university of groningen / faculty of arts

Predicting sociability automatically from Facebook data

Classifying sociability of Facebook users, based on textual features of their status updates and social network features based on their Facebook connections, with supervised machine learning methods

Xiaoying Chen

Bachelor thesis
Informatiekunde
Xiaoying Chen
s2714140
June 20, 2017

ABSTRACT

This experiment tried to predict sociability based on Facebook data with supervised machine learning methods. Sociability is defined in this research as a combination of extraversion and agreeableness. To improve the accuracy of sociability recognition, multiple linguistic features were extracted from Facebook status updates. Since the dataset used is small, 10-fold cross validation was applied during the classification to make sure all data get trained and tested. Taking punctuation, pronouns, and positive and negative words as features, a Support Vector Machine classifier yielded the best performance of classifying sociability based on status updates. Social network features, such as network size, are also taken into account. Although social network features are weakly associated with sociability, they still affect the performance of the sociability prediction. Shuffling the dataset improves the accuracy by 44.05%, which might be due to overfitting problem. Regardless of shuffling or not, the Support Vector Machine achieved the highest accuracy score after adding all effective social network features.

CONTENTS

PREFACE

I applied the knowledge I learned from previous year of studying Information Science, and learned new things when developing the program. During this period, I received a lot of help and support from my parents, my sister, class mates, and friends. I am extremely grateful for their support. I want to thank my supervisor dr. L.M. Bosveld-de Smet, who gives me a lot of useful advice and feedback.

1 INTRODUCTION

Facebook is one of the most popular social media platforms. People interact with friends on Facebook. Is it possible to measure how sociable people are on Facebook? To answer this question, the term 'sociability' must be defined. Sociability is described by Cheek and Buss (1981) as a willingness to interact with people. It describes how people interact with others in society. The personalities that also describe the interaction of people are extraversion and agreeableness. People with extraversion and agreeableness are usually sociable. In this study, I define sociability as the combination of extraversion and agreeableness.

Measuring people's personality can make decision making process effective. When recommender system recommends products to users, it can select products to users according to their personalities (Tkalcic et al., 2009). Another example of personality measurement application is job recruitment. When someone is applying for a job, it is useful for the human resource manager to know the applicant's personality. For example, Barrick and Mount (1991) mentioned that extroverts perform better in socially based jobs, such as, sales. But it is very time-consuming to annotate each person's personality manually. Labeling personalities with a machine learning method is more efficient.

In this research, I want to determine whether it is possible to predict sociability based on Facebook data using machine learning methods. Two types of data – textual and numerical – are used. Textual data refers to status updates. These status updates are user's thoughts that users shared on their Facebook pages. Only textual features from status updates are taken in this research. Numerical data are social network properties such as network size. Because the dataset used in this study is relatively small, this study can determine which machine learning methods perform better with a small dataset.

The main question of this research is: "Can sociability be automatically predicted from Facebook data using machine learning methods?" To answer this question, I split it up in two sub-questions. The first sub-question is: "Is it possible to predict sociability from Facebook status updates using linguistic features?" Based on the first question, the second question is: "Can social network properties improve the accuracy of sociability recognition from Facebook status updates?"

The first step in answering the research questions is to obtain background information about the relationship between personalities and features (both linguistic and social network features). The relevant literature is also discussed. When enough information was obtained, the dataset used for this experiment was collected from MyPersonality project (Celli et al., 2013) and preprocessed for the classification task. Since there are two sub-questions, the experiment was split into two sub-experiments. Both sub-experiments followed the same steps: feature selection, cross validation, classification and evaluation. After the data is classified into high and low sociability, the calculated accuracies are discussed. Using those scores, the research questions are answered.

2 BACKGROUND

The goal of this research is to classify people as low or high sociability from their Facebook status updates and social network properties with supervised machine learning methods. To select the effective linguistic and social network features for sociability prediction, the correlations between features and personality are reviewed.

2.1 PERSONALITY

One of the popular personality models is Norman's Big Five personality model. It categorizes people based on five personality traits, namely: extraversion, agreeableness, openness to experience, neuroticism, and conscientiousness. Openness to experience, neuroticism, and conscientiousness will not be discussed in this research, since they are not related to sociability. Extraversion refers to a person who is outgoing, sociable and energetic (McCrae and John, 1992). People with agreeableness are friendly and behave nicely to others. They are often trusted by others (Costa et al., 1991). The personalities can be measured in degree. When someone is low in extraversion, this person is usually labeled as an introvert (Eysenck, 1956).

As mentioned earlier, extraversion and agreeableness are merged into sociability in this research. When someone is high sociability, it means this person is sociable, friendly and outgoing. If a person is shy, prefers to be alone, unfriendly, and distrustful (McCrae and John, 1992). This person is categorized as low sociable.

2.2 LINGUISTIC FEATURES AND PERSONALITY

Personality can impact the language used when writing in daily life. Previous research found a correlation between linguistic cues and personalities. Gill and Oberlander (2002) studied the language use in text written by extroverts and introverts. They concluded that people with extraversion usually wrote longer sentences with more exclamation marks. Extroverts liked to refer to others, while self-referencing was frequently used by introverts. Sentences written by introverts were more concrete.

Pennebaker and King (1999) found that extraversion and agreeableness were positively correlated with positive emotion words. Agreeableness was also negatively correlated to negative emotion words. Yarkoni (2010) researched a large blog corpus and showed that extraversion was positively related to positive emotions and social processes. The study also found that agreeableness is negatively correlated to anger and swear words, but positively correlated to love words.

2.3 SOCIAL NETWORK AND PERSONALITY

Other studies analyzed the relationship between personality and social network use. Amichai-Hamburger and Vinitzky (2010) investigated the connection between the use of Facebook and personalities. They found that extroverts had more friends on Facebook than introverts, but there was no evidence to prove that people with agreeableness had more friends.

Betweenness centrality calculates how central a person stands in a social network when the person brings two or more people or groups in connection. It is calculated by counting the times a person falls on the shortest paths between all people in the network (Brandes, 2001).The study of Wehrli et al. (2008) concluded that extroverts had a central position in their social network, while there was no relation between centrality and agreeableness.

According to Friggeri et al. (2012), extroverts were more willing to act as bridges between groups, while introverts were more likely to stay in a largely isolated group. The person who links two groups or people in connection is called as bridge. Brokerage is a social network property which calculates the extent that someone is acting like a bridge in social network (Haythornthwaite, 1996).

Density is a social network measurement that calculates how many people in a person' social network interact with each (Haythornthwaite, 1996). Additionally, Golbeck et al. (2011) showed that extroverts often were part of multiple groups. Since members of one group are not likely to know the members of another, the social network of extroverts was less dense than introverts, who tend to stick with one group.

2.4 PERSONALITY PREDICTING WITH MACHINE LEARNING

Several studies predicted Big Five personalities from text. Iacobelli et al. (2011) collected writing from approximately 3000 bloggers to classify their personalities. They selected features that significantly related to each personality and performed with a Support Vector Machine (SVM) classifier. They achieved an accuracy of 71.68% on extraversion and 78.31% on agreeableness. Wright and Chin (2014) reported that Part of Speech n-grams can significantly improve the performance of personality prediction from text with SVM classifier. The experiment of Mairesse et al. (2007) was based on an essay corpus (Essay) and conversation corpus(EAR). They extracted linguistic features using LIWC and classified with multiple classifiers. The accuracy score was 56% for both extraversion and agreeableness.

Celli et al. (2013) organized a shared task of predicting personality from Facebook data and Essay corpus. They concluded that select features over a very large feature space were very effective to predict personality. Markovikj et al. (2013) used only the Facebook dataset and yielded f-score of 0.904 with ranking algorithms and pointed out that punctuation, adjective and verbs were indicative features of extraversion and agreeableness.. Alam et al. (2013) also participated in the shared task. They used a bag-of-words approach, tokenized the status sentences into tokens, and transformed tokens into a vector with the Tf-idf Vectorizer. They tried several classifiers (SVM, Bayesian Logistic Regression, and Multinomial Naive Bayes (MNB)) and achieved an average f-score of 0.58 with MNB.

3 DATA AND MATERIAL

To perform the research, a dataset of Facebook data with Big Five personality scores of users was prepared. Based on the given personality information, the dataset was annotated by high and low sociability. After categorizing the data into high and low sociability, necessary data from the dataset was preprocessed to make the classification task more efficient.

3.1 COLLECTION

The dataset used in this research comes from MyPersonality project Celli et al. (2013). It contains 9917 Facebook status updates from 250 users. The dataset contains user id, status updates, time of post, scores of each personality of the Big Five, a yes-/no- label for each personality, and network properties. Every user in the dataset took the IPIP personality questionnaire. Personality scores are on a scale from 1 to 5, and are calculated for each user based on their answers to the questionnaire. The yes-/no- label for personality are categorized with the median split from the calculated scores. The network properties include network size, betweenness centrality, normalized betweenness centrality, brokerage, normalized brokerage, density, and transitivity. The dataset is in a CSV file. An example of raw data is available in figure 1 of 'Appendices'.

3.2 MATERIAL

The experiment is programmed in Python with the relevant Python toolkit. The module 'csv' was used to read the columns of CVS file, as it easily extracts information from the dataset. The extracted data are stored in pickle files. The toolkit 'panda' was used to restructure the dataset, so that features could be easily extracted when necessary. 'NLTK.word_tokenize' were used to tokenize the sentences into tokens. The Afinn toolkit was used to calculate the positive and negative polarity of the status updates. The classification task was performed with 'scikit-learn'.

3.3 ANNOTATION

As mentioned earlier, the personalities of users are given in two forms: scores and a yes-/no- label. To annotate the dataset, these personality information is used. In the first stage, high sociable people are users who get yes-labels for both extraversion and agreeableness. The rest of the users are considered low sociability. But this did not seem to work. When categorize in this way, the number of high sociable people is 57 and, the other 157 users are categorized as low sociability. The distribution of the two groups is not balanced, which might leads to unreliable results. The program have too

much information about one class and not enough information about another class. A solution is to categorize users in three groups: high sociability (users with yes-labels for both extraversion and agreeableness), neutral sociability (user with one yes- label and one no-label) and low sociability (people with no-labels for both extraversion and agreeableness). With this method, the number of high sociability, neutral sociability and low sociability users are 57, 116 and 77, respectively. The data is still not distributed evenly. So, I decided to not use yes-/no-labels when annotating the dataset.

Another method to annotate dataset in high and low sociability is based on the personality scores. Since the scores are measured from 1 to 5, 3 is the median score. I decide to split high and low sociability round the median. Users with both extraversion and agreeableness scores higher than 3 are categorized as high sociability. All others are categorized as low sociability. Using this rule, 128 users are high sociable and 122 are low sociable. The number of status updates from high sociability is 5261 compared to 4656 status updates from low sociable users. Now, the users are nearly evenly distributed. Thus, the data is categorized with this method.

3.4 PREPROCESSING

The raw dataset contains more information than necessary. The relevant data for this research are user id, status updates, scores for extraversion and agreeableness, network size, betweenness centrality, normalized centrality, brokerage, normalized brokerage, and density. To avoid running the whole raw data and structure again and again during development of the program, the dataset needed to be preprocessed.

The raw data was read with the Python function 'csv.DictReader'. This function reads the column from CSV files by specifying the column name, and returns all values from that column. All statuses from users were put in a dictionary with user ids as keys and the values were the statuses of users, for example, 'user_1': ['status_1', 'status_2']. Next, which class the user belongs to was checked with the above method (users who score higher than 3 for both extraversion and agreeableness are high sociability). This class label was added to each user, so the dictionary becomes in this format: 'user_1': (['status_1','status_2'], label). The social network properties were also stored in the dictionary with user ids as keys and social network properties as values. Both dictionaries were stored separately in pickle files.

For the first research question which only focuses on linguistic features, a data frame with status updates and sociability label was created with the Python module 'Pandas'. The status updates and labels were read from the pickle file. Every status updates got a corresponding sociability label.

Social network properties were taken in research question 2. The data frame for experiment 2 contained status updates, sociability label, social network size, betweenness centrality, normalized betweenness centrality, brokerage, normalized brokerage, and density of users. It was exactly the same type data frame like create with only status updates, only social network features were added. See figure 2 of 'Appendices' for data frame with status updates, figure 3 for social network features in the data frame.

Because the dataset is relatively small, I performed cross-validation on the whole dataset for classification instead of splitting data into training and test sets. More details for cross validation are described in the next chapter.

4 METHOD

As mentioned earlier, the experiment is divided into 2 sub-experiments to answer the two research questions separately. The first sub-experiment only focuses on linguistic features with different classifiers, while the second sub-experiment is based on sub-experiment 1 and adds social network properties. Both sub-experiments follow the same general steps: feature selection, cross-validation, classification, and evaluation.

4.1 EXPERIMENT 1

In the first experiment, the goal is to classify the high and low sociability based on status updates with linguistic features. The data used in this experiment is the data frame with only status updates and labels created in preprocessing.

4.1.1 Tokenization

Since the status updates read from the data frame are strings of sentences, the status updates needed to be tokenized before sending the status updates to the classifier, as the classifier yielded a better score when learned from tokens features instead of whole sentence. Tokenization was done with 'nltk.word_tokenize'. This tokenizer extracts words from sentences.

4.1.2 Feature Selection

It is essential to extract linguistic features to improve the accuracy of the classification task. As mentioned earlier, according to previous studies, sentence length, punctuation, pronouns and positive and negative words are related to extraversion and agreeableness. In this experiment, the length of sentence and words and occurrences of punctuation, pronouns, and positive and negative words are counted. The most frequent 40 words used by people with extraversion or agreeableness are used as a feature. Part of Speech tagging was performed with 'nltk.pos_tag'. To measure the positivity and negativity of words, two methods were used. The first method was to use the word lists which positivity and negativity were labeled. Hereby three lists were used: positive and negative word list created by Liu (2010) and a General Inquirer Category Listings (H4LVD) . The second method was to use the toolkit 'Afinn', that calculates the positivity and negativity of the sentence. Each feature was extracted separately and tested separately to measure the effect of each specific linguistic feature.

For the purpose of getting the extracted linguistic features in the classifier, vectorizers were defined for each feature. The vectorizer was defined in Python 'Class'. Every class contains two functions: 'fit' and 'transform'. The function 'fit' is an empty function, which returns 'self'. The function 'transform' returns the list with feature dictionaries of each status update. The

class takes the scikit-learn class 'TransformerMixin' as parameter, which fits and transforms the data into arrays. To keep the program simple, all linguistic features were extracted with the same structure. Thus, all linguistic features were structured in dictionaries with the same format: {'feature_1': 2, 'feature_2': 5}. The value of the feature means the occurrence of the feature in the status update. The dictionary for Pos-tagger was structured in: {'word_1':tag,'word_2':tag}. For an example of vectorizer see figure 4 in 'Appendices'.

4.1.3 Cross Validation

The dataset contains 9917 status updates of Facebook users, which is a relatively small number. To test all data in the dataset, I used 10-fold cross validation. It split the dataset into 10 parts; each time the program takes nine parts as training set and one part as a test set. The process was performed 10 times until all data from the dataset were trained and tested. The data was randomly split.

One of the parameters of the k-fold cross is 'random state'; this was randomly set to 0. When setting the parameter 'shuffle' to 'True', all status updates of users were mixed. In other words, the order of status updates is changed and status updates from one user might be split into training and test set. When the parameter 'shuffle' is set to 'False', the order of status updates is not changed and when a user is in the test set, there is no information about this user in the training set. When testing with the parameter 'shuffle=False', the accuracies of prediction based on linguistic features do not show significant improvements compared to the baseline. Since this experiment is only based on linguistic features and every status update is unique, the 'shuffle' is set to 'True'.

4.1.4 Classification

There are different classifiers that properly fit in text classification tasks. To figure out which works best in this dataset, three of the most commonly used classifiers are used in this research. The classifiers applied in this research are: Naïve Bayes, Support Vector Machine and decision tree.

Naïve Bayes classifier is available in scikit-learn—it is callable with the function 'MultinomialNB'. The parameter set to this function is 'alpha=0.01', which shows the best score after several tests with other parameters. The Support Vector Machine classifier is useable with the scikit-learn function 'SVC'; it is 'rbf' kernel and no extra parameter are set. The function 'DecisionTreeClassifier' can perform the classification task with the decision tree.

In this research, the classifier takes two types of input. The first input is the tokenized status updates. This input is adding to the classifier with the scikit-learn function 'TfidfVectorizer'. This function takes the tokenizer as parameter and then tokenize the input data into tokens. For each tokens it calculates Tf-idf ('tf' refers to the occurence of a word in the status update, 'idf' means the inversion of the occurences of the word in all status updates) for each word per status update. Additionally, N-grams are taken as a parameter of 'TfidfVectorizer'; the range of n-gram is from 1 to 4. This range was selected because it returned the best score after several tests with other possible ranges. The arrays was converted with the scikit-learn function 'TfidfTransformer' into sparse features.

The linguistic features are the second input to the classifier. For the purpose of combining features to the classifier, the scikit-learn function 'FeatureUnion' was applied. This function combines all features and the classifier in a 'Pipeline', which allows the classifier train and predict the input data based on multiple features. To let the classifier accept and use the features, they needed to transform into sparse feature representation. This was done with the scikit-learn function 'DictVectorizer'. The 'DictVectorizer' converted the arrays created in section 'Feature Selection' into sparse features.

To show the effect of each classifier and linguistic feature, every linguistic feature was tested separately with each classifier. For example, when testing the effect of pronouns on the classification with SVM, the pipeline contained the following components: tokenized status updates with 'TfidfVectorizer' and 'TfidfTransformer', pronouns that convert with 'DictVectorizer', and SVM classifier. The result of testing showed that some linguistic features performed worse than when classified only with tokenized status updates with 'TfidfVectorizer' and 'TfidfTransformer'. These linguistic features have a negative impact on the classification task.

The final step of this experiment was to combine all effective linguistic features into the classifier, and test which classifier yielded the better performance. Thus when combining all linguistic features, linguistic features that impact the classification task negatively were not added to the pipeline of the classifier.

4.2 EXPERIMENT 2

In this experiment, the classification task is based on both linguistic and social network features. This experiment was built based on experiment 1 with the addition of, the social network properties. Before implementing the social network features into the classifier, it is essential to check whether there are association between sociability and social network properties. These associations were calculated with Fisher's exact test.

4.2.1 Correlation social network properties and sociability

To investigate whether there is a correlation between social network properties and sociability, the Fisher's exact test was performed. Fisher's exact test determines the association between two nominal variables. Each social network property is split into two groups using the median split. All scores lower than the median are considered as low-score. Higher scores from the median are grouped as high-score. Fisher's exact test was performed separately for all social network properties. This test was performed with a 2x2 cross table from SPSS. It calculated the frequency of low and high scores appearing in low and high sociability. To determine the effect of the strength of the association between sociability and social network properties, Cramer's V was applied.

4.2.2 Social network features

All social network properties are given in numerical values. There are in total five social network properties (network size, betweenness centrality,

brokerage, transitivity, and density). The normalized scores for betweenness centrality and brokerage are also provided in the dataset and were structured in the data frame during preprocessing of data.

The social network properties were stored in columns in the data frame. Multiple features have to be combined in the pipeline. Except for linguistic features, social network features also needed to be extracted from the data frame. To select the specific column in a data frame and fit the values of the selected column into pipeline of classifier, the scikit-learn function 'FunctionTransformer' was applied. This function selects the values from a column and converts the values into arrays. The converted arrays were transformed into sparse features with function 'numpy.matrix' in a self-defining vectorizer. The vectorizer is defined with the same method as in experiment 1.

4.2.3 Cross validation

10-fold cross validation was also performed in this experiment. Both 'False' and 'True' settings for the parameter 'shuffle' are used in this experiment, since there is a large difference of the accuracies of classify before and after shuffling the dataset. The 'random_state' was also randomly set to '0' as experiment 1.

4.2.4 Classification

As experiment 1, Naïve Bayes, SVM, and decision tree classifiers were applied. The same parameter (alpha=0.01) was used for Naïve Bayes. Since there is a difference of the performance in linear kernel of SVM and 'rbf', both kernels are applied when classifying with SVM.

For each of the linguistic features, the column status updates was selected with the scikit-learn function 'FunctionTransformer'. The same method of processing the linguistic features as experiment 1 is performed in this experiment. The social network features were also taken into the pipeline with 'FunctionTransformer' and convert into sparse features. The same structure of pipeline and the feature union for classifying in experiment 1 was used in this experiment. The social network features were tested separately with each classifier; the effective features were combined to test a final accuracy.

4.3 EVALUATION

For both experiments, to evaluate the performance of the program, the final scores are compared to the baseline. The average scores were calcultated for each 10-fold cross validation.

4.3.1 Baseline experiment 1

Since a 10-fold cross validation is performed during classification, the dataset is randomly split into 90% training set and 10% test set when calculating the baseline. After tokenizing the dataset with 'NLTK.word_tokenize', the tokenized words were taken to the classifier. A dummy classifier (scikit-learn class: 'DummyClassifier') classified the dataset with the simple rules, in this case, it predicts data with the distribution of training set' class.

4.3.2 Baseline experiment 2

Since experiment 2 is based on experiment 1, the baseline of experiment 2 is the final results of experiment 1. In this way, experiment 2 shows whether there are any sociability prediction from Facebook status updates improves or worsens when social network features are added. An additional baseline to experiment 2 is to get the accuracy of all effective linguistic features per classifier when the parameter 'shuffle' is set to 'False', as this helps evaluate the performance of adding social network properties with 'shuffle' is set to 'False'.

5 RESULTS AND DISCUSSION

5.1 EXPERIMENT 1

Linguistic features

Table 1: Accuracy of linguistic features separately with shuffling dataset.

	SVM	NB	Decision tree
Tf-idf	0.625	0.615	0.537
Extra words	0.624	0.614	0.535
Punctuation	0.625	0.612	0.533
Pronouns	0.626	0.615	0.536
Pos Neg	0.626	0.614	0.538
Pos-tag	0.604	0.614	0.538
Afinn	0.624	0.615	0.535
H4Lvd	0.622	0.614	0.540
Text Length	0.622	0.614	0.544

Table 1 shows the accuracies of linguistic features separately with each classifier with shuffling the dataset. The results of 'shuffle='False' is available in figure 7 in 'Appendix'. The accuracy scores of Tf-idf weighting with SVM and Naive Bayes are around 0.62, and there is little difference in accuracy after adding other features to the classifier. Accuracy scores for decision tree are all around 0.54, which is much lower than the results of other two classifiers.

Taking Pos-tagger as a feature, the accuracy of SVM is 0.59, which is 2% lower than tokenizing the status updates with TfidfVectorizer. Taking Pos-tagger as a feature with NB, yields the highest accuracy 0.620 and it is the only effective feature with NB. Pos-tagger performed best using NB and decision tree, while it has the lowest score using SVM. The effective features with SVM are: punctuation, pronoun, and positive and negative words. Pronouns, positive and negative words, Pos-tagger, H4Lvd word list and length of sentence are effective linguistic features with the decision tree classifier.

Some significant features mentioned in previous research were not significant in this study. Pennebaker and King (1999) pointed out that extraversion and agreeableness positively correlated with positive words, while both positive and negative words appear in both high sociability and low sociability in this research. The number of positive words occurs in high sociability is 2088 and 1754 for low sociability. The difference between the occurrences of positive words in high and low sociability is not large.

One reason for this might be that all features are significant for either extraversion or agreeableness. Some features might be significant for extraversion, while it has no effect on low or high agreeableness.Gill and Oberlander (2002) mentioned that extroverts wrote longer sentences, while they did not found correlation between agreeableness and sentence length. When I combine the two personalities into one category, the effect of the feature might be neutralized.

There is a difference between the results of classifying before and after shuffling the dataset. When the status updates from a user are split into training and test set, the accuracy is higher than before shuffling the dataset. Each user has their own writing style. When the program learns the label from the training set for this writing style, it is easier to classify the status updates in the test set when meets the same style again.

Combining all linguistic features

Table 2: Accuracy of combining all effective linguistic features .

	SVM	NB	Decision tree
Baseline	0.5	0.5	0.5
classification	0.626	0.620	0.546

The results of combining all effective features show that SVM yields the best performance; it is 12.6% higher than the baseline. The accuracy of the Naive Bayes classifier has a slight advantage over the decision tree classifier. The highest score of Naive Bayes is 12% higher than the highest score of the Decision tree.

The reason that SVM performed the best out of the three classifiers might be that it fit the data into a regression model and did not heavily rely on the words that appear in the training set as much as the NB and the decision tree did. Based on the typical words in the training set, the NB and the decision tree calculate the probabilities that words occur in both classes and use that to classify the status updates. A sentence with words that only appeared in test set is difficult to classify with NB and decision tree.

The small corpus had a large effect on the performance of all three classifiers. The dataset contains 9917 status updates, while 3706 status updates contain fewer than 10 words. This is approximately 37% of the dataset. The average length of status updates is 18. From these short sentences is hard to disguise the personality of people, it is also not easy for human to perform the task.

5.2 EXPERIMENT 2

Correlation social network features and sociability

Table 3: Correlation social network features and sociability .

	Fisher's Exact test	Effect size
Network size	0.017	0.152
Betweenness Centrality	0.057	0.128
Normalized betweenness	0.023	0.153
Brokerage	0.016	0.160
Normalized brokerage	0.039	0.151
density	0.022	0.151

The results of Fisher's exact test showed that all significant values of network properties, except for betweenness centrality, are below the 0.05 significance level. When the p-value is lower than 0.05, it means that there is a significant association between the two nominal values. In this case,

except for betweenness centrality, all network properties are associated with sociability. Furthermore, an effect size of 0.10 to 0.29 is considered a weak correlation. Cramer's V score for all social network properties is from 0.128 to 0.160. Therefore, there is a weak correlation between sociability and the mentioned social network properties.

Social network features

Table 4: Accuracy of social network properties separately with shuffling dataset.

	SVM	NB	Decision tree
Network size	0.647	0.620	0.950
Betweenness	0.955	0.606	0.970
Normalized Betw	0.530	0.620	0.936
Brokerage	0.975	0.607	0.964
Normalized brok	0.530	0.620	0.577
Density	0.630	0.620	0.608

When 'shuffling' was set to 'True', the accuracies of classifying with social network features are extremely high by some social network features. When taking brokerage as a feature, shuffling the dataset and classifying with SVM and decision tree, the accuracies are 0.975 and 0.964, respectively. But normalized brokerage and normalized betweenness centrality only achieves 0.530. Normalization of brokerage convert the large numbers of brokerage (for example, the brokerage of a user is 70505) to numbers in scale from 0 to 1. This leads to that multiple users having the same normalized brokerage, regardless high or low sociability, which reduces the effect on the classification with SVM and decision tree. The results of NB are relatively stable; the accuracies are between 0.606 and 0.620. When no shuffling of the dataset is performed, the accuracies of all three classifiers with social network features are between 0.506 and 0.607.

Table 5: Accuracy of social network properties separately without shuffling dataset.

	SVM	NB	Decision tree
Baseline not shuffle	0.523	0.530	0.50
Network size	0.549	0.510	0.544
Betweenness	0.534	0.506	0.522
Normalized Betw	0.512	0.512	0.554
Brokerage	0.607	0.506	0.549
Normalized brok	0.515	0.515	0.513
Density	0.527	0.515	0.608

Compare to the baseline, social network features improve the accuracy when classifying with SVM and decision tree with 'shuffle' is set to 'False'. The most effective social network feature by SVM is brokerage, the accuracy is 0.607. Density performs the best with decision tree and yields an accuracy of 0.608. None of the social network features are effective with NB classifier. This might be caused by that some social network features are unique numbers, when there are no identical number in the training set, NB cannot calculate the probability of the belonging class of this number based on the known information. While SVM do not require identical data, it can predict sociability based on the data in the same feature sparse.

Combining all features

Table 6: Combining all effective linguistic features and social network features .

	SVM	NB	Decision tree
Baseline shuffling	0.626	0.620	0.546
Shuffling	0.996	0.575	0.986
Baseline without	0.523	0.530	0.50
Without Shuffling	0.595	0.495	0.554

Because all social network features with the decision tree classifier yield higher accuracy scores than the baseline, all social network properties are taken together for the final classification. Shuffling the dataset achieves extremely high accuracy score with SVM and the decision tree classifiers (with the highest accuracy of 97.5%). The improvements of classifying with a shuffled dataset with SVM and the decision tree are 36.96% and 44.05%, respectively. The accuracy for NB was decreased by 4.5% after adding social network properties.

From the accuracy score of without shuffling dataset, it can be concluded that social network properties have a slight impact on the prediction of sociability. As the results of Fisher's exact test show, there is a weak correlation between social network properties and sociability. The improvement in prediction is 7.2% with SVM and 5.59% with the decision tree classifier. The accuracy score of NB was decreased by 3.53%.

When the statuses of a user are not split randomly into training and test sets, the accuracy scores are from 0.495 to 0.595. The reason for the high accuracy is that each user has written more than one status update, and social network properties are assigned to each status update. In other words, multiple status updates contain the same social network properties. Each user has a unique number for betweenness centrality and brokerage. When these status updates are split into training and test sets, the model can learn from the social network properties and label the status updates in the test set with the same label as in the training set. This is an overfitting problem. The performances are too perfect in the training set, while when the program meets new and unseen data, the performance decreases. Overfitting error comes easily when using a decision tree classifier, thanks to the rules the model created for decision making. When the program knows the answer for the test data from the training set, the classification is very accurate. In contrast, when the program is unfamiliar with the test data, it is difficult for the program to make the right decision with a decision tree. The Naïve Bayes classifier is relatively immune to overfitting problems, because it relies on probability algorithms. It calculates the probability of each status update and features and determines which class it belongs to.

6 CONCLUSION

This research investigates the possibility of sociability recognition with machine learning methods. It is possible to predict sociability from Facebook status updates with SVM, NB, and decision tree classifiers. Due to the short length of the status updates and small numbers of statuses in the sample, the linguistic feature selection does not show a significant effect on the prediction of sociability. The most frequent words of extraversion and agreeableness listed by Yarkoni (2010) do not show any improvement on the classification task with all three classifier. Using pronouns, positive and negative words and Pos-tagger as features with the Support Vector Machine classifier yields the best performance with an accuracy score 0.626. This is not consistent as the result of Alam et al. (2013). They found that NB was the most effective classifier to predict personality. In this study, SVM performs better than NB. This might be caused by that different method used, the difference in programming, and settings or parameters that were given to the classifier.

Amichai-Hamburger and Vinitzky (2010) found that personality is connected with the use of Facebook. This research found that social network properties are weakly correlated with sociability. These social network features can improve the prediction of sociability using the SVM and decision tree classifiers. No improvement using the NB classifier was found. Train and test the status updates from the same user lead to overfitting error because every status updates from the same user shares the same social network values. It increases the accuracy from 0.6 to 0.99. However, test with unseen data also improves the performance of the prediction. The improvement of SVM is 7.2%. Due to the size of the dataset, the program does not perform well overall. However, SVM is the best classifier with a small corpus.

The small corpus limits the performance of classification. Also, the linguistic features are not significantly correlated to sociability. Future work can be based on small corpus and try to find methods which improve the accuracy of classification. Also, it will also be great to find whether if there are any significant linguistic features which can be used to predict the personalities based on Facebook status updates with larger corpus.

BIBLIOGRAPHY

Alam, F., E. A. Stepanov, and G. Riccardi (2013). Personality traits recognition on social network-facebook. *WCPR (ICWSM-13), Cambridge, MA, USA*.

Amichai-Hamburger, Y. and G. Vinitzky (2010). Social network use and personality. *Computers in human behavior 26*(6), 1289–1295.

Barrick, M. R. and M. K. Mount (1991). The big five personality dimensions and job performance: a meta-analysis. *Personnel psychology 44*(1), 1–26.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of mathematical sociology 25*(2), 163–177.

Celli, F., F. Pianesi, D. Stillwell, and M. Kosinski (2013). Workshop on computational personality recognition (shared task). In *Proceedings of the Workshop on Computational Personality Recognition*.

Cheek, J. M. and A. H. Buss (1981). Shyness and sociability. *Journal of personality and social psychology 41*(2), 330.

Costa, P. T., R. R. McCrae, and D. A. Dye (1991). Facet scales for agreeableness and conscientiousness: A revision of the neo personality inventory. *Personality and individual Differences 12*(9), 887–898.

Eysenck, H. J. (1956). The inheritance of extraversion-introversion. *Acta Psychologica 12*, 95–110.

Friggeri, A., R. Lambiotte, M. Kosinski, and E. Fleury (2012). Psychological aspects of social communities. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pp. 195–202. IEEE.

Gill, A. and J. Oberlander (2002). Taking care of the linguistic features of extraversion. In *Proceedings of the 24th annual conference of the cognitive science society*, pp. 363–368.

Golbeck, J., C. Robles, and K. Turner (2011). Predicting personality with social media. In *CHI'11 extended abstracts on human factors in computing systems*, pp. 253–262. ACM.

Haythornthwaite, C. (1996). Social network analysis: An approach and technique for the study of information exchange. *Library & information science research 18*(4), 323–342.

Iacobelli, F., A. J. Gill, S. Nowson, and J. Oberlander (2011). Large scale personality classification of bloggers. In *Affective computing and intelligent interaction*, pp. 568–577. Springer.

Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*, pp. 627–666. Chapman and Hall/CRC.

Mairesse, F., M. A. Walker, M. R. Mehl, and R. K. Moore (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research 30*, 457–500.

Markovikj, D., S. Gievska, M. Kosinski, and D. J. Stillwell (2013). Mining facebook data for predictive personality modeling. In *Seventh International AAAI Conference on Weblogs and Social Media*.

McCrae, R. R. and O. P. John (1992). An introduction to the five-factor model and its applications. *Journal of personality 60*(2), 175–215.

Pennebaker, J. W. and L. A. King (1999). Linguistic styles: language use as an individual difference. *Journal of personality and social psychology 77*(6), 1296.

Tkalcic, M., M. Kunaver, J. Tasic, and A. Košir (2009). Personality based user similarity measure for a collaborative recommender system. In *Proceedings of the 5th Workshop on Emotion in Human-Computer Interaction-Real world challenges*, pp. 30–37.

Wehrli, S. et al. (2008). Personality on social network sites: An application of the five factor model. *Zurich: ETH Sociology (Working Paper No. 7)*.

Wright, W. R. and D. N. Chin (2014). Personality profiling from text: introducing part-of-speech n-grams. In *International Conference on User Modeling, Adaptation, and Personalization*, pp. 243–253. Springer.

Yarkoni, T. (2010). Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality 44*(3), 363–373.

A APPENDICES

To review the code of the program, visit the github repository.
Github: https://github.com/chenxiaoying/scriptie.git

Figure 1: Raw data .


```
"#AUTHID","STATUS","sEXT","sNEU","sAGR","sCON","sOPN","cEXT","cNEU","cAGR","cCON","cOPN","DATE","NETWORKSIZE"
,"BETWEENNESS","NBETWEENNESS","DENSITY","BROKERAGE","NBROKERAGE","TRANSITIVITY"
"b7b7764cfa1c523e4e93ab2a79a946c4","realizes traditional media hate her and has shunned her penvcils in favor
of the tablet. That'll learn 'em.",2.65,3.00,3.15,3.25,4.40,"n","y","n","n","y",11/23/09 07:17
PM,180,14861.6,93.29,0.03,15661,0.49,0.1
"318bf822d4f2bd3920367560218619c0","rhinanna is just getting crazier and crazier. or more
stupid.",4.50,4.00,3.00,4.50,3.75,"y","y","n","y","n",09/10/09 02:58
AM,318,49024.8,97.88,0.02,49584,0.49,0.06
"318bf822d4f2bd3920367560218619c0","is flashing lights kanye
style",4.50,4.00,3.00,4.50,3.75,"y","y","n","y","n",10/15/09 04:02 AM,318,49024.8,97.88,0.02,49584,0.49,0.06
"318bf822d4f2bd3920367560218619c0","i got a break! a $3000 healthcare break!
whoooooo",4.50,4.00,3.00,4.50,3.75,"y","y","n","y","n",10/14/09 03:55
AM,318,49024.8,97.88,0.02,49584,0.49,0.06
"ecbddbfe00e0f83cfdb802a7186061c7","2
DAYS!!!!!!!!!!!!!!!",4.30,2.15,3.60,3.30,4.10,"y","n","y","n","y",09/18/09 03:13
PM,739,267574,98.39,0.01,270029,0.5,0.07
"ecbddbfe00e0f83cfdb802a7186061c7","wishes it was December 17th sooner :
(",4.30,2.15,3.60,3.30,4.10,"y","n","y","n","y",12/06/09 11:34 PM,739,267574,98.39,0.01,270029,0.5,0.07
"ecbddbfe00e0f83cfdb802a7186061c7","cannot wait to pass out - too bad that's not gonna happen til friday
night...",4.30,2.15,3.60,3.30,4.10,"y","n","y","n","y",12/17/09 07:11
PM,739,267574,98.39,0.01,270029,0.5,0.07
"4d035bd3fd8d9595d15cea9e388964be","had a wonderful night with family and friends... the night cap was
wonderful though... WEEEEE!! I got my shirt!!!",3.70,2.90,3.40,3.35,4.05,"y","y","n","n","y",12/27/09 04:04
AM,57,1509.5,98.02,0.05,1522,0.49,0.03
"172400f46880b309ca5e97d322bb8f01","""Necessity is the plea for every infringement of human freedom. It is
the argument of tyrants; it is the creed of slaves."" -
*PROPNAME*",3.45,2.85,2.80,2.70,4.15,"n","y","n","n","y",11/06/09 07:59
PM,122,6529.3,89.94,0.07,6893,0.47,0.26
"172400f46880b309ca5e97d322bb8f01","Progress is made by lazy men looking for easier ways to do things.-
Heinlein",3.45,2.85,2.80,2.70,4.15,"n","y","n","n","y",01/20/10 08:18 PM,122,6529.3,89.94,0.07,6893,0.47,0.26
```


Table 7: Accuracy of linguistic features separately without shuffling.

	SVM	NB	Decision tree
Tf-idf	0.524	0.519	0.50
Extra words	0.521	0.518	0.506
Punctuation	0.523	0.518	0.506
Pronouns	0.526	0.517	0.504
Pos Neg	0.523	0.518	0.50
Pos-tag	0.51	0.530	0.506
Afinn	0.524	0.519	0.512
H4Lvd	0.524	0.517	0.509
Text Length	0.526	0.516	0.509

Figure 2: Textual data in data frame .

	label	text
0	low-social	A textbook a week, argh...
1	low-social	Still recovering... I look like a big fly!0_0
2	low-social	can read with his right eye now. No worries. ^^
3	low-social	Happy Thanksgiving everyone~
4	low-social	Addicted to Les Miserables...
5	low-social	Block 2 is over! Time to relax...
6	low-social	Finally, a three day weekend to.... catch up w...
7	low-social	Moving is such a hassle
8	low-social	Memory~ All alone in the moonlight~ I can smil...
9	low-social	for the united fans reading, dont worry be hap...
10	low-social	just read the mark hughes interview in the mir...
11	low-social	got a job working on the seasfront..love it!
12	low-social	"The road of excess leads to the palace of wis...
13	low-social	tip for today..never talk to a potential emplo...
14	low-social	tony gubba does my nut in
15	low-social	got sick of the crap 5 haircuts so upgraded to...
16	low-social	owen gets the number seven?! you just have to ...
17	low-social	No time table on life, always follow your heart
18	low-social	any cougars in ross?
19	low-social	Any *PROPNAME* Jokes?
20	low-social	Barcelona, bueno aires, rio de janerio, san di...
21	low-social	*PROPNAME* to miss play off....good start to t...
22	low-social	really looking forward to sunday
23	low-social	being robbed of 3 points is one thing but a wo...
24	low-social	classic reality check from keane
25	low-social	well holy shit i think we could make the world...
26	low-social	Just watched the *PROPNAME* press conference.....
27	low-social	money is a disease
28	low-social	down to 10 men and we're still kicking arse
29	low-social	I was never a huge Michael Jackson fan, but fo...
30	low-social	mad river & uncle Fatty's tnite, Rockit tmrw, ...
31	low-social	The world seems a lot different without one of...
32	low-social	snipers get more head
33	low-social	likes the sound of thunder.
34	low-social	is so sleepy it's not even funny that's she ca...

Figure 3: Social network features in data frame .

	betw	brok	den	label	netw_size	norm_betw	norm_brok	\
0	68247.60	70505	0.03	low-social	381	94.78	0.49	
1	68247.60	70505	0.03	low-social	381	94.78	0.49	
2	68247.60	70505	0.03	low-social	381	94.78	0.49	
3	68247.60	70505	0.03	low-social	381	94.78	0.49	
4	68247.60	70505	0.03	low-social	381	94.78	0.49	
5	68247.60	70505	0.03	low-social	381	94.78	0.49	
6	68247.60	70505	0.03	low-social	381	94.78	0.49	
7	68247.60	70505	0.03	low-social	381	94.78	0.49	
8	68247.60	70505	0.03	low-social	381	94.78	0.49	
9	9100.53	9251	0.04	low-social	139	96.27	0.49	
10	9100.53	9251	0.04	low-social	139	96.27	0.49	
11	9100.53	9251	0.04	low-social	139	96.27	0.49	
12	9100.53	9251	0.04	low-social	139	96.27	0.49	
13	9100.53	9251	0.04	low-social	139	96.27	0.49	
14	9100.53	9251	0.04	low-social	139	96.27	0.49	
15	9100.53	9251	0.04	low-social	139	96.27	0.49	
16	9100.53	9251	0.04	low-social	139	96.27	0.49	
17	9100.53	9251	0.04	low-social	139	96.27	0.49	
18	9100.53	9251	0.04	low-social	139	96.27	0.49	
19	9100.53	9251	0.04	low-social	139	96.27	0.49	
20	9100.53	9251	0.04	low-social	139	96.27	0.49	
21	9100.53	9251	0.04	low-social	139	96.27	0.49	
22	9100.53	9251	0.04	low-social	139	96.27	0.49	
23	9100.53	9251	0.04	low-social	139	96.27	0.49	
24	9100.53	9251	0.04	low-social	139	96.27	0.49	
25	9100.53	9251	0.04	low-social	139	96.27	0.49	
26	9100.53	9251	0.04	low-social	139	96.27	0.49	
27	9100.53	9251	0.04	low-social	139	96.27	0.49	
28	9100.53	9251	0.04	low-social	139	96.27	0.49	
29	325950.00	333695	0.01	low-social	822	96.83	0.50	
30	325950.00	333695	0.01	low-social	822	96.83	0.50	
31	325950.00	333695	0.01	low-social	822	96.83	0.50	

Figure 4: Positive and Negative words vectorizer. Each time a word from status updates also occurs in positive word list, count 1 for 'is_pos'. Each time negative words in negative word list, plus 1 by 'is_neg'

```
class PosNeg(TransformerMixin):

    def fit(self, x, y=None):
        return self

    def transform(self, data):
        feat = []
        pos_wordlist, neg_wordlist = pos_neg_words()
        for lines in data:
            line = nltk.word_tokenize(lines)
            pos, neg = 0, 0
            for word in line:
                if word in pos_wordlist:
                    pos += 1
                if word in neg_wordlist:
                    neg += 1
            feat.append({'is_pos': pos, 'is_neg': neg})
        return feat
```