Tweet Analytics for Political Position Estimation

Aoife Mitchell
Computational Linguistics Group
Trinity College Dublin
The University of Dublin
Ireland
Email: mitcheao@tcd.ie

Anna Esposito
Università degli Studi della Campania
"Luigi Vanvitelli"
Caserta, Italy
and
International Institute for
Advanced Scientific Studies (IIASS)
Vietri sul Mare, Italy
Email: iiass.annaesp@tin.it

Carl Vogel
Computational Linguistics Group
Trinity College Dublin
The University of Dublin
Ireland
Email: vogel@tcd.ie

***Abstract*—An increasingly popular method of predicting trends and forecasting voting outcomes is to create a prediction model based on alignment of publicly available social media content produced by voters with voting behaviours. This paper aims to analyse whether Twitter data extracted from local Dublin City Council members' Twitter accounts in comparison with corresponding Councillor and Motion data from the Council-Tracker.ie website can be interpreted and used to create a model to predict the voting outcome of local Dublin City Council votes to pass motions. The aim was to explore whether through utilising machine learning techniques along with natural language processing techniques, reliable and data driven predictions can be generated for policy-making proposals brought forward. The acquired experimental results suggest that the approach used was marginally adequate in supporting the proposed hypothesis, although some interesting results were derived. Of the models analysed the Decision Tree model produced the most accurate results with an accuracy score of 0.71 (baseline: 0.63). Analysis of the models and an ablation study showed that the features derived from tweet texts and motion texts along with overall properties of a Councillor's twitter account were the most powerful indicators. The behaviour of a tweet, such as its acquired number of favorites or retweets, were not indicative of the results in both the random forest model and decision tree model.**

I. Introduction

The rise of social media has lead to the use of social media as a means of sharing perspectives such as their personal and political values. This gives occasion for other users to respond through likes or comments. Politicians and government officials all over the world are using social media as a way to communicate with their audience, inform them on important issues and affirm their standpoint on current events. It is also an instructive way of interacting with their audience and receiving feedback, thus informing them of the general reaction of the public, which could potentially transpire to be influential on their future decision making. All of this data provides a wide range of opportunities for data scientists to analyse a vast wealth of user-distributed information. This paper attends to data retrieved from the Twitter accounts of Dublin City Council members, containing data such as extracted information from the text they have tweeted,quoted and retweeted and other attributes of the tweets such as the number favourites. It also involves data obtained from the CounciTracker.ie website [1]. The aim is to assess whether using this twitter data from each of the Dublin City Council members' twitter accounts, an effective predictive model could be built using popular machine learning algorithms and text preprocessing techniques to estimate councillor votes on motions.

This research is aligned with work in cognitive infocommunications [2], [3] that addresses linguistic and behavioural interaction [4]–[8] and professional and social media analytics [9], [10]. A wider literature addresses language and communication analysis for political science [11]–[17]. This paper is organised as follows: §II describes the corpus, annotations and methods of analysis; §III details feature engineering; §IV profiles relevant features; §V presents the results; §VI discusses findings and concludes.

II. Materials & Methods

Here we describe the methods the research design, including an overview of the corpus and the performed data analysis.

A. Data Acquisition

Data analyzed (text of council motions and councillor tweets) were harvested from online sources. The motion data was collected using the beautiful soup, requests, selenium and pandas packages in Python, and the twitter data was collected using the rtweet and dplyr packages in R.

A website called CouncilTracker.ie [1], contains data on each Dublin councillor and the motions they voted on. The CouncilTracker.ie website contains data retrieved from Dublin City Council's monthly council meetings derived from printed copies of the minutes of these meetings each month. Developers of the website then input by hand the relevant information concerning each of the councillors, the motions that were brought forward and the dates they were proposed. Each of the original sheets received from Dublin City Council contained vague information on the motions such as 'Motion No.4' therefore, inspection of the meeting minutes was required to explain within a short statement the subject of the vote, which according to the CouncilTracker was "often a challenge".

Once the records for the motions were sourced, then we located each of the Dublin Council members' accounts on twitter and recording their associated Twitter usernames. Some council members were not on Twitter, and one was suspended for violating the Twitter rules: these members were excluded

from the twitter experiments. Twitter data was facilitated through setting up an twitter developer account to access the Twitter API and then using the rtweet package extract the twitter data of each council member into a CSV file.

For motion data, five main features were scraped from the CouncilTracker.ie website: motion text, motion date, councillor's name, party affiliation, and vote. The resulting data frame contained 5,026 observations concerning 77 councillors and 145 motions. Each row relates one councillor and one of the motions they voted in. Every motion the councillor voted in was recorded in a new row. There were 5 columns: Councillor, motion_text, motionDate, partyAffiliation and forAgainst.

There are a number of useful existing tools in scraping data from twitter such as the Tweepy and Twint packages in python and the rtweet package in R. We collected the twitter data of each of the Councillors from Twitter's REST and stream API via the get_timeline function in rtweet [18]. This function took an input vector of Councillor usernames and returned a table data frame of up to 3,200 statuses posted per Councillor, where each observation corresponded to a different tweet. The Twitter API limits the number of statuses retrieved to the most recent 3,200 statuses tweeted quoted or retweeted by each of the councillors although this transpired to be a minor limitation as only three of the councillors exceeded over 3,200 tweets. The resulting data frame contained 151,159 tweets from 73 councillors with 90 columns of twitter data.

B. Data Alignment

To build a model of how a councillor will vote in a future motion based on their twitter data we format the data in such a way that for each motion and councillor we have aligned all of the tweets that the councillor tweeted ***before*** the date of that motion. Each motion with each tweet before the motion row by row in a CSV file. The desired format of the data-set was for each councillor and particular motion to be aligned with all of the tweets that councillor has tweeted up until the day of voting in that particular motion.

Once both the motion and tweet dates and names were in a comparable format, the data was ready to be aligned. One data-frame contained the motion data for each of the councillors and another comprised the tweets from each of the councillors. For every row in the tweet data we compared the councillor name and tweet date with the motion dates and councillor names in the motion data and if the names matched and the tweet date was before the motion date, we appended the rest of the corresponding data for this tweet along the motionText, motionDate, for Against. In other words, we compared the twitter screen_name and motion data councillor name and if it was they corresponded with each other, we then compared the motion date and tweet date and if the tweet date was before the motion date, then the twitter data for this tweet became a row. For example, if a councillor had tweeted 10 times before voting in a particular motion, there would be 10 rows of data, with one row for each of the 10 tweets (and its corresponding favourite count, retweet count etc), each row would also contain the additional features councillor name, motionText, motionDate, forAgainst value for this particular motion.

In total, the fully aligned data-set had 2,591,406 rows of data after aligning every tweet that a councillor has tweeted before voting on a motion, for every motion that they voted on. Henceforth, we refer to two data-sets, the new fully aligned twitter data-set and the original motion data-set. The next section will outline some approaches to further preparing the data for analysis through dealing with missing or incomplete records, removing irrelevant data and feature engineering.

We applied feature engineering techniques to extract the most important data from the motion text, tweet text and quoted texts to find patterns and reduce the dimensions of the features used to train the models. Once the important information was extracted from the each of the texts we dropped the source-text columns.

C. Relevant data

Using the get_timeline function in the rtweet package in R to retrieve the twitter data of council members yielded 100 columns of data containing an abundance of noisy, trivial information from their twitter accounts. Columns such as country_code, geo_coords, account_language etc, were amongst a large amount information in the data demonstrating little variation or relevance to the task at hand. We removed 49 columns of this unwanted data from the data-set.

III. Overview of feature engineering

In the following paragraphs we will the feature engineering techniques used and how they were implemented.

A. Similarity of texts

We calculate the Jaccard Index (1) score between two texts.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{1}$$

For each observation in the full twitter data-set we calculated the Jaccard Index between the motion text and tweet text as well as for the motion text and quote text, treating all texts as lower-case. We created two new columns, one consisting of the Jaccard Index between the tweet and motion called 'Similarity' and the other consisting of the similarity score between the quote and the motion called 'quotedSim'. We calculated the Jaccard similarity of open-class words between the motion text and tweet texts as well as the closed-class words. We tokenised the words of the texts and calculated the similarity of the open class words in tweets and open class words in motion. We used the same method for the closed-class words in quoted text and closed-class words in the motion text. The same method applied to the quoted and motion text.

B. Word negation

A comparison of the negated words in the text could be useful making predictions. When looking at the negated items per total item in the text and negated items per total items in the motions if there's high textual overlap and there's negation in the text and not in the motion there's clear correlation between the text and motion. Thus, if the number of negated words in a text is much higher in the tweet or quoted tweet than in the motion then it might be easier to infer that the councillor is against said idea. We created a list of negations in English such as "not", "n't" , "no", "nobody", "none", "never",

"rarely", "hardly", "scarcely", etc. We obtained a count of the negated words in the tweet text and quoted text and a count of the negated words in the motion text by creating an algorithm using textblob and tokenised the words and obtained the number of negations in the texts and added these negated counts to the data-set as three new features, namely 'motionNegCount', 'tweetNegCount' and 'quoteNegCount'

C. Sentiment Polarity

The sentiment polarity of the texts using the VADER package in python, which has specially attuned atrributes to better understand social media texts, this package classified each the text in each row by first tonkenising the text and counting the number of positive negative and neutral words in the text , then using this overall count to classify the text as as either "positive", "negative" or "netural" [19]. This experiment was carried out on the motion text, tweet text and quote text and three additional columns called 'motionSentiment' ,'tweetSentiment' and 'quoteSentiment' were created.

D. Data reduction

In order to be able to train the model the dimensions of the data and number of features needed to be reduced. The hashtags column in the data set had to be excluded as the encoding of the hashtags created too many extra features. In addition, In order to make the data-set viable within the limited computational resources, the rows where both the tweet and quote text had a Jaccard similarity score of zero with the motion text were deleted. This still left 1,550,721 rows of data. Columns from the motion data, such as the party affiliation, gender were deleted as well as the date columns for the motion, quote and tweet date. We had origionally aligned all of the features from the motion data-set such as party-affilliation and gender to the full twitter data-set but they were unable to be included in the analysis. In addition, in order to make the data-set viable within the limited computational resources, the rows where the tweet or quoted text and motion text had a Jaccard Similarity score of zero were deleted. This left 1,550,721 rows of data. Table I shows the features used in training the machine learning models and a short description their meaning .

E. Baseline model feature engineering

Some features were engineered from the collected motion data in order to extract features and improve the performance of the baseline model. Firstly, usually the motions began with "To" and was followed by the verb that indicated the main action of the motion for example, "To rezone a site on Seville Place, Dublin 1". To see if these verbs could be used to indicate how a councillor may vote we extracted them from the motion statement and formed and new column in the motion data-set. We also formed new columns such as mostCommonVote which was a column containing each of the councillor's most common votes in past motions. If a councillor votes 'For' most motion this could indicated how they might vote in the next. We also created a lastVote column where for each motion a councillor voted in, it contained how they voted in the last motion so that if the councillor had entered a time where they had become more agreeable, this may indicate how they would vote next. Another column contained the gender of each of the councillors to enable study of gender interactions.

Column	Dtype	Description
Councillors	object	Councillor's full name
Similarity	float64	Jaccard Index between the motion and tweet
screen_name	object	Councillor's name on twitter
is_quote	bool	True/False if the tweet is a quote
is_retweet	bool	True/False if the tweet is a retweet
favorite_count	int64	Count of likes the tweet received
retweet_count	int64	Count of retweets the tweet received
hashtags	object	The contents of hashtags in the tweet
quoted_text	object	The text of the tweet that was quoted
quoted_favorite_count	float64	No. of likes on the quoted tweet
quoted_retweet_count	float64	No. of retweets on the quoted tweet
quoted_screen_name	object	Username of the quoted user
quoted_name	object	Name of the quoted user
quoted_followers_count	float64	Quoted user's no. of followers
quoted_friends_count	float64	Quoted users's no. of friends
quoted_statuses_count	float64	No. of tweets quoted user tweeted
quoted_verified	bool	True/False if quoted user is verified
retweet_favorite_count	float64	No. of total likes of the retweet
retweet_retweet_count	float64	No. of times the retweet has been retweeted
retweet_screen_name	object	Username of original author of the retweet
retweet_name	object	Name of who originally tweeted the retweet
retweet_followers_count	float64	No. of followers of original tweeter
retweet_friends_count	float64	Friends Count of author of retweet
retweet_statuses_count	float64	Tweet count of author of retweet
retweet_verified	bool	Author of retweet verified or not
followers_count	int64	Follower count of Councillor on twitter
friends_count	int64	Friend count of Councillor on twitter
listed_count	int64	No. of lists Councillor subscribes to
statuses_count	int64	No. of tweets Councillor has tweeted
favourites_count	int64	No. of posts Councillor has liked
verified	bool	Councillor verified or not on twitter
forAgainst	int64	Councillor's vote in the motion
OpenClassSim	float64	Jaccard Index of open class words in motion & tweet
ClosedClassSim	float64	Jaccard Index of closed class words in motion & tweet
QuoteOpenClassSim	float64	Jaccard Index of open class words in motion & quote
QuoteClosedClassSim	float64	Jaccard Index of closed class words in motion & quote
quoteSimilarity	float64	Jaccard Index of motion and quote
motionSentiment	object	Sentiment polarity of the motion
tweetSentiment	object	Sentiment polarity of the tweet
quoteSentiment	object	Sentiment polarity of the quote
motionNegCount	int64	Count of negated words in the motion
tweetNegCount	int64	Count of negated words in the tweet
quoteNegCount	int64	Count of negated words in the quote

TABLE I: Features in Full Twitter Data

F. Approach to machine learning models

As this was a multi-class classification problem where 'For','Against' and 'Abstain' are the three classes of outcomes. Due to the large dataset size we carried out initial experiments on a random subset of the data. We used the following three models: Naive Bayes classifier, Decision tree classifier and a random forest classifier. We completed some cross validation on each of the machine learning algorithms in order to tune the optimal hyperparameters. Steps were taken to reduce computational cost of tuning. For example, for the random forest classifier and decision tree classifier, 'gini' was chosen as the criterion for splitting the nodes of the decision trees as opposed to 'entropy', since studies [20] demonstrate that as a result of the use of logarithms in the 'entropy' criterion, the 'gini' criterion is faster and less computationally expensive while yielding similar results.

IV. Statistical Analysis and Ablation Study

The purpose of a statistical analysis for this data is to quantify how important a feature is. A partial ablation study was carried out: we trained the models with all of the features included in contrast to one of the features excluded, for each feature in turn, in order to get a better understanding of the feature's impact. After completing the ablation study we

implemented statistical analysis tests to explore whether the difference in accuracy between the features excluded and all of the features is significantly different. We also significance-tested accuracy between the machine learning models.

The test for normality was carried out on a table of sample true positive and true negative, false positive and false negative counts from each classifiers. The rows of the resulting table contained scores when the classifier was trained on all of the features vs when one feature was excluded from the features in order to understand the contribution of that feature to the data-set and the significance of the results when it is excluded. The features were divided into feature groups, namely 'authorproperties' which were all the features associated with the councillor and tweet authors, 'tweetText' which were the all features derived from the tweet texts and motion texts and finally 'tweetBehaviour' which invokes tweet interactive properties such as the number of likes and retweets a tweet elicited. Pearsons chi-squared test is used to test the likeliness that the results obtained were due to chance [21]. We used a form of chi-quared test called McNemar-Bower's chi-squared test which is a test for marginal homogeneity that analyses the symmetry of rows and columns in a two-dimensional contingency table to analyses the accuracy between the three classifiers. We then carried out Wilcoxon's non-parametric test for significance in the results by comparing the true postive, true negative, false positive and false negative counts for each of the analysed features. This test is a ranked test that is used to test for significance on repeated measurements by examining the mean ranks of their population [22].

V. Results

A. Text-data subset experiments

Initially, small experiments were carried out on just the motion and tweet text using a subset of the data. We trained a SVM, Decision Tree Classifier and multinomial Naive bayes classifier to analyse on the tweet text data using cross validation to select the hyperparameters. We then ran a number of experiments such as lementisation and stemming of the text using the Porter stemmer and wordNetLemmatizer from NLTK in Python. We completed cross validation for each of the hyperparameters for each of the models and selected the best hyperparameter with the highest accuracy and lowest error so as to not over fit or under fit the model by generating errorbar plots and estimating the best hyperparameter value.

Results of these tests on a small subset showed that stemming, lemmentisation and removal of stopwords has little effect on the models. Support Vector machine classifier with only the preprocessed tweet and motion text, a random subset of 1,000 rows at converted to dense sparse matrices and concatinated obtained an average accurracy of 0.72. The Removal or non removal of stop words showed no change in accuracy using the SVM model. Both achieved an average accuracy score of 0.71. A similar assesment was made for the lemmentisation and stemming of the text data, the average accuracy score remained at 0.71 and 0.72 respectfully.

B. Baseline Model results

A baseline model used only features extracted from the motion data website.[1] Each of the features in the motion were trained individually on the baseline model and all together using the Decision Tree Classifier, multinomial, Naive Bayes Classifier and SVM model. Table II below shows the accuracy results for each of the evaluated features when trained using Decision Trees. The most accurate results were achieved when all of the features were used where as the gender of the councillor was the poorest at predicting the voting outcome, only achieving results close to a majority-class baseline model.

Architecture	EvaluatedFeature	Accuracy
Decision Tree	All Features	0.68
Decision Tree	Councillor	0.64
Decision Tree	Verbs	0.65
Decision Tree	Party Affiliation	0.65
Decision Tree	Gender	0.63
Decision Tree	lastVote	0.64
Decision Tree	mostCommonVote	0.65
Baseline	most frequent	0.63

TABLE II: The accuracy scores of features when trained on a Decision Tree classifier

The following table III illustrates the accuracy results for each of the features when trained using a multinomial Naive Bayes Classifier.The accuracy results of this model were slightly better than the decision tree model and achieved the best results of the three models. A combination of all of the features and the verbs feature, although a modest score of 0.70, both achieved the highest accuracy out of each of the evaluated features. In contrast, both the Councillor and mostCommonVote features performed the worst with a score of 0.62, performing even worse than the model that always chose the most frequent output. These features were poor at providing insight into the councillors' votes.

Architecture	EvaluatedFeature	Accuracy
Naive Bayes	All Features	0.70
Naive Bayes	Councillor	0.62
Naive Bayes	Verbs	0.70
Naive Bayes	Party Affiliation	0.63
Naive Bayes	Gender	0.66
Naive Bayes	lastVote	0.63
Naive Bayes	mostCommonVote	0.62
Baseline	most frequent	0.63

TABLE III: The accuracy scores of features when trained on a Naive Bayes multinomial classifier

Finally, the features were also evaluated on the SVM model and the results very similar to the previous two models. The verb feature and all features evaluated performed the best with an accuracy score of 0.69 and 0.69 respectfully. This time the gender feature performed the worst with a score of 0.62, similarly to before just performing slightly worse than the rest of the remaining features. See Table IV.

Overall, the results of the trained model were underwhelming in vote prediction accuracy. The highest accuracy was generally achieved by incorporating all of the features.

[1] A random guess classifier for a three-label problem would have expected accuracy of 0.33; a majority classifier (**For**) has expected accuracty of 0.63.

Architecture	EvaluatedFeature	Accuracy
SVM	All Features	0.68
SVM	Councillor	0.63
SVM	Verbs	0.69
SVM	Party Affiliation	0.63
SVM	Gender	0.62
SVM	lastVote	0.63
SVM	mostCommonVote	0.65
Baseline	most frequent	0.63

TABLE IV: The accuracy scores of features when trained on a Support Vector machine model

C. Full twitter data-set model results

The results of all of the features trained on the each of the three models is shown in V. The decison tree model obtained the most accurate results with an accuracy score of 0.71 with Random forest coming in slightly behind with a a score of 0.68 and finally the multinomial naive bayes model performing the worst with very poor score 0.30. Overall, ability of the model to identify patterns in the twitter data was unsatisfactory. In comparison with the baseline motion data models these models were not much better at predicting the results with the highest accuracy that the motion data baseline model acheived being a score of 0.70 and for the twitter data models a score of 0.71.

Architecture	EvaluatedFeature	Accuracy
Decision Tree	All Features	0.71
Naive Bayes	All Features	0.30
Random Forest	All Features	0.68

TABLE V: Accuracy results of full twitter data model analysis

D. Statistical evaluation and partial ablation results

Some statistical experiments sought better understanding of the results and the behavior of the machine learning algorithms used. The test for normality was carried out on a table of true positive,true negative, false positive and false negative counts for each of classifiers. The rows of the resulting table contained results from when the classifier was trained on all of the features vs when one feature was excluded from the features in order to understand the contribution of that feature to the data-set and the significance of the results when it is excluded. As shown in the table VI, each of the p-values are less than the 0.05 therefore we reject the null hypothesis that the data in normally distributed, thus we can carry out non parametric statistical tests on the data.

Architecture	Evaluated	Value
Decision Tree	p-value	0.0000033885542052
Naive Bayes	p-value	0.00000000005197406
Random Forest	p-value	0.000000003924622404

TABLE VI: Shapiro Test for normality

A histogram of the voting outcomes in the twitter data-set is shown 1. Following this, McNemar-Bowker's test was carried out to deduce whether the predictions from the random forest model and decision tree model were statistically significant. The results calculated a p-value in scientific notation of 2.2e-16 and had 3 degrees of freedom. The evaluated p-value is

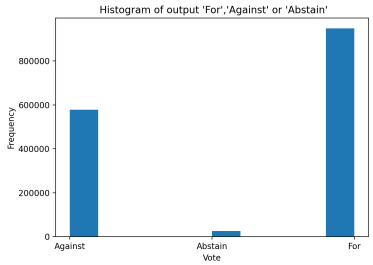

Fig. 1: A Histogram of the voting output of councillors in the Fully aligned twitter data-set fairly left skewed

less than 0.05 therefore we reject the null hypothesis that the contingency table of predictions is symmetric. The naive bayes classifier performed so poorly that the a comparison of the contingency tables couldn't be completed due to it missing one row as it was unable to predict any 'For' votes in the analysis. Furthermore, the results of the ablation study proved that the model that performed the best out of the three models analysed was the decision tree classifier. The difference in accuracy vs one feature excluded in is plotted in figure 2. From the results of decision tree ablation study in figure 2 we can see that the motion sentiment has the biggest positive impact on the accuracy of the model and the accuracy of the model improves the most out of all of the features when the motion sentiment is included rather than when it is not. In contrast the is_retweet feature negatively impacts the model the most when it is included. Overall with this model the similarity measures, and negated words count for each of the as well as the counts associated with the councillor's twitter account, such as how many tweets they have tweeted or friends they have on twitter are the features that have a positive impact on the decision tree's accuracy. In comparison to the properties of the quoted user and the retweeted user such as quoted_statuses_count which had a negative impact on the model's accuracy.

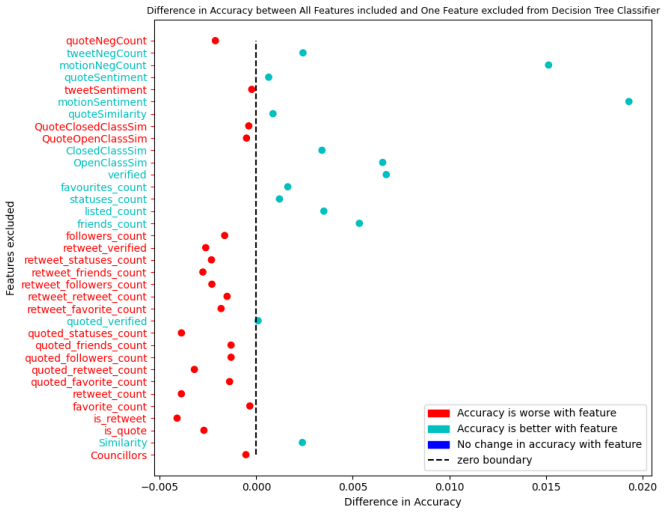

Fig. 2

The naive bayes classifier performed the worst of the three

models analysed and could not make accurate predictions from the features analysed. The difference in accuracy vs one feature excluded results of the ablation study for the naive bayes classifier is plotted in figure 3. Most of the features showed no change in accuracy of the model whether they were included or not. Some count features such as favourites_count positively impacted the accuracy of the model while others such as quotes_followers_count had a negative impact. This model was unable to detect the patterns in the data-set and thus could often only predict the 'For' votes in the data which were the most common votes in the data-set. Finally, results

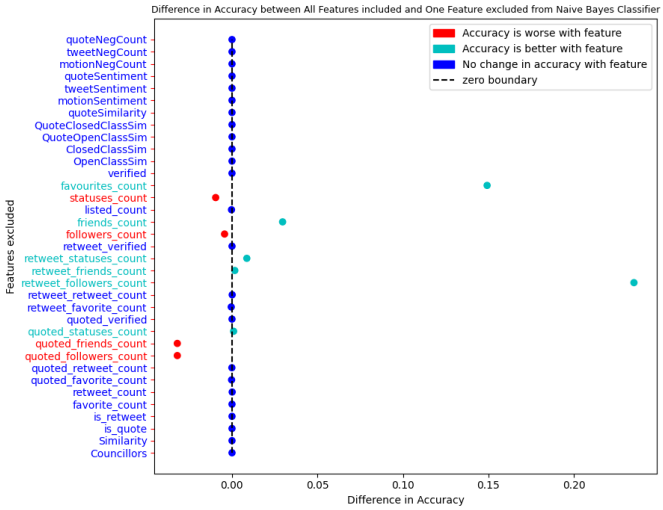

Fig. 3

of the ablation study for the random forest classifier for the difference in accuracy of all features vs one feature excluded is plotted in figure 4. This shows a different indication of feature importance in the data. The sentiment of the motion, negation in the motion and the similarity between the open class words and the closed class words between the councillor's tweet and the motion text were the only features where the model's accuracy improved when they were included as opposed to them being excluded. The tweet sentiment and number of favorites on the tweet negative impacted the accuracy of the model the most when they were included.

A Wilcoxon non-parametric t-test was implemented on the feature property groups, the features that were properties of the councillor and their twitter accounts VII shows a table of the results of the Wilcoxon non-parametric t-test to test whether the features are statistically significant. With a chosen alpha value of 0.05 the decision tree model shows a significant difference in the accuracy of the model when the tweetBehaviour features are excluded vs when all features are included as the evaluated p-value is 0.00000610. Thus, the impact of the tweetBehaviour on the model was negatively significant. The tweetText properties were shown to be positive in the boxplot in figure 5 and table VII shows the p-value of tweetText properties was less than 0.05 therefore the impact of the tweetText features in the data-set were deduced as being positively significant. In contrast the inclusion of the author-property group of features was not significant on the accuracy results as the p-value of the t-test was 0.15 which is greater

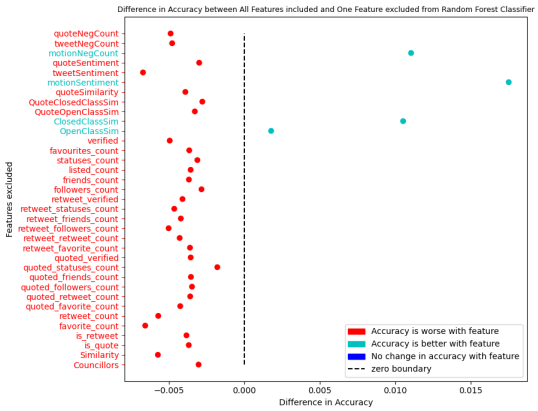

Fig. 4: The difference in accuracy vs one feature excluded

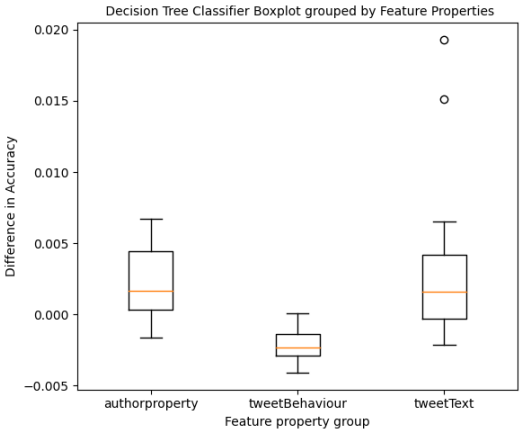

Fig. 5: Boxplot of grouped properties of the full twitter data-set

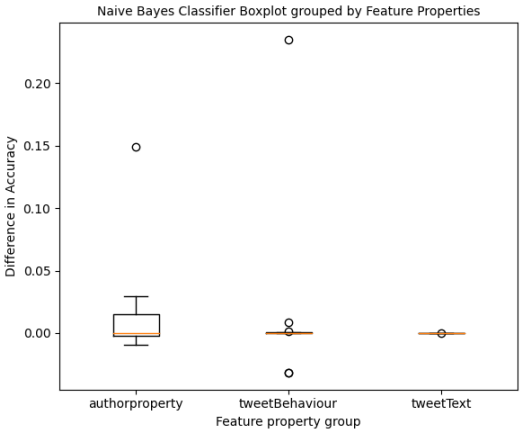

Fig. 6: Boxplot of grouped properties of the full twitter data-set

than 0.05. On ther other hand, The only feature property group that made a significant difference in accuracy in the naive bayes model was the tweetText features with a p-value of

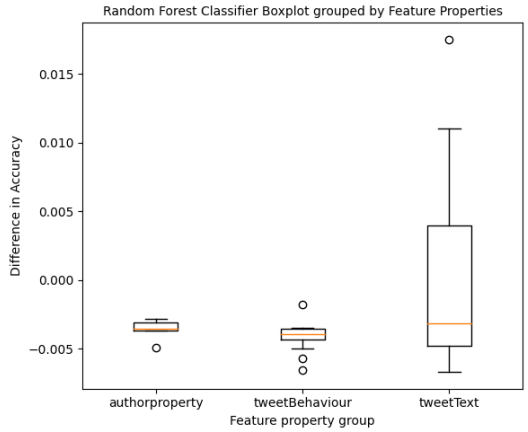

Fig. 7: Boxplot of grouped properties of the full twitter data-set

0.00048828125. As for the random forest model, as shown in figure 7 the authorproperty and tweetBehaviour property groups had a negative impact overall on the accuracy of the naive bayes model whereas the tweetText feature group have a high range and some features having a positive impact on the accuracy whilst others, a negative impact. In table VII we can see that the p-values for feature groups authorproperty and tweetBehaviour were both below 0.05 therefore their presence in training the random forest model had a significant(negative) difference on the accuracy of the model. The results show that the tweetText group overall did not have a significant difference on the accuracy of the model.

Furthermore, upon inspection of figure 5 by comparing the medians of each of the groups represented by an orange coloured line in the box plot, one can see that both the authorproperty feature group and tweetText feature group have median accuracies that are different from the tweetBehaviour feature group as the median line of a box plot lies outside of the box of the tweetBehaviour box plot. The naive bayes box plots show little or no change in medians of each of the property groups but showed a difference in the interquartile range(length of the box) meaning the data is more dispersed than the other feature groups. The large whiskers at either end of the box indicates a wide range in the data which thus indicates a wider distribution of the authorproperty data in comparison to the two other feature groups.As for the random forest model, the medians of the authorproperty and tweetText show that the difference between tweetText and tweetBehaviour is likely to be significant, whereas the rest of the medians are very closely inside the other feature's boxes. The interquartile range of the tweetText feature group is highly scattered and dispersed.

E. Summary

This section evaluated and discussed the results of the analysis carried out on the data. The decision tree model predictions for all features were found to be significantly different from the random forest predictions and therefore most accurate, significantly, out of the three models. The ablation study and subsequent statistical analysis showed that the 'tweetText' property features had a significant positive impact

Architecture	FeaturePropertyGroup	w	p-value
Decision Tree	authorproperty	4.5	0.15625
Decision Tree	tweetBehaviour	1.0	0.00000610
Decision Tree	tweetText	2.0	0.0341796875
Naive Bayes	authorproperty	12.0	0.8125
Naive Bayes	tweetBehaviour	49.0	0.34838867
Naive Bayes	tweetText	0.0	0.00048828125
Random Forest	authorproperty	0.0	0.015625
Random Forest	tweetBehaviour	0.0	0.00000305
Random Forest	tweetText	34.0	0.73339844

TABLE VII: Wilcoxon non-parametric paired t-test consisting of the difference between all Features included and one group of Feature properties excluded

on the decision tree model's accuracy and the 'tweetBehaviour' property features had a significantly negative impact.

VI. Discussion & Conclusions

A. Limitations

Results are impacted by data pre-processing. For example, there are some limitations to the accuracy of automatic Parts-Of-Speech tagging. There are also some limitations to carrying out sentiment analysis such as the detection of sarcasm and irony in the text. The biggest limitation was the boundedness of computational resource whilst analysing high-dimentional data. To resolve this steps were taken to extract the most important and useful information from the data-set such as extracting the sentiment polarity of the texts, the count of negated words in the texts, the removal of redundant data and the removal of any texts that had a similarity of 0. Another point to note was that some councillors didn't have twitter accounts and therefore their data was unavailable.

B. Final remarks

We began outlining the aims and approaches to analysing whether twitter data could be used in order to predict how Dublin City Councillors would vote in motions brought forward at the Monthly Council Meetings. Then, how the data for the analysis was collected was discussed along with how it was aligned and formatted for analysis. Following this, the techniques involved in analysing the data were put forward and with some of the limitations of the analysis being tackled and alternative approaches utilised. Important information was extracted from the twitter data through data preprocessing and feature engineering techniques such as sentiment analysis and Jaccard similarity, whilst redundant information was removed to improve the accuracy of the model. We evaluated each of the models and discussed the results. The most accurate model was the decision tree model with an accuracy score of 0.68 and interestingly, the significantly important features tested using a form of Pearson's chi squared test [21] were the tweetText features, meaning the features and properties extracted from the texts in the data-set were had the biggest positive impact on predicting the outcome of the motions votes. Another interesting thing to note it that the inclusion of popularity of a tweet features such as the number of retweets and favourites of the tweet, negatively impacted the accuracy of both the random forest and decision tree models. One might have thought that the combination of the popularity of the tweet along with the similarity to a motion may be indicative

of a vote. However this, we believe, may be due to the fact that likes on Twitter are not necessarily considered as important as other social media platforms such as Instagram or Faceboook. Overall, with the highest accuracy achieved being just 0.71, we conclude that using the implemented methods and only coarse abstractions of twitter data text, a model that can make accurate predictions on how a councillor will vote in motions brought forward requires more finely grained abstractions.

REFERENCES

[1] D. Inquirer and CivicTech.ie, "CouncilTracker.ie," https://www.counciltracker.ie/, 2019.

[2] P. Baranyi and A. Csapo, "Cognitive infocommunications: Coginfocom," in *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2010, pp. 141–146.

[3] P. Baranyi, A. Csapo, and P. Varlaki, "An overview of research trends in CogInfoCom," in *IEEE 18th International Conference on Intelligent Engineering Systems*, ser. INES, 2014, pp. 181–186.

[4] A. Esposito, A. M. Esposito, and C. Vogel, "Needs and challenges in human computer interaction for processing social emotional information," *Pattern Recognition Letters*, vol. 66, pp. 41–51, 2015.

[5] C. Vogel and A. Esposito, "Interaction analysis and cognitive infocommunications," *Infocommunications Journal*, vol. 12, no. 1, pp. 2–9, 2020.

[6] M. Koutsombogera and C. Vogel, "Observing collaboration in small-group interaction," *Multimodal Technologies and Interaction*, vol. 3, no. 3, p. 45, 2019.

[7] M. d. Vázquez, R. Justo, A. L. Zorrilla, and M. I. Torres, "Can spontaneous emotions be detected from speech on TV political debates?" in *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2019, pp. 289–294.

[8] F. Haider, M. Koutsombogera, O. Conlan, C. Vogel, N. Campbell, and S. Luz, "An active data representation of videos for automatic scoring of oral presentation delivery skills and feedback generation," *Frontiers in Computer Science*, vol. 2, no. 1, pp. 1–13, 2020.

[9] C. Vogel, "Multimodal conformity of expression between blog names and content," in *4th IEEE International Conference on Cognitive Infocommunications*, P. Baranyi, A. Esposito, M. Niitsuma, and B. Sølvang, Eds., 2013, pp. 23–28.

[10] A. Karampela and C. Vogel, "Nouns and verbs in professional reporting of extreme events," in *Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications*, 2019, pp. 253–258.

[11] M. Gabel and J. D. Huber, "Putting parties in their place: Inferring party left-right ideological positions from party manifestos data," *American Journal of Political Science*, vol. 44, pp. 94–103, 2000.

[12] M. Laver and J. Garry, "Estimating policy positions from political texts," *American Journal of Political Science*, vol. 44, no. 3, pp. 619–634, 2000.

[13] M. Laver, Ed., *Estimating the Policy Position of Political Actors*. Routledge, 2001.

[14] M. Laver, "Position and salience in the policies of political actors," in *Estimating the Policy Position of Political Actors*, M. Laver, Ed. London: Routledge/ECPR Studies in European Political Science, 2001, pp. 66–75.

[15] S. Van Gijsel and C. Vogel, "Inducing a cline from corpora of political manifestos," in *Proceedings of the International Symposium on Information and Communication Technologies*, M. A. et al., Ed., 2003, pp. 304–310.

[16] M. Laver, K. Benoit, and J. Garry, "Extracting policy positions from political texts using words as data," *American Political Science Review*, vol. 97, 2003.

[17] J. DiGrazia, K. McKelvey, J. Bollen, and R. F., *More Tweets, More Votes: Social Media as a Quantitative Indicator of Political Behavior.* PLoS ONE 8(11): e79449. https://doi.org/10.1371/journal.pone.0079449, 2013.

[18] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/

[19] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text." *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, 2014.

[20] P. Aznar, "QuantDare," https://quantdare.com/decision-trees-gini-vs-entropy/#:~:text=The%20Gini%20Index%20and%20the,both%20of%20them%20are%20represented., 2020.

[21] K. Pearson, *On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.Magazine Series, 5, 50 (302): 157–175.*, 1900.

[22] D. Rey and M. Neuhäuser, *Wilcoxon-Signed-Rank Test*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1658–1659. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_616