A Framework for Analyzing Veracity in Social Media

By

Nagaraju Vadranam

**Committee Members**

Dr. K.M. George(Advisor)

Dr. Johnson P Thomas

Dr. Esra Akbas

*Department of Computer Science, Oklahoma State University.*

*Stillwater, OK, USA*

**

# ABSTRACT

As social media become a common medium for information sharing, the veracity of the contents become a serious issue. Quality and reliability of information found in social media has been studied by several researchers. This is a relatively new area of research and a complex problem. One set of solution may not work in all cases. This paper presents a framework for veracity estimation. The framework is implemented and tested using Twitter data. The approach is to construct labeled data from tweets and use supervised learning to build predictive models. The framework is tested on tweets collected on different topic areas. Both conventional and fuzzy clustering techniques were tested

TABLE OF CONTENTS

Sections Page

1. [INTRODUCTION 1](#_TOC_250003)
2. [RELATED WORK](#_TOC_250002) 2
3. [DATA COLLECTION](#_TOC_250001) 4
4. VERACITY FRAMEWORK 5
5. EXPERIMENTS 8
6. COMPARISONS 10
7. CONCLUSION 16

[REFERENCES 1](#_TOC_250000)8

# INTRODUCTION

According to (August 8, 2018) Statista Infographics report, the combined monthly active user count for the social media platforms Instagram. Facebook, WhatsApp, Snapchat, and Twitter is more than 752 million users. Twitter alone has more than 9 million active users for the month. Thus, data generated by social media fall into the big data domain. These numbers represent the magnitude of the social media information domain and the potential for abuse. Current news stories validate the misuse of Facebook for political purposes during the 2016 Presidential election. So, veracity of information found or propagated through social media is a major concern. Several research projects to detect rumors and ascertain veracity of information were reported in the literature. While critiquing several recent work, Soroush Vosoughi [19] suggest that veracity prediction on social media is a new field. According to other authors, the plentiful and diverse data provide new opportunities in big-data. At the same time, it has raised questions about blindly trusting processes and content such as data collection, pre-processing, storage, data quality, nodes which store data, and cloud services [16]. These problems give the context to our work. While veracity has to be addressed from different angles, our focus is content veracity, so we design and implement a veracity detection framework for social media content.

This paper presents a veracity estimation framework that we developed using ideas from the areas of natural language processing, clustering, and neural network. In this work, we focus on the veracity of tweets. Two approaches are followed: 1) rate the tweets as veracity positive, negative, or neutral. 2) assign to tweets a veracity score in the interval (0, 1). For the first approach we combine clustering and supervised learning algorithms and for the second approach we use fuzzy clustering. We employ fuzzy c-means clustering algorithm [2] and nonnegative matrix factorization [9] as clustering algorithms. A concept level description of framework is provided in section III. Results of experiments are described in section IV. Section V provides conclusion and direction of future work

# II Related Work:

In this section we provide synopsis of papers that are closely related to our approach in some form. As rumor detection is very closely aligned with veracity detection, we review several papers in that subject. [21] provides a survey of papers that have studied rumors in social media ranging over history, psychological studies, factors that determine the diffusion of rumors etc. Social media platforms such as Twitter are more and more beings used as platforms for news about breaking stories and other events. These platforms are also used for propagation of false information or fake news. This paper provides a rumor classification system architecture with four components, namely rumor detection, rumor tracking, stance classification, and veracity classification. For each component, authors also describe approaches to track it. In their conclusions, they emphasize that more work need to be done in this area. According to Goel and Uzuner [8], “Academic research on fraud detection suggests that detecting fraud is a complex problem and no one set of predictors will be always successful in fraud detection. This may be partly due to the fact that once the fraud indicators are publicly known, companies can ﬁnd ways to outsmart them and ﬁnd other creative ways to conceal fraud.” They analyze annual reports of companies to identify fraudulent reporting. They use qualitative predictors for fraud detection. They report that use of both positive and negative sentiment is more pronounced in fraudulent reports. Their study suggests that fraudulent reporting contains higher sentiment content than truthful ones. Kwon et. al. [9] claim as one of the first papers on analysis of rumor propagation in social media. Their approach identifies rumors based on “temporal, structural, and linguistic properties of rumor propagation”. They built classifiers based on decision tree, random forest, and SVM to classify at topic as rumor or non-rumor by analyzing related tweets. Giasemidis et.al. [6] approach rumor identification as a supervised binary classification problem. Our approach has similarity with some of their ideas. They defines features and aggregate at the rumor level where as we associate features to each individual tweet. They also used Linguistic Inquiry and Word Count (LIWC) to obtain linguistic characteristics and sentiment. In [12], the authors use Convolutional Neural Networks (CNNs) to learn hidden representations of rumors. They present a single tweet credibility model. In previous work [3, 4], the authors focus on the credibility of information propagated through Twitter. They use crowdsourcing to create a labeled dataset of newsworthy events based on tweets. The labeled dataset is used to train two automatic classifiers. They have used the 2010 Chilean earthquake data as test case. The same authors in an earlier paper addressed the topic of information credibility on Twitter. They focus on real world emergencies. They trained a supervised classifier to automatically find newsworthy topics. They use several features for training

There are several papers that study veracity of tweets. In [14], the authors propose and analyze a veracity model based on entropy and topic modeling. In [18], the authors propose and evaluate three formulae for assessing veracity of tweets. Lukoianova and Rubin [11] study veracity in big data across three main dimensions: objectivity/subjectivity, truthfulness/deception, and credibility/implausibility. The three dimensions are combined to provide a veracity index.

Available tools are used for computation. Nurse et. al. [13] provide visualization tool for displaying trust measures computed by the system. Their experiments suggest that users prefer to know how the trust score is computed by the system. Derczynski, et. al. [5] state that analyzing and determining veracity of social media content has received increased attention from the field of natural language processing. Advanced systems and annotation schemes have been developed for rumor analysis support. They conduct an exercise in SemEval conference. The authors “propose a shared task where participants analyze rumors in the form of claims made in user-generated content, and where users respond to one another within conversations attempting to resolve the veracity of the rumor.” They “ran a task in SemEval consisting of two subtasks: (a) stance classification towards rumors, and (b) veracity classification. Subtask A corresponds to the core problem in crowd response analysis when using discourse around claims to verify or disprove them. Subtask B corresponds to the AI-hard task of assessing directly whether or not a claim is false.” While they show good results based on the participations, admit that finding out accurately the truth of a story.

Other closely related work is personality assessment based on tweets. Ahmad and Siddique [1] use keywords – dominance, influence, steadiness, and compliance – for data collection from tweets. Text mining and sentiment analysis were done on the tweets. The conclusions are not definitive. Tandera, et. al. [17] build a system to predict personality based on the Big Five personality model (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism). Machine learning algorithms are used and Facebook data are used for training.

# III.DATA COLLECTION

Data is collected from twitter through twitter Application programming interface(API). The data is stored in HDFS and the data is collected through a variety of topics such as Peace talks topic happened in Singapore between president trump and north Korean supreme leader Kim Jong un, NRA (national Rifle Association regarding gun control and school shootings), Democrats and the Mueller report (regarding the collusion and obstruction of justice)

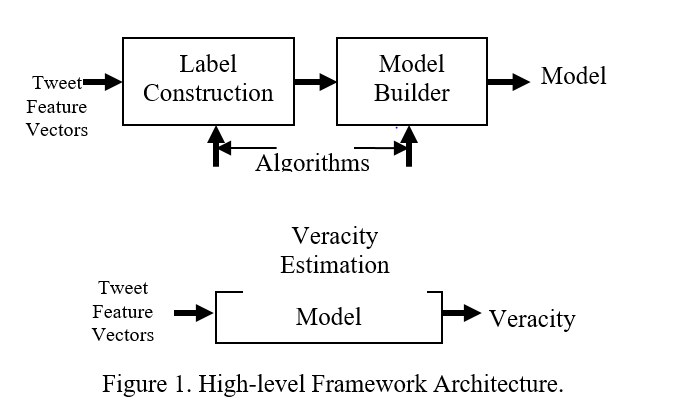
All this data collection process is done by using Apache Flume in Hadoop cluster. Apache flume is an open-source software where a webserver generates log data and this data is collected by an agent (Twitter Agent) in Flume. The channel buffers this data into a sink, which finally pushes it to centralized stores. The process of collecting data is mentioned in the following steps,

* Create a twitter account by signing up.
* Go to <https://developer.twitter.com/>and apply for developer account.
* After it is approved then collect consumer secret, consumer token and access token.
* There are three components for a twitter agent, namely source, sink and channel.
* Flume connects to twitter API and receives data in JSON format and stored in the HDFS.
* Add the flume source to the flume class-path.
* Now, create a configuration file for the flume agent the specifying the consumer key, consumer secret, access token and access token secret and keywords, hdfs sink path.

Data is collected in JSON format which contains key value pairs for various fields for which we are using for out analyses and data modelling.

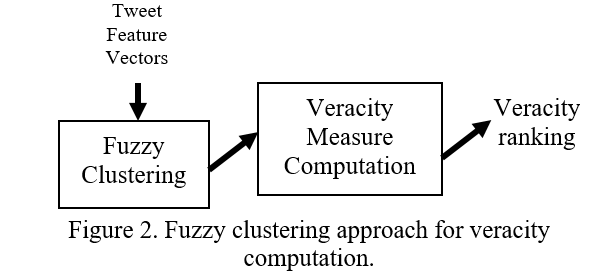
# IV. Veracity Framework

In this section we define the framework at the concept level. For this purpose, we assume that a set of k features are associated to each tweet depending on the veracity analysis domain and these features map tweets into a k-dimensional space. As mentioned in the introduction, we tested two approaches. As stated given a tweet, the first approach attempts to make a determination of it as positive, neutral, or negative. The basic framework’s high-level architecture is shown in Figure 1. It consists of two parts. The first part constructs a veracity estimation model and the second part predicts the veracity from the features input. Common to both parts is feature vector construction module.



The label construction part classifies the tweet feature vectors (and by implication the tweets) into three clusters. Each cluster is assigned a label 0, 1, or 2 to obtain a labeled set of data. Then the labeled data is used to train a supervised learning algorithm which becomes a veracity estimation model. The model can be used to estimate the veracity of tweets (possibly in real-time). To test our framework implementation, we have used k-means algorithm for clustering and ANN for prediction.

The second approach that we followed is fuzzy clustering. The motivation is to avoid the pitfalls in assigning definitive statements of positive, neutral, or negative. In this approach we assign a veracity measure between 0 and 1 to tweets. We have tested c-means clustering and nonnegative matrix factorization. The basic framework is shown in Figure 2.



.

The idea is to classify the feature vectors (training set) using fuzzy clustering algorithms into three clusters. Choose the cluster that is closely associated to veracity positive tweets (currently this is a manual processes). Choose the fuzzy membership value of tweets as veracity measures of tweets. The membership value is a number in the interval [0, 1]. The process so far is similar to model building in the first approach. This also gives a ranking of tweets in the training set which can be considered as the model. To compute the veracity of a tweet, construct the feature vector and determine the closest one above and below in the model using a chosen similarity measure (there are several similarity measures available in the literature). The veracity measure of the model is chosen as the average of the measures of the two closest tweets. For this approach, we have used fuzzy c-means algorithm [2, 20] and nonnegative matrix factorization algorithms which are explained below.

## Fuzzy clustering

Conventional or hard clustering algorithms classify a set into disjoint subsets were each element belong to exactly one subset. In the case of fuzzy clustering, an element belongs to every subset with a degree of membership. Given a set X and x ϵ X, [20] gives the definition of a fuzzy c-clustering μ = (μ1, …, μc) where μi(x) are functions assuming values in [0, 1] and μ1(x)+…+μc(x) = 1. For the fuzzy c-means clustering algorithm, we refer the reader to [2].

## Nonnegative Matrix Factorization (NMF)

Nonnegative matrix factorization [7] is a special case of linear dimensionality reduction (LDR) which has become a popular method for dimension reduction in applications. LDR is equivalent to approximation of a m-by-n matrix X with a low-rank product WH, X ≈ WH, where W is a m-by-k matrix and H is a k-by-n matrix. The matrix X represents n data points in an m dimensional space. The columns of W are basis elements and the columns of H are coordinates in the basis W. W and H are computed such that the Frobenius norm of X-WH is minimized. NMF is a special case of LDR where X, W, and H are non-negative matrices.

NMF has been used for clustering or topic modeling in the context of text analytics. We adapt the approach as a fuzzy clustering method for tweets. In order to do this, we need to define the function μ = (μ1, μ2, μ3). For this purpose, we view X as a n-by-m matrix where the rows represent points (total n) in the feature space. Columns represent feature values of the tweets which are represented as points. Feature values are computed as positive numbers. Therefore, X is a nonnegative matrix. The fuzzy clustering is shown in Algorithm 1 below:

**Algorithm 1**: *Fuzzy clustering*

Step 1: Compute the NMF of X, X ≈ WH, where W is n-by-3 matrix and H is a 3-by-m matrix. Each row of W represents a tweet.

Step 2: Let each column of W be a cluster.

Step 3: Normalize the rows of W so that the sum of the elements is 1.

Step 4. The columns of the normalized W be the three clusters, and the row values are the membership functions μ1, μ2, μ3.

In the next section, we present experiments and results of the experiments.

# V. Experiments

We implemented the framework described in the previous section in Python. We have used several tools available in python. At present the interfaces are via command line scripts from the Linux prompt. To test the implementation, we used two sets of tweets collected using Apache Flume. The tweet topic areas are 1) the North Korean peace talk and 2) NRA (National Rifle Association). Peace talk data were collected from May 31 to July 27. Of the 8990152 tweets collected, 525478 were identified as original tweets and considered for our experiments. NRA dataset was collected during the period June 7 to August 3 (due to Flume related issues, some data was missing during that period, however this should not affect our experiments as time period is not a parameter.) Of the 1942518 tweets collected, 98920 were original tweets and used for our experiments. The Python calls *clean(owner\_tweet\_text)* and *Counter.\_\_call\_\_(clean(ownertweet*)) were used in data preparation. We associated five features with the tweets. Thus the tweets are mapped to a five dimensional space. The five features and the justification for choosing them are shown in Table 1. The considerations in choosing the features are based on our own observation of tweets and information from the literature. We observed from the tweets of very influential persons, that retweeters who are not followers of original tweeter tend to be more objective. The quote “Our results show that fraudulent MD&As on average contain three times more positive sentiment and four times more negative sentiment compared with truthful MD&As” [8] provides the reason for measuring positive and negative sentiments as two features contributing to veracity of tweets. The fourth feature we used is the weight computed by the Gensim package. The idea is adaptation of the personality assessment using words [1]. The fifth feature we used is entropy as it is measure of uncertainty. For each tweet entropy is computed using Shannon’s entropy formula [15]:

Distribution of words in the tweet is considered as the probability distribution.

Table 1. Feature Description.

| Feature | Name | Rationale |
| --- | --- | --- |
| V1 | Non-follower retweet count | Independence of retweets. |
| V2 | Positive sentiment | Positive and negative sentiment of fraudulent tweets are higher than truthful tweets [8]. |
| V3 | Negative sentiment |
| V4 | Word weight | Sum of weights of words occurring in a tweet distinguish user traits? |
| V5 | Entropy | Measure of uncertainty implicit in statements. |

## Computation of features

To compute features, we considered only original tweets. AFINN database was used for sentiment score computation. The five feature computations are done as follows:

### V1 = if (retweet count > follower count) then retweet count – follower count; else 0.

### V2 = score = get\_afinn\_scores(Owner\_tweet\_text); int(score['positive']).

### V3 = score = get\_afinn\_scores(Owner\_tweet\_text); int(score['negative']).

### V4 = sum of weights of words in the tweet

### V5 = entropy computed by the formula (1)

### 

We have performed several experiments with normalized and un-normalized values of the features V1-V5. If data is not normalized, then the features whose computed values are high seem to dominate the clusters computed by k=means algorithm. Hence, the results presented here are based on normalized data except for nonnegative matrix factorization. Normalization introduces negative coordinates which will affect factorization.

**Results:**

The Table 2 below shows the summary of results of our experiments. Some other data are given in the appendix. We used three clusters to classify tweets as veracity, positive, negative and neutral. The five features form the five coordinates of the points representing tweets. The steps involved in the experiment were:

### Data cleaning

### Feature construction

### Selection and application of clustering algorithm

### Label the clusters and use it for training an ANN model

### Validate using data not used in training

### The clusters were manually inspected by randomly choosing tweets used for training.

Table 2. Training and validation results.

|  |  |  |
| --- | --- | --- |
| Clustering Algorithm | Accuracy of model fit | Accuracy of outside testing |
| K-means | 98% | 93% |
| Fuzzy C-means | 72% | 55% |

### 

## VI. Comparisons

## In order to see the validity of proposed model we have chosen to compare with already established models. we use the following models for comparisons:

## OTC (Objective, Truthful and Credible) model.

## Text Classification using Convolution Neural Networks.

## OTC Model Summary:

## 

## In [22] it is argued that the Big Data can have different characteristics which effects the quality of the useful information depending on various factors such as processing and data collection and Big Data tend to have biases, ambiguities and inaccuracies.so to mitigate this problem authors of [22] chose to represent data in three dimensions (veracity Dimensions) namely

## 1)Objectivity/Subjectivity

## 2)Truthfulness/Deception

## 3)Credibility/Implausibility

## They chose to combine the measures of veracity dimensions into one composite index, which lies between 0 and 1. where 0 – being biased and 1 – being truthful.

## Tools Used:

## 1)Text Blob (python Package) for Objectivity /Subjectivity measure.

## 2)Empath (python Package) for Truthfulness/Deception measure.

## Model Implementation:

## In this section we define the model implementation at the concept level. For this purpose, we assume that a 3 Veracity measures are to be calculated for each tweet and then a Veracity Composite Index is then calculated for each tweet as mentioned in the model summary. To start with we calculate the three veracity dimensional measures as below:

## 1)Objectivity /Subjectivity Measure:

## The Objectivity/ Subjectivity measure is calculated by using TextBlob which is a python package free to download from python standard repository. It takes in user tweet text as a parameter for TextBlob() constructor call and provide the sentiment Object which then can be used to calculate the subjectivity/Objectivity of the tweet which also lies between 0 and 1

## 

## EX: *sentiment Object = TextBlob(Owner\_tweet\_text). sentiment*

## *Objective Score= (1 - sentimentObject.subjectivity)*

## 2)Truthfulness/Deception Measure:

## The Truthfulness/Deception measure is calculated using the Empath package which analyze the tweet for deceptive contents and gives back a key valued pair of personality traits for that tweet.

## It has specifically assigned a category measure for deception in a tweet .so the general idea is if a tweet is not deceptive it tends to be on a truthful note.

## *EX: Truth Measure= (1 – Empath (). analyze(Owner\_tweet\_text, normalize=True) [‘deception’])*

## 3) Credibility/Implausibility Measure:

## The process of finding out how credible a tweet is as specified in [22] using the process called Mutual information(MI) basically by getting the word association strength the Mutual Information between two words word1, word2 is calculated by

## 

## /p(word1) p(word2))

## Calculating veracity Index:

## 

## To get a veracity composite index we had summed up above 3 veracities dimensions’ measures and divide by 3.

## Comparison - I:

## To compare this results with the results that we already had with our proposed model we need to cluster the tweets based on veracity composite index into 3 clusters to compare against the k-means generate labels.

## The OTC model was run on the same data sets The National Rifles Association (NRA) data set and Trump & North Korea peace talks data set (Peace Talks) and also an additional dataset on democrats (Collusion and Obstruction of justice dataset) a subset (owner tweets) of 18549 tweets

## Results:

## The below table contains 12063 tweets clustered into 3 using k-means and same tweets were made into 3 quarters to see how the veracity composite index is split among the clusters. The K-means labels [ 0, 1 ,2] are [positive,neutral,biased] respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **NRA Data** | | |
| K-means labels | **0 (positive)** | **1(neutral)** | **2(biased)** |
| **Q1** | # Tweets | 107 | 1538 | 2830 |
| Veracity Index Range | 0.3333-0.50699 | 0.050355-0.50777 | 0.00034-0.50772 |
| **Q2** | # Tweets | 98 | 1587 | 2793 |
| Veracity Index Range | 0.50804-0.54516 | 0.50783-0.5455 | 0.50786-0.54553 |
| **Q3** | # Tweets | 91 | 1571 | 2816 |
| Veracity Index Range | 0.545644-0.6166667 | 0.545567-0.6667 | 0.54554-0.6667 |

## The NRA data The biased tweets are comparatively mode than the positive, neutral tweets combined a similar test is run on Democrats data using both proposed model and implemented OTC model on subset of total (owner tweets) of 18549 tweets for comparison

## Then we divided these tweets into three cluster to try to correlate with the k-means clusters that was generated using the proposed veracity model and Veracity composite index

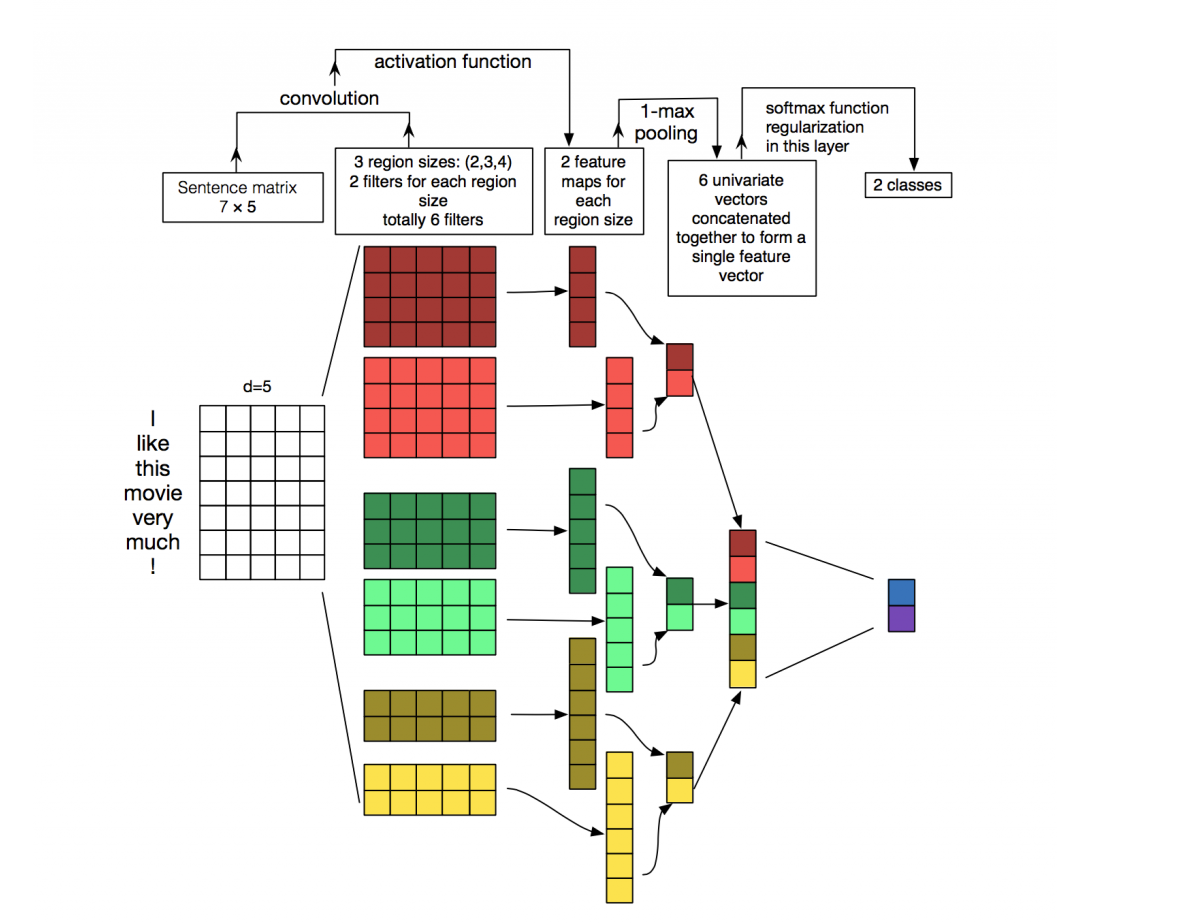
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Democrats Data** | | |
|  | K-means labels | **0 (positive)** | **1(neutral)** | **2(biased)** |
| **Q1** | # Tweets | 159 | 2113 | 3910 |
| Veracity Index Range | 0.333333-0.5129 | 0.000133-0.51291 | 0.003-0.51291 |
| **Q2** | # Tweets | 167 | 2046 | 3967 |
| Veracity Index Range | 0.51294-0.5486 | 0.5129-0.54868 | 0.5129-0.5486 |
| **Q3** | # Tweets | 166 | 2133 | 3884 |
| Veracity Index Range | 0.548782-0.66666 | 0.54872-0.66666 | 0.54868-0.6667 |

## 

1. **Text Classification using Convolution Neural Networks:**

A convolutional neural network(CNN) has an input layer and a n output layer and contains multiple hidden layers in between.in CNN every layer purpose is to learn a distinct small feature form the whole sentence structure to identify stances and lexical occurrences of different classes. there are three different layers in CNN Convolution layer, max pooling layer and fully connected layer.

NLP tasks are sentences or documents are typically represented as a matrix. Each row of the matrix corresponds to one token, typically a word, but it could be a character. That is, each row is vector that represents a word. Typically, these vectors are word embeddings (low-dimensional representations) like word2vec or GloVe, but they could also be one-hot vectors that index the word into a vocabulary. For a 10-word sentence using a 256-dimensional embedding we would have a 10×256 matrix as our input.



**Fig:** illustration for sentence classification of a convolutional neural network

But in our case we have used the word2vec vectors for each word averaging them to form a vector for a tweet.

Here are some of the hyper parameters that the model is using

**# Model Hyper parameters**

*tf.flags.DEFINE\_integer("embedding\_dim", 256, "Dimensionality of character embedding (default: 128)")*

*tf.flags.DEFINE\_string("filter\_sizes", "3,5,7,9 ", "Comma-separated filter sizes (default: '3,4,5')")*

*tf.flags.DEFINE\_integer("num\_filters", 300, "Number of filters per filter size (default: 128)")*

*tf.flags.DEFINE\_float("dropout\_keep\_prob", 0.25, "Dropout keep probability (default: 0.5)")*

*tf.flags.DEFINE\_float("l2\_reg\_lambda", 0.01, "L2 regularization lambda (default: 0.0)")*

**# Training parameters**

*tf.flags.DEFINE\_integer("batch\_size", 128, "Batch Size (default: 64)")*

*tf.flags.DEFINE\_integer("num\_epochs", 500, "Number of training epochs (default: 200)")*

*tf.flags.DEFINE\_integer("evaluate\_every", 100, "Evaluate model on dev set after this many steps (default: 100)")*

*tf.flags.DEFINE\_integer("checkpoint\_every", 100, "Save model after this many steps (default: 100)")*

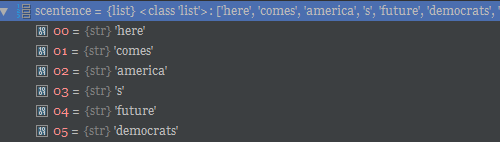
*tf.flags.DEFINE\_integer("num\_checkpoints", 5, "Number of checkpoints to store (default: 5)")*

in our model the maxpooling layer is looking for three feature maps for each region and softmax regularization layer will output three classes (namely truthful, neutral and biased) encoded into model as following labels [1,0,0] -- biased, [0,0,1] -- >truthful and [0,1,0]-neutral

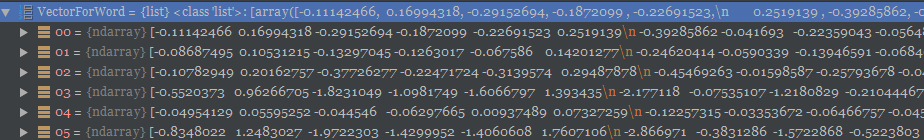
Word2vec Embedding:

For a given set of words, the word2vec library will generate a word vector for each word from pertained neural network model

For example, for the tweet1:

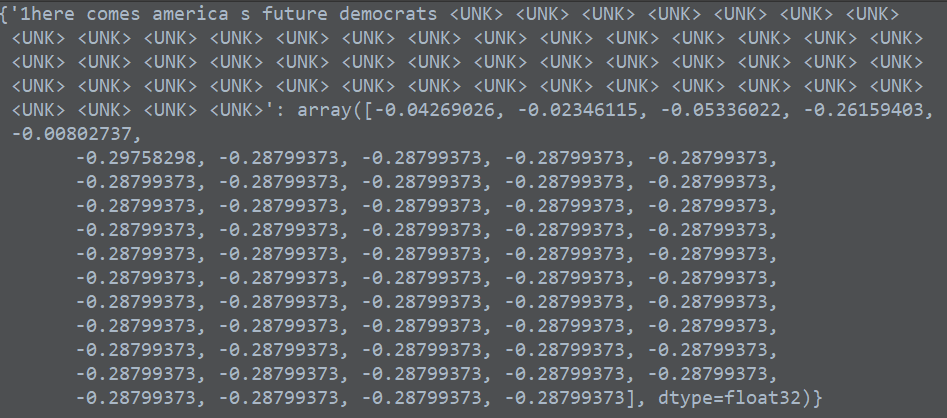


The word2vec Word vectors as follows:

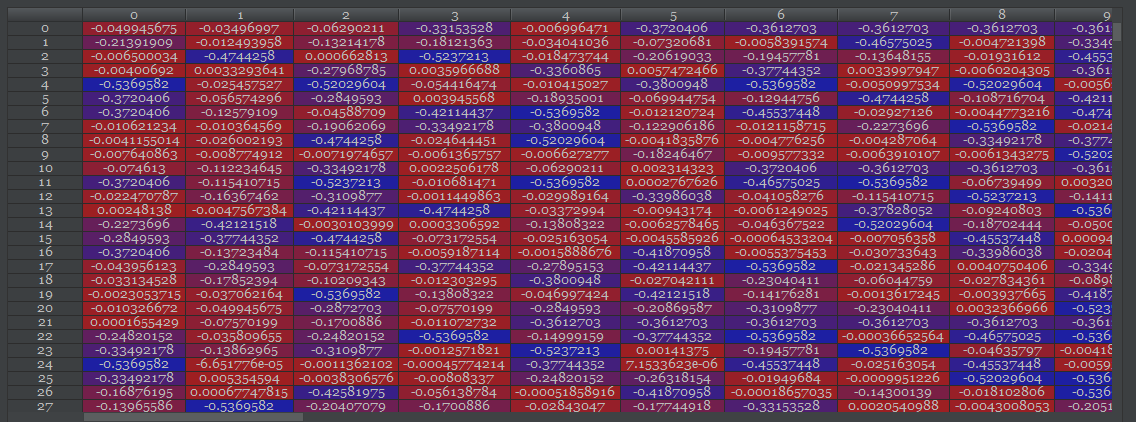


To get the sentence(tweet) vector (for tweet) we averages (np.average(np.asarray(VectorForWord), axis=1) these word vectors to form a sentence vector

Which looks like this



For every tweet it looks like this. Where each row represents a tweet



**Training the model:**

A relatively small dataset of 3 clusters which consists of truthful tweets, biased tweet, neutral tweets are gathered with manual filtering and labeling of tweets and clustered tweets from OTC model to get the training data, the training dataset has 1104 tweets in total which was taken from Democrats tweets data set.

**Testing the model:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DataSet** | **Democrats** | **NRA** | **Peace Talks** | **Collusion** | **Syrian** |
| **Truth** | **1498** | **1364** | **2554** | **5410** | **6368** |
| **Biased** | **10154** | **9916** | **8288** | **33772** | **22293** |
| **Neutral** | **7749** | **6454** | **17927** | **24787** | **24054** |
| **Total(tweets)** | **19401** | **17734** | **28769** | **63969** | **52715** |

The model Produce the Accuracy of 71~75 depending on data set where. the model predicted with a 75.64% accuracy for democrats and NRA datasets but for Collusion dataset the prediction accuracy percentage is 73% and for the peace talks and Syrian data set the prediction accuracy is 71%.

# VII.CONCLUSION

In this paper we report a framework that we designed and implemented to estimate the veracity of social media data. As social media usage becomes more and more mainstream, and individuals find friends in the social media, veracity of information presented is an important concern for individuals and society. Our approach combines several ideas including sentiment analysis, clustering, and machine learning. Our tests of the implemented system with a traditional clustering algorithm yields good result. We also tried fuzzy clustering algorithms. Fuzzy clustering does not perform well. However, it is a useful tool to rank tweets in a veracity scale. Illustrative data and results are given in the appendices.

In order to make the system more user-friendly, a better interface is needed. Design and implementation of a GUI capable of streaming data and performing real-time classification and prediction is considered future work.

##### **APPENDIX - I**

*Example tweets from peace talks*

### Classified as neutral by K-means algorithm

@john\_sipher @BrendaMackay13 Trump-Putin date leaves other extremists feeling empty! â€œMaybe we havenâ€™t done anything to directly undermine their democracyâ€” I get that, â€ Abu Bakr al-Baghdadi, the leader of isis, said. â€œBut weâ€™ve been nemeses of America for years, and that ought to be worth something.â€

### Classified as negative by K-means algorithm

@krassenstein Not tying to fear-monger but while weâ€™re all distracted, bigger story is Intel Community is warning us about a Cyber 9/11. This could be much worse than â€œelection meddlingâ€ Notice what tRump admin is doing about this, theyâ€™re reversing policies that were meant 2 protect US. Why?

### Classified as positive by K-means algorithm

@boreskes @RepMaxineWaters @Oathkeepers Congrats on giving Trump a second term. If you donâ€™t like the flag; leave.

*Example tweets from NRA related tweets*

### Classified as neutral by K-means algorithm

@nicktiedo\_251 @NRA Pssst Criminals donâ€™t obey laws. Guns donâ€™t shoot themselves. The NRA is not the problem. People are.

### Classified as negative by K-means algorithm

@MikeBoyyyyy24 â€œWhatâ€™s the second amendment?â€

### Classified as positive by K-means algorithm

@PatrickSvitek @tedcruz @DineshDSouza @NRA Yet another reason to vote Beto O'Rourke in Texas!

### Predicted as negative

They are the brave. @NRA @DLoesch @GOP you are the stupid. Listen to those that protect our democracy. #BanAssaultWeapons #NeverAgain #NoRA #GunControlNow https://t.co/sV6CA4yjeL

**APPRENDIX - II**

Table 3. Comparison of crisp classification and fuzzy ranking.

|  |  |  |
| --- | --- | --- |
| K-means Labels | Fuzzy ranking | Fuzzy label |
| \*2 | 0.595962 | 1 |
| 2 | 0.334069 | 1 |
| 2 | 0.340504 | 2 |
| 2 | 0.389601 | 1 |
| 2 | 0.338638 | 2 |
| 2 | 0.34946 | 1 |
| 2 | 0.413092 | 1 |
| 2 | 0.406084 | 1 |

\*Tweet for two 1: @realDonaldTrump Oh, itâ€™s ok to politicize the CA wildfires while people lose everything they have and even their lives. But after a school shooting, itâ€™s too soon to talk about gun control. Americans see right through you. Stop OBSTRUCTING progress. @Emma4Change @cameron\_kasky @JerryBrownGov

##### References

1. Nadeem Ahmad and Jawaid Siddique, “Personality Assessment using Twitter Tweets”, Proceedings of the 21st International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017,6-8 September 2017,
2. James C. Bezdek, Robert Ehrlich, and William Full, “FCM: The Fuzzy C-Means Clustering Algorithm”, Computers & Geosciences, Vol. 10, No. 2-3, 1984, pp. 191-203.
3. Carlos Castillo, Marcelo Mendoza, and Barbara Poblete, “Information Credibility on Twitter”, Proceedings of WWW 2011, March 28–April 1, 2011, Hyderabad, India, pp. 675-684,
4. Carlos Castillo, Marcelo Mendoza, and Barbara Poblete, “Predicting information credibility in time-sensitive social media”, Internet Research Vol. 23, No. 5, 2013, pp. 560-588, DOI 10.1108/IntR-05-2012-0095.
5. Leon Derczynski, et. al., “SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours”, Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), Vancouver, Canada, August 3 - 4, 2017, pp. 69-76.
6. G. Giasemidis, et al. (2016) Determining the Veracity of Rumours on Twitter. In: Spiro E., Ahn YY. (eds) Social Informatics. SocInfo 2016. Lecture Notes in Computer Science, vol 10046. Springer, Cham, pp. 185-205.
7. Nicolas Gillis, “The Why and How of Nonnegative Matrix Factorization”, arXiv:1401.5226v2, Mar 2014.
8. Sunita Goel and Ozlem Uzuner, “Do Sentiments Matter in Fraud Detection? Estimating Semantic Orientation of Annual Reports”, Intelligent Systems in Accounting, Finance and Management, Volume 23, Issue 3, Special Issue in Accounting, Auditing and Finance Applications, July/September 2016, pp. 155-254
9. Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang “Prominent features of rumor propagation in online social media”, Proceedings of the 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 1103–11083.
10. Daniel D. Lee and H. Sebastian Seung, “Algorithms for non-negative matrix factorization”, Proceedings of the 13th International Conference on Neural Information Processing Systems, Denver, CO, 2000, pp. 535-541.
11. Tatiana Lukoianova and Victoria L. Rubin, “Veracity Roadmap: Is Big Data Objective, Truthful and Credible?”, Advances In Classification Research Online, 24(1). doi:10.7152/acro.v24i1.14671
12. Tu Ngoc Nguyen, Cheng Li, and Claudia Niederée, “On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners”, Proceedings of the Social Informatics: 9th International Conference, SocInfo 2017, Part 2, pp. 141-158.
13. Jason RC Nurse, Ioannis Agrafiotis, Michael Goldsmith, Sadie Creese, and Koen Lamberts, “Two sides of the coin: measuring and communicating the trustworthiness of online information”, Journal of Trust Management 2014, 1:5.
14. Jyotsna Paryani, Ashwin Kumar T.K., and K. M. George, “Entropy-Based Model for Estimating Veracity of Topics from Tweets”, Proceedings of the Computational Collective Intelligence: 9th International Conference, ICCCI 2017, Nicosia, Cyprus, September 27-29, 2017.
15. C. E. Shannon, “A Mathematical Theory of Communication”, The Bell System Technical Journal, Volume 27, Issue 3, July 1948, pp. 379-423.
16. J, Sänger, C. Richthammer, S. Hassan, and G. Pernul, “Trust and big data: A roadmap for research”, Proceedings of the 25th International Workshop on Database and Expert Systems Applications DEXA, pp. 278-282, IEEE, 2014.
17. Tommy Tandera, et. al., “Personality Prediction System from Facebook Users”, Proceedings of the 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia, pp. 604-611.
18. Ashwin K.T., P. Kammarpally, and K. M. George, “Veracity of information in twitter data: A case study”, Proceedings of the International Conference on Big Data and Smart Computing BigComp, pp. 129-136, IEEE, 2016.
19. Soroush Vosoughi, “Automatic Detection and Verification of Rumors on Twitter”, MIT Thesisi, May 7m 2015.
20. M.S.Yang,” A Survey of fuzzy clustering” Mathl. Comput. Modelling Vol. 18, No. 11, 1993, pp. 1-16,
21. Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter, “Detection and Resolution of Rumours in Social Media:
22. Tatiana lukoianova,Victoria L.rubin (2014) ”Veracity Roadmap:is Big Data objective,Truthful and Credible?”.
23. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 1746–1751.
24. Johnson, R., & Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. To Appear: NAACL-2015, (2011).
25. Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., & Hao, H. (2015). Semantic Clustering and Convolutional Neural Network for Short Text Categorization. Proceedings ACL 2015, 352–357.