

Abstract

GOAL: To Design and Build an Intelligent Opinion Analysis System to Predict the Outcome of the 2020 Presidential Election from Social Network

DESIGN OBJECTIVES:

- Real Time Big Data Stream Collection from a Social Media Twitter
- Big Data Manipulation with Semi-Structured Database System MongoDB
- Identify Key Opinion Features Expressed in Unstructured Text Corpus
- Measure Polarity of Opinion in Phrase in Text in Word Level and Context Level
- Develop Ranking Algorithm to Measure Polarity of Text at Document Level
- Develop Predictive Model through Training Machine Learning SVM Classifier

Introduction and Background

Challenges with Big Data:

- Huge Volume:** 7 Million Unstructured; Unformatted; Disorganized Tweets
 - Impractical to manually identify polarity
- Varying Subjectivity:** Meaning is not immediately clear
 - Ambiguous Emotional state, Sarcasm
- Fast-Growing Real-Time Data Streams**

Opinion Analysis Approach:

- Identify Sentiment Features at Token Level
- Identify and Associate Scores to Context-level Features, N-Grams
- Document Level Ranking using Sentiment Scores from the Extracted Feature Set

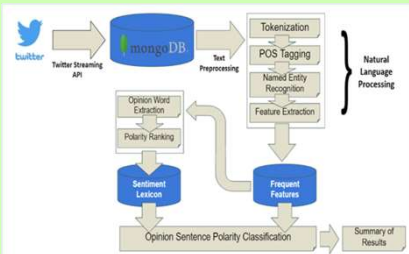


Fig 1: Comprehensive Opinion Analysis System Architecture

Building the Feature Set and Document Vectorization

Scores determined from two external opinion lexicons:

- SentiWordNet:** built from Princeton's WordNet lexical database; words grouped into sets of Cognitive Synonyms (Synsets) connected by similar polarity relations
- VADER:** Valence Aware Dictionary and sEntiment Reasoner; built to determine expressed sentiment in Social Media

Token Level

SentiWordNet

PageRank Algorithm:

Let $G = \langle N, L \rangle$,
 $N = \text{set of synsets}$
 $L = \text{set of directed links}$

Let $W = \text{row-normalized adjacency matrix of } G$

$$W[i, j] = \frac{1}{|F(i)|}, \quad \text{iff link from node } n_i \text{ to } n_j \\ W[i, j] = 0, \quad \text{otherwise}$$

Where $F(i) = \text{set of forward neighbors of } n_i$

Intuition: Scores from synset n_i propagate and are shared evenly among its forward neighbors

Synset score: $a_i^{(k)} \leftarrow \alpha \sum_{j \in B(i)} \frac{a_j^{(k-1)}}{|F(j)|} + (1 - \alpha) e_i$

Where: a_i represents degree of polarity of n_i

e_i is a constant used as an internal polarity source to seed synsets

$0 \leq \alpha \leq 1$ is a control parameter

Valence Aware Dictionary Reasoning

Lexical terms, Internet slang (e.g., 'sux'), emojis and sentiment-laden initialisms and acronyms (e.g., 'LOL', 'OMG') scored by average of 10 manual evaluators

Context Level

SentiWordNet

Polysemy Problem: one-word, multiple meanings

For each POS and sense associated with a term:

$$\text{pos(neg)Score} = \frac{\sum_{i=1}^n \text{pos(neg)Score}_i}{n} \\ f_m = \text{MAX}(|\text{posScore}|, |\text{negScore}|)$$

Context Dependencies: same nonzero pos/neg score for given word

Intuition: polarity may be determined by how often word appears among all pos/neg Tweets

$$\text{Score}_{\text{pos}}(w_i) = \frac{P(\text{pos}|w_i)}{P(\text{pos})} \quad P(\text{pos}|w_i) = \frac{\text{No. of } w_i \text{ in posTweets}}{\text{No. of } w_i \text{ in dataset}} \\ P(\text{pos}) = \frac{\text{No. of posTweets}}{\text{No. of all Tweets}}$$

Sigmoid normalization function: $\frac{2}{(1 + e^{-\text{Score}(w_i)})} - 1$

Valence Aware Dictionary Reasoning

Determine token Valence Score by applying module rules to preceding Boosters/Dampeners/Negators (e.g., 'really', 'sort of', 'not')

Compound Score: summation of polarity measure of each token updated with valence score

$$\text{Normalization Function} = \frac{\text{Score}_{\text{comp}}}{\sqrt{\text{Score}_{\text{comp}}^2 + \alpha}}$$

Ranking Algorithm and ML Classifier

3 Distinct Feature Sets Extracted:

Tweet Collection Size	Context Features Included?	Feature Set Size
7330058	No	54447
300K (150K per candidate)	No	17753
300K (150K per candidate)	Yes	21094 (Biden), 21332 (Trump)

Ranking Algorithm

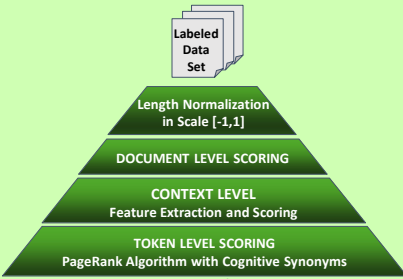


Fig 2: Document Vectorization for Ranking Algorithm

For each Tweet in candidate collection:

- Associate scores to tokens found in feature set
 - order precedence: trigrams > bigrams > unigrams
- Aggregate polarity at document level
- Normalization function: $\text{Score}_{\text{final}} = \frac{\text{total token score}}{\sqrt{\text{sum of squares of token scores}}}$
- Assign sentiment label: $\begin{cases} \text{pos,} & \text{if } \text{Score}_{\text{final}} > 0 \\ \text{neg,} & \text{if } \text{Score}_{\text{final}} < 0 \\ \text{neu,} & \text{if } \text{Score}_{\text{final}} = 0 \end{cases}$

Support Vector Machine Classifier

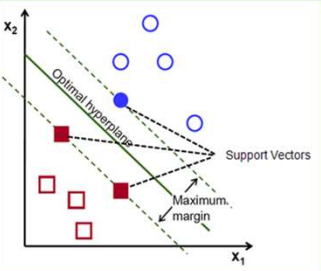


Fig 3: Support Vector Machine (SVM)

PURPOSE: accurately label Tweets in pos/neg classes based on input feature vectors

INPUT: Set of Tweets represented as feature vectors $x_i \in \mathbb{R}^n, i = 1, \dots, n$

GOAL: Determine weight vector w and bias value b such that: $\begin{cases} w^T x + b \geq 1 & \text{for positive} \\ w^T x + b \leq -1 & \text{for negative} \end{cases}$

Kernel Function: $K(x_i, x_j) = (x_i^T x_j + c)^d$
Where: x_i, x_j are input vectors
 c is a margin tuning constant
 d is the polynomial degree

Experimental Results

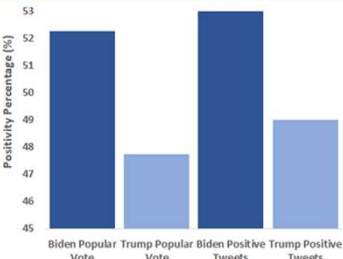


Fig 4: Comparison of 2020 Presidential Election Result vs. Ranking System Predicted Positivity

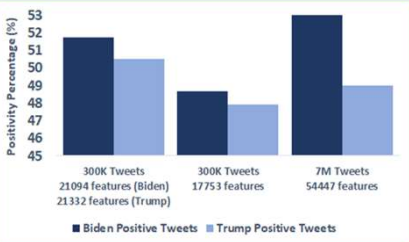


Fig 5: Positivity Prediction Comparisons by the Number of Tweets and the Number of Features

	Biden				Trump			
Accuracy	0.996646360036552				0.9972055435972962			
	Precision	Recall	f1-score	Support	Precision	Recall	f1-score	Support
Macro	0.9970	0.9897	0.9933	65660	0.9977	0.9913	0.9945	26481
Weighted Average	0.9967	0.9967	0.9966	65660	0.9972	0.9972	0.9972	26481
POS	0.9965	0.9991	0.9978	33929	0.9968	0.9996	0.9982	13633
NEG	0.9966	0.9985	0.9975	26495	0.9974	0.9982	0.9978	10834
NEU	0.9978	0.9715	0.9845	5236	0.9990	0.9762	0.9874	2014

Table 1: Accuracy of SVM Classifier ($c = 1.0, d = 6$)

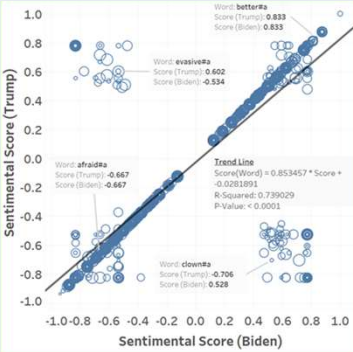


Fig 6: Comparison of Sentiment Word Scores Between Candidate Feature Sets

Conclusion

- Developed Opinion Analysis System correctly predicts presidential election results within 1.5% of Biden's and 1.8% of Trump's popular vote respectively
- SVM ML Classifier with Polynomial Kernel Function Able to Verify Ranking Results with 99% Accuracy
- Approach is an Effective Robust Means of Measuring Polarity in text for election prediction as well in any other text documents
- Most Accurate Results Achieved through:
 - Larger Tweet Data Set combined with
 - Comprehensive Sentiment Feature Identification

Challenges/Future Work

Limitations of Approach:

- Sarcasm not inferable through aggregation method
- Deep Learning ML Approach for Context Learning
- Memory limitations exponentiated working with large data set; Multiple Batch Processing; Extended to Distributed Parallel Processing
- Impact of hashtags need to be calculated
- Isolating polarity given multiple focuses of discussion; need to determine further syntactic dependencies