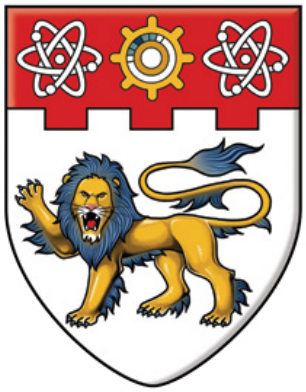

NANYANG TECHNOLOGICAL UNIVERSITY

CZ4045 Natural Language Processing

Course Assignment

Group 12

Semester 1 2015/2016

Chen HaoMeng

Jasek Otakar

Tran Minh Tri

Tye Yong Meng

Tan Dai Zhen Joyce

Table of Contents

5.0 Before Crawling

5.0.1 Introduction

Prediction has always been the talk of the town when popular events are going to take place. In the latest clash between boxing legends Floyd Mayweather and Manny Pacquiao, they even had boxing legends of previous decades to come together to provide expert analysis on which fighter would win the boxing match. In the end Floyd Mayweather won. It would not have surprised anyone who believed in numbers because Floyd had a perfect winning streak of 47-0 while Manny had 5 losses in his fight career before the fight. We have chosen to predict the election results for the US because it is the next big thing.

5.0.2 Our Approach

We will be crawling a corpus out of Reddit and Twitter for the comments made about two presidential candidates; Donal Trump and Hillary Clinton. This will be done by our python scripts which will call upon available APIs from Reddit and Twitter.

Reddit is an online community which allows its users to post any comments and vote on content. [1] According to the company's page, it reported on 15 September 2015 that in August 2015, it had 202,818,688 unique visitors from 208 different countries. [2] In addition, US president Barack Obama uses it to reach out to his supporters. [1] As such, we believe that Reddit is a good resource to predict the winning outcomes of US presidential candidates.

Twitter is a social media which allows users to post or reply via tweets. Tweets are text messages and have a limit of 140 characters. Tweets can be viewed by anyone regardless of being a member or not. [3] Twitter currently has 316 million of monthly active users and 23% of the accounts reside in the US. If we take 23% of 316 million users, we have 72.68 million users which is still a substantial amount. [4]

After gathering the corpus, we would use sentiment analysis to separate positive comments from negative ones. Instead of just analysing text, we will include emoticons. We will then compare the ratios of positive and negative comments

between the two candidates. Our hypothesis is that the candidate with a higher ratio of positive comments will stand a higher chance of winning the election.

5.0.3 Other Approaches

5.0.3.1 Phone Polling

Phone polling was conducted by polling companies in Britain. They collected a sample of 1,000 people arbitrarily and weighted the samples according to demographics such as age, location and gender to represent the population. Polling companies apply different weights on the samples based on certain criteria. An example of criteria would be asking the participants' past voting choice and use this information to weight their voting intentions accordingly. The biggest limitation using this approach is that is it unsustainable in the future even though it proves to be more accurate than online polling most of the time. Internet polling is less costly and can gather more information from more people. As such, it would be a better representative of the population. [5] Hence, in the near future, phone polling will most likely be replaced with online polling.

5.0.3.2 Data Science

Nate Silver is one of the famous statisticians who shot to fame in 2008 when he forecasted the outcome of the primaries and presidential victors in 49 states. [6] He is the founder of FiveThrityEight website which publishes political articles and his predictions. [7] The success of his prediction was attributed to many factors. Nate Silver gathered data from a very broad scope ranging from demographics to economic variables. The crucial factor was his ability to choose the correct regression models with complex statistical modelling software. [8] It may seem that Nate Silver's method of prediction is the answer for all political predictions. However, his predictions were far from accurate for the UK election this year. In addition, his predictions became less accurate since 2012 US presidential election. [9] As such, there is a need to find an alternative solution to predict the political results.

5.0.3.3 Research Papers

Similar research has been done to analyze comments from twitter to predict the outcome of elections. In this research, the way it does its sentiment analysis is by

assigning a sentence to be positive if there are any positive words in the sentence. As such, in the case of a sentence with a positive word and a negative word, it can be both a positive and negative statement at the same time. The lexicon used is from OpinionFinder. From the sentiment analysis, the two presidential candidates which it compared were Obama and Mccain. Since the campaign can only have one winner, it is expected that the sentiment for one candidate would vary inversely from each other. However, they seem to slightly correlate in the sentiment analysis. [11]

In this research paper, it found other sources that proved traditional social media to be a reliable option to predict election outcomes. However, the same thing cannot be said for twitter because tweets are a mere 140 characters. In addition to this, a market consultancy even said that 40% of the tweets are "pointless babble". This research paper focused on German election. The methods used were downloading tweets in German and then translating it to English. Thereafter, it used the LIWC2007 (Linguistic Inquiry and Word Count) to assess the emotional components of tweets. The research found out that twitter though was dominated by a small number of heavy users, the tweet volumes is close to the results of federal election. [12]

This research paper acknowledges the usefulness of using tweets to predict certain things like movie successes but not so much for elections. It uses the algorithms from [11 & 12] on the 2010 US Senate special election in Massachusetts to prove that the success of predicting the elections is a coincidence because it was not repeatable on another data set. There are a few possible reasons for this failure to predict. One of them is the manipulation by spammers. Fake accounts can be easily created and by spamming positive remarks on a certain politician can distort the view of any observer. [13]

This research paper agrees with [13] and took a step further by including more information from the tweets such as geo location of each user. It has two algorithms. The first one gets the location of the user through the location field while the second one checks the confidence of the predicted location based on the contents it received from algorithm one. Algorithm 2 is necessary because users sometimes key in irrelevant information such as 'bedroom' as the location. At the end of it, it concluded that it is feasible to predict American presidential elections using tweets but there are

several limitations. One of them is the current programs does not integrate the dynamics of political conversations in social media. [10]

Another research paper suggested that a web-derived lexicon will bring about a tremendous improvement on a lexicon-based sentiment classifier. [14]

So it seems that the current technology or level of research fails to predict the outcome of elections consistently. As such, we do not plan to come up with a new algorithm which ambitiously aims to predict accurately our all data sets due to the time constraint. Hence, our team has decided to investigate on the contents of social media which were not included in these researches. We will be focusing on the effect of emoticons to better classify (positive or negative) the comments made by users.

5.1 Corpus Crawling

Question 1

The statistics for each candidate includes documents crawled from both Twitter and Reddit.

Statistics for Hillary Clinton crawls

- Number of sentences = 13448
- Number of tokens = 163817
- Number of type = 11014
- Number of documents = 9024

Statistics for Donald Trump crawls

- Number of sentences = 45206
- Number of tokens = 543019
- Number of type = 17796
- Number of documents = 27135

5.2 Learn a Deep Learning Model With The Crawled Corpus

Based on our twitter corpus from Donald Trump, we sorted all the nouns according to the number of appearances they have. We choose 5 of the most frequent words as our key words: Trump, politics, president, candidate and opinion.

Word2vec: Word2vec is a toolkit provided by Google, providing a method to build vectors model based on continuous bag-of-words (CBOW) and skip-gram architectures. It can translate a word into its vector representation, as well as training a model from a large corpus to support various analysis tasks.

Based on vector representation of words, we can measure their relations with each other. We perform 3 tasks based on 3 questions of section 5.2 of the assignment.

Question 2

Given vector representation of words in our corpus, we simply do a normalized dot product (cosine distance) to compute their "distance". The distance is 1 if they are equal (in direction), smaller "distance" means the words are more different.

Outcome For Our Test: *(Code: /5.2/NLP 5.2/word-analogy.c)*

Enter word (EXIT to break): politics

Word: politics Position in vocabulary: 1764

Word	Distance
soaring.	0.859098
4.21	0.855930
times	0.851998
#Trump.	0.845365
knows	0.843575
good	0.836803
Current-American	0.834254
stands	0.830490
liberals	0.825006
its	0.824097

Enter word (EXIT to break): Trump

Word: Trump Position in vocabulary: 38

Word	Distance
#Stumped	0.647869
http://t.co/O5v9oX3pydCarson	0.641029
Donald	0.639887
@LindaSuhler:	0.601711
Blasts	0.562340
vote	0.542295
Michigan!	0.536790
@WashTimes:	0.530438
#Trump2016Carson	0.529623
Compares	0.506741

Enter word (EXIT to break): president

Word: president Position in vocabulary: 847

Word	Distance
taking	0.874387
chance	0.872044
worse	0.865401
Josh	0.856510
gave	0.852727
class	0.849465
likes	0.843745
Muslims	0.836635
defense	0.835478
doesn't	0.833406

Enter word (EXIT to break): candidate

Word: candidate Position in vocabulary: 578

Word	Distance
political	0.819578
somebody	0.811906
told	0.809150
dinner	0.808545
America.	0.799716
few	0.797652
liberal	0.797198

hard!-	0.791712
donate	0.789009
threatening	0.787662

Enter word (EXIT to break): opinion

Word: opinion Position in vocabulary: 481

Word	Distance
American	0.864201
Islamophobe	0.853458
#LiberalDelusion	0.845501
Voter?	0.843300
an	0.813199
correcting	0.799435
donate	0.793326
racist,	0.698671
tune	0.657279
tunnel,	0.647378

Question 3

We change the corpus to Google News dataset to search for closest words to each of the five chosen keywords.

This dataset is much bigger than ours (3.4 GB in size compare to 6.4 MB), thus the result is also better: words are more natural and correct.

Outcome For Our Test: *(Code: /NLP/NLP 5.2/word-analogy.c)*

Enter word (EXIT to break): politics

Word: politics Position in vocabulary: 2029

Word	Distance
partisan_politics	0.683224
Politics	0.674026
political	0.671894
politcs	0.622195
poltics	0.594164
Lisa_Vorderbrueggen_covers	0.586606
partisanship	0.573556
politicians	0.570558
politician	0.569530
politicking	0.568017

Enter word (EXIT to break): Trump

Word: Trump Position in vocabulary: 13034

Word	Distance
Donald_Trump	0.810392
impersonator_entertained	0.594226
Ivanka_Trump	0.592458
Ivanka	0.560721
mogul_Donald_Trump	0.559245
Trump_Tower	0.548555
Kepcher	0.546859
billionaire_Donald_Trump	0.544727
Trumpster	0.541282
tycoon_Donald_Trump	0.538397

Enter word (EXIT to break): president

Word: president Position in vocabulary: 348

Word	Distance
President	0.800628
chairman	0.670875
vice_president	0.670023
chief_executive	0.669128
CEO	0.659013
pesident	0.626521
Vice_President	0.621666
executive	0.618248
prez	0.576191
Presdient	0.571838

Enter word (EXIT to break): candidate

Word: candidate Position in vocabulary: 1620

Word	Distance
candidates	0.794275
candiate	0.705062
Candidate	0.677797
challenger	0.628802
canidate	0.623805
candidacy	0.618346
candi_date	0.616838
nominee	0.590141
mayoral_candidate	0.589086
cadidate	0.587563

Enter word (EXIT to break): opinion

Word: opinion Position in vocabulary: 1966

Word	Distance
opinions	0.716355
opinon	0.633364
opnion	0.561680
Opinions	0.549686
opinons	0.549233

Opinion	0.541371
views	0.524808
viewpoint	0.524092
opionion	0.487257
veiws	0.469507

Question 4

We switch back to our corpus to find three linguistic regularities. Again the distance is normalized dot product.

Outcome For Our Test: *(Code: NLP/NLP 5.2/regularities.c)*

```
Enter three words (EXIT to break): president politician bad

Word: president   Position in vocabulary: 847

Word: politician   Position in vocabulary: 0

Out of dictionary word!


Enter three words (EXIT to break): president man bad

Word: president   Position in vocabulary: 847

Word: man         Position in vocabulary: 173

Word: bad         Position in vocabulary: 897

                              Word              Distance
------------------------------------------------------------------------
                             media                  0.622181


Enter three words (EXIT to break): candidate trump nice

Word: candidate  Position in vocabulary: 578

Word: trump  Position in vocabulary: 542

Word: nice  Position in vocabulary: 2321

                              Word              Distance
------------------------------------------------------------------------
                            allday                  0.689042
```

Enter three words (EXIT to break): election vote rich

Word: election Position in vocabulary: 1324

Word: vote Position in vocabulary: 175

Word: rich Position in vocabulary: 3511

Word	Distance
poll!	0.594140

5.3 Sentiment Dataset Construction

The manual annotation is done by hand to define the positive value and negative value of each sentence from the corpus. We need to analyze the crawled data and to determine the polarity value of them.

By using the Cohen's kappa coefficient, we can measures the inter-rate agreement from the sample. It calculates the score of the homogeneity or consensus of among given agreement and optimize the raters by human judges.

The Kappa value and Accuracy value forms the same simulated binary data. The value of the Accuracy is generated in direct proportion to the value of Kappa. The Accuracy value of the agreement is characterized:

- 0 – 0.2 as "slight";
- 0.21 – 0.4 as "fair";
- 0.41 – 0.6 as "moderate";
- 0.61 – 0.8 as "substantial";
- 0.81 – 1 as "almost perfect";

We define when the Kappa value is equal to "0", and the corresponding value of the accuracy is equal to "0.5".

Question 5

1. We set the first 100 texts as a subset "S1", median 100 texts as a subset "S2", last 100 texts as a subset "S3".

2. **The result of A1:**

	Yes	**No**
A	36	64

3. **The result of A2:**

	Yes	**No**
A	43	57
B	35	65

		A_2^{G2}	
		Yes	**No**
A_2^{G1}	**Yes**	5	30
	NO	38	27

4 & 5.

The Kappa value of the IAA:

- $K = (P_o - P_e) / (1 - P_e)$;

The observed proportionate agreement $P_o = (5+27) / 100 = 0.32$;

- A declares "Yes" is 0.43 of the time:
- B declares "Yes" is 0.35 of the time;
- The overall probability of random agreement

- $P_e = 0.43 * 0.35 + (1 – 0.43) * (1 – 0.35) = 0.521$;
- $I_1 = (0.32 – 0.521) / (1 – 0.521) = (-) 0.42$;

6.

The result of A3:

	Yes	**No**
A	37	63
B	40	60

		A_3^{G2}	
		Yes	**No**
A_3^{G1}	**Yes**	17	23
	No	20	40

The observed proportionate agreement $P_o = (17+40) / 100 = 0.57$;

- A declares “Yes” is 0.37 of the time:
- B declares “Yes” is 0.4 of the time;
- The overall probability of random agreement
- $P_e = 0.37 * 0.4 + (1 – 0.37) * (1 – 0.4) = 0.526$;
- $I_2 = (0.57 – 0.526) / (1 – 0.526) = 0.093$;

7.

- For S2, the kappa value of IAA is $I_1 = (-) 0.42$, the according accuracy value is around 0.2.
- For S3, the kappa value of IAA is $I_2 = 0.093$, the according accuracy value is around 0.5.

- Since I_1 is smaller than I_2, the agreement decision in S3 is more coherent than the S2's. In the other words, people will have higher rate of the same opinion (either both are positive or negative) in S3 compared with S2.

5.4 Building The Opinion-Mining Engine

Question 6

From the dataset created in section 5.3, we are able to establish the training and testing set for our model with 3-fold validation technique. We alternatively select 2 sets as the training set and the other as the testing set. We first apply POS tagging on all three sets with TextBlob (https://github.com/sloria/textblob-aptagger/tree/master), then use Weka (http://www.cs.waikato.ac.nz/ml/weka) to filter our strings with its StringToWordVector and subsequently build an SVM Classification model.

a) **S1 + S2 as training set and S3 as testing set:**
   - Precision 0.78
   - Recall 0.67
   - F-Measure 0.83

b) **S2 + S3 as training set and S1 as testing set:**
   - Precision 0.73
   - Recall 0.72
   - F-Measure 0.79

c) **S1 + S3 as training set and S2 as testing set:**
   - Precision 0.69
   - Recall 0.76
   - F-Measure 0.81

d) **Average:**
   - Precision: 0.73
   - Recall: 0.72
   - F-Measure: 0.91

Due to human errors, S1 may contain some annotation errors. We can compare the annotations by the SVM model trained with S2 + S3 when testing it on S1 with the annotations made by human to mark down the differences between them. Then we manually check again to fix possible errors according to these differences.

5.5 Using Deep Learning For Sentiment Analysis

Question 7

In this section we repeat the process in 5.4 with different sets. Each set is used to create a classification model with Weka and LibSVM, then has its performance tested with 3-fold cross validation. The first set is the 300-opnionated-document set in section 5.4, the second set is the same as the first set except all strings are filter with Google's deep learning library Word2vec *(Code: /5.4_5.5/NLP_5455/word2vec.c)*, and the last set is the set of 300 random strings from an online opinionated Twitter corpus (http://markahall.blogspot.sg/2012/03/sentiment-analysis-with-weka.html).

Performance of each set is as followed:

a) **Performance of the first set (section 5.3's 300 documents):**
   - Precision 0.78
   - Recall 0.67
   - F-Measure 0.83

b) **Performance of the second set (deep learning set):**
   - Precision 0.84
   - Recall 0.73
   - F-Measure 0.88

c) **Performance of the third set (random set):**
   - Precision 0.73
   - Recall 0.76
   - F-Measure 0.75

As seen from the results above, the model developed with the help of the deep learning tool Word2vec gives the best performance. The main reason is that by having its features extracted with Word2vec, the documents keep most of its significant

features while getting rid of insignificant ones. SVM also performs better with nominalized vectors instead of word vectors.

We can further increase the model performance by including some information from WordNet or Wikipedia as these knowledge bases provide valuable tags about the emotion expression level of words. A possible algorithm to include those tags is to assign weights to words in our corpus, with the weights be determined according to the levels of the tags.

5.6 Creativity

Since this part requires more ideas on deep-learning module for analysis of comments, our group has decided to use Emoticon Annotation to make categories of the corpus from Tweet and Reddit.

In our instance, we present a framework of lexicon, which contains disambiguated entity with a three-dimensional probability distribution as “negative”, “positive”, and “neutral” polarities. At the beginning, we crawled tweet comments and stored each document as a text file (*.txt format). We then develop a system that automatically detects the emotion associated to each document:

1. Percentage of people who feel happy of their candidates selection
2. Percentage of people who feel annoy of their candidates selection
3. Percentage of people who are not care about the selection

Text messages are classified based on their emotion and sentiment context. After collecting labeled tweets, we proceed onto data preprocessing.

First, we define our “Emoticon” features. The table below is the full list of emoticons that we were used in our system:

Table 1: Full List of Emotions Used for Emotion Analysis

Category	**Emoticons**
Happy Emoticons	':)', ';)', '=)', ':]', ':P', ':-P', ';P', ':D', ';D', ':>', ':3', ':-)', ';-)', ':^)', ':o)', ':~)', ';^)', ';o)', "':')", ':-D', ':->'
Sad Emoticons	':(', '=(', ':-(', ':^(', ':o(', ';^(', "':'(", ':-<'

Based on the emotional release, the results are illustrated in table 2:

Table 2: Summary of Attitudes

Negative Attitude	1201
Positive Attitude	1419
Neutral Attitude	4115
Total	**6735**

The Weka workbench is an organized collection of state-of-the-art machine learning algorithms and data preprocessing tools. The results in table 2 were converted to Weka format and a "Vote" classifier were used to cross-validate of our result. The fluctuant (both positive and negative) rate takes 38.9% possession.

In order to make a comparison, we have applied Sentiment Analysis as well, as shown in the table below.

Table 3: List of Vocabulary for Sentiment Analysis

Category	**Text**
Happy Text	'elated', 'overjoyed', 'enjoy', 'excited', 'proud', 'joyful', 'happy', 'blessed', 'amazing', 'wonderful', 'excellent', 'delighted', 'enthusiastic', 'calm', 'peace', 'silent', 'serene', 'consent', 'convince', 'satisfied', 'relax'
Sad Text	'nervous', 'anxious', 'tension', 'afraid', 'fearful', 'distress', 'stress', 'angry', 'annoy', 'tense', 'bother', 'disturb', 'irritate', 'mad', 'furious', 'sad', 'sorrow', 'hapless', 'fatigue', 'gloomy', 'miserable', 'hopeless', 'depress', 'fatigued', 'unhappy', 'laugh', 'not'

The table below shows the corresponding results.

Table 4: Results for Sentiment Analysis

Negative Attitude	748
Positive Attitude	158
Neutral Attitude	5829
Total	6735

As same process, the fluctuant (both positive and negative) rate takes 13.5% possession in sentiment text analysis. When comparing "Emoticons" and "Text", the system has a higher sentiment detection when it implements "Emoticons" features (38.9%) rather than "Text" features (13.5%).

Since the dataset we crawled is a sample (little size). We may enhance our emoticon detection to achieve further higher accuracy of data analysis. As further enhancement, emoji and emoticons can be classified into more emotion classes adopted by the various papers.

This is our video link.

https://www.youtube.com/watch?v=WOrVyYbJyE4

References

[1] Christy Loerzel, "What is Reddit and why should you care?" [Online] Available: http://www.symantec.com/connect/blogs/what-reddit-and-why-should-you-care [Accessed 29 Oct 2015]

[2] Reddit, "Reddit About", [Online] Available: https://www.reddit.com/about/ [Accessed 29 Oct 2015]

[3] Tech Target, "Twitter" [Online] Available: http://whatis.techtarget.com/definition/Twitter [Accessed 29 Oct 2015]

[4] Twitter, "Company Facts", [Online] Available: https://about.twitter.com/company [Accessed 29 Oct 2015]

[5] Alberto Nardelli and Tom Clark , "Meet the pollsters who are predicting the general election results" [Online] Available: http://www.theguardian.com/politics/2015/apr/08/meet-the-pollsters-predicting-general-election-results [Accessed 29 Oct 2015]

[6] LeighBureau, "Biography" [Online] Available: http://www.leighbureau.com/speakers/NSilver/ [Accessed 29 Oct 2015]

[7] Nate Silver, [Online] Available: http://fivethirtyeight.com/ [Accessed 29 Oct 2015]

[8] David Smith, "How Nate Silver won the election with Data Science" [Online] Available: http://blog.revolutionanalytics.com/2012/11/in-the-2012-election-data-science-was-the-winner.html [Accessed 29 Oct 2015]

[9] Dylan Byers, "Nate Silver: Polls are failing us" [Online] Available: [http://www.politico.com/blogs/media/2015/05/nate-silver-polls-are-failing-us-206799 [Accessed on 29 Oct 2015]

[10] Lei Shi, Neeraj Agarwal, Ankur Agrawal, Rahul Garg, Jacob Spoelstra, "Predicting US Primary Elections with Twitter" [Online] Available: http://snap.stanford.edu/social2012/papers/shi.pdf [Accessed 07 Nov 2015]

[11] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, Noah A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series" [Online] Available:

http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1536/1842/ [Accessed on 29 Oct 2015]

[12] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, Isabell M. Welpe, "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment" [Online] Available: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/download/1441/1852 [Accessed 07 Nov 2015]

[13] Daniel Gayo-Avello, Panagiotis T. Metaxa, Eni Mustafaraj "Limits of Electoral Predictions Using Twitter" [Online] Available: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2862/3254 [Accessed 07 Nov 2015]

[14] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, "The viability of web-derived polarity lexicons" [Online] Available: https://www.aclweb.org/anthology/N/N10/N10-1119.pdf [Accessed on 29 Oct]