAuthAccessTokenSecret("XXX");
TwitterStream twitterStream = new TwitterStreamFactory(cb.build()).getInstance();
```

//use a listener on Twitter's sample stream

```
twitterStream.addListener(listener);
twitterStream.sample();
```

//create a status listener and collect tweets from the UK

```
StatusListener listener = new StatusListener() {
public PrintWriter pt  = new PrintWriter("Tweets//t"+count+".txt");
public PrintWriter pu  = new PrintWriter("Users//u"+count+".txt");
public void onStatus(Status status) {
	User user = status.getUser();
	String timeZone = user.getTimeZone();
	GeoLocation geo = status.getGeoLocation();
	int utcOffset = user.getUtcOffset();
	String username = status.getUser().getScreenName();
	String jsonStatus = TwitterObjectFactory.getRawJSON(status);
	int tzExists = 0;
	if(timeZone!=null){
		if(timeZone.compareToIgnoreCase("london")==0)
			tzExists = 1;
	}
	int keepThis;
	if(geo!=null){
		double lat = geo.getLatitude();
		double lon = geo.getLongitude();
		if((lat>=49.8 && lat<=61.3) && (lon>=-8.8 && lon<=2.1)){
			keepThis = 1;
		}else{
			keepThis = 0;
		}
	}else{
		keepThis = -1; //no geolocation available
	}
	if(keepThis!=0 && (keepThis==1 || ((utcOffset==3600 || tzExists==1) &&
	(status.getLang().compareToIgnoreCase("en")==0 ||
	status.getLang().compareToIgnoreCase("und")==0)))){
		//write to file
		pu.println(user.toString());
		pt.println(jsonStatus);
	}
	File checkSize =new File("Tweets//t"+count+".txt");
	double megabytes = checkSize.length()/1048576;
	if(megabytes >= 100){
		System.out.println("\t\tFile network"+count+".txt too big.");
		pt.close();
		pu.close();
		count++;
		pt  = new PrintWriter("Tweets//t"+count+".txt");
		pu = new PrintWriter("Users//u"+count+".txt");
	}
} //end of onStatus
public void onDeletionNotice(StatusDeletionNotice statusDeletionNotice) {
	System.out.println("Got a status deletion notice id:" +
	statusDeletionNotice.getStatusId());
}
public void onTrackLimitationNotice(int numberOfLimitedStatuses) {
System.out.println("Got track limitation notice:" + numberOfLimitedStatuses);
}
public void onScrubGeo(long userId, long upToStatusId) {
	System.out.println("Got scrub_geo event userId:" + userId + " upToStatusId:"
	+ upToStatusId);
}
public void onException(Exception ex) {
	ex.printStackTrace();
}
@Override
public void onStallWarning(StallWarning stallWarning) {
	System.out.println("Got Stall Warning:" + stallWarning);
	}
}; //end of StatusListener
```

2. Parse Tweets and extract sentiment (useful examples)

//necessary libraries

```
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;
import edu.stanford.nlp.ling.CoreAnnotations;
import edu.stanford.nlp.neural.rnn.RNNCoreAnnotations;
import edu.stanford.nlp.pipeline.Annotation;
import edu.stanford.nlp.pipeline.StanfordCoreNLP;
import edu.stanford.nlp.sentiment.SentimentCoreAnnotations;
import edu.stanford.nlp.trees.Tree;
import edu.stanford.nlp.util.CoreMap;
```

//parse json example

```
JSONParser pTweet = new JSONParser();
JSONArray aTweet = (JSONArray) pTweet.parse(new FileReader(currentFile));
for(int i=1; i<aTweet.size(); i++){
       JSONObject oTweet = (JSONObject) aTweet.get(i);
       String text = (String)(oTweet.get("text"));
       JSONArray mentions = (JSONArray) oTweet.get("mentions");
       JSONArray hashtags = (JSONArray) oTweet.get("hashtags");
       String userlocation = (String)(oTweet.get("user_location"));
       String timezone = (String)(oTweet.get("time_zone"));
              ...
```

//locate useful tweets and apply sentiment analysis example

```
if(userlocation.toLowerCase().contains("ireland") ||
timezone.equalsIgnoreCase("dublin")){
       if(tLowercase.contains("fine gael") || tLowercase.contains("finegael")){
              String saText = cleanText(oTweet);
              int score = findSentiment(saText);
       }
  for(int j=1; j<mentions.size(); j++){
       JSONObject m = (JSONObject) mentions.get(j);
       String mscreenname =(String)(m.get("m_screenname"));
       String screen = mscreenname.toLowerCase();
       if(screen.compareTo("finegael")==0){
              String saText = cleanText(oTweet);
              int score = findSentiment(saText);
              if(score>2){ //positive
              ...
```

//sentiment analysis

```
private static int findSentiment(String tweet) {
Properties props = new Properties();
props.setProperty("annotators", "tokenize, ssplit, parse, sentiment");
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
    int mainSentiment = 0;
    if (tweet != null && tweet.length() > 0) {
        int longest = 0;
        Annotation annotation = pipeline.process(tweet);
        for (CoreMap sentence : annotation
                    .get(CoreAnnotations.SentencesAnnotation.class)) {
            Tree tree = sentence
                    .get(SentimentCoreAnnotations.SentimentAnnotatedTree.class);
            int sentiment = RNNCoreAnnotations.getPredictedClass(tree);
            String partText = sentence.toString();
            if (partText.length() > longest) {
                mainSentiment = sentiment;
                longest = partText.length();
            }
        }
    }
    return mainSentiment;
```

```
}
```

//clean tweet example

```
private static String cleanText(JSONObject tweet){
	String clean=(String) tweet.get("text");
clean = clean.toLowerCase();

if(((Long)tweet.get("is_rt"))==1){ //is a retweet
String regex = "\\s*\\brt\\b\\s*";
clean = clean.replaceAll(regex, "");
}
JSONArray media = (JSONArray) tweet.get("media");
for(int l=0; l<media.size(); l++){
	JSONObject med = (JSONObject) media.get(l);
	String mediatext = ((String)(med.get("media_text"))).toLowerCase();
clean = clean.replaceAll(mediatext, "");
}
JSONArray mentions = (JSONArray) tweet.get("mentions");
for(int j=0; j<mentions.size(); j++){
	JSONObject m = (JSONObject) mentions.get(j);
	String mscreenname =(String)(m.get("m_screenname"));
String screen = mscreenname.toLowerCase();
String regex = "@"+screen;
clean = clean.replaceAll(regex, "");
}
JSONArray hashtags = (JSONArray) tweet.get("hashtags");
for(int k=0; k<hashtags.size(); k++){
	JSONObject h = (JSONObject) hashtags.get(k);
	String htext = ((String)(h.get("h_text"))).toLowerCase();
if(htext.compareTo("ge16")==0){
clean = clean.replaceAll("#ge16", "");
}
clean = clean.replace(htext, "");
}
String[] parts = clean.split(" ");
String removeUrl="";
for(int p=0; p<parts.length; p++){
	if(!parts[p].startsWith("http")){
removeUrl=removeUrl+" "+parts[p];
	}
}
clean=removeUrl;
clean = clean.replace('@', ' ');
clean = clean.replace('#', ' ');
clean = clean.replace(':', ' ');
clean = clean.replace('~', ' ');
clean = clean.replace('+', ' ');
clean = clean.replace('-', ' ');
clean = clean.replace('.', ' ');
clean = clean.replace(',', ' ');
clean = clean.replace('?', ' ');
clean = clean.replace('!', ' ');
clean = clean.replaceAll("&", " and ");
clean = clean.replaceAll("fine gael", "finegael");
clean = clean.trim().replaceAll(" +", " ");

	return clean;
}
```