Twitter sentiment analysis using machine learning algorithms

[1] Anjelin Genifer Edward Thomas, [2] R.B.Sarooraj

Abstract-- *In the era where social network is considered as a means to express people's opinions and emotions on various topics, it becomes important to understand those views and get in- sights from it. Twitter is one of the most popular networking media for individuals to express their opinions using tweets on any subject of their choice. All of these tweets is "data" for the marketing company, which they can mine and ex- tract useful information to enhance their products. Data is considered to be the most valuable resource right now and we have various technologies to analyze, manage, process and integrate them. The aim of this paper is to mine emotions or sentiments from the available data (tweets) from social media mainly Twitter. Various sentiments can be seen when a user posts or tweets about a recent incident, or a newly released movie or a brand new product. These sentiments help us under- stand the reception of that particular subject. Sentiment Classification of twitter data is basically categorizing the tweets posted by individuals based on polarity or emotions such as Positive, Negative and Neutral. The tweets by every users vary based on the usage of language, emoticons, hash- tags etc which needs to be first preprocessed and converted into a standard format. After preprocessing, useful features needs to be extracted to perform Sentiment Analysis using various Machine Learning techniques.*

Keywords-- *Sentiment Analysis, Machine Learning, Artificial Neural Net- work, Classification Algorithm*

I INTRODUCTION

Twitter is an online social networking tool which receives more than 500 million messages each day on topics all over the world. The amount of information available on twitter, ranging from all industries makes it a hub of data for doing sentiment analysis. The best thing about twitter is that the data (tweets) are easily available to the users. Twitter limits its tweets to 140 characters which makes it easy for a user to express their views in short. Using Twitter over other social networks because of its extent to various topics and a range of people across the world.

In today's fast paced age, everyone wants to get immediate feedback or result about how the product is or is the movie good or what is going on in the other part of the country. Also, people constantly keep an eye on social networking sites to keep themselves updated. They immediately classify any information into positive/negative/neutral. This default behavior of humans is nothing but the sentiments. With the explosion of user generated reviews it has become the need of the hour for analyzing the sentiments of the tweets so that the choices can be made better and profit can be earned. This idea is known as Sentiment Analysis.

[1] Post Graduate Student, SRM Institute of Science and Technology, Email: anjelin_genifer@yahoo.co.in

[2] Assistant Professor, SRM Institute of science and technology Sehore, Bhopal, MP, India, Email:saroorab@srmist.edu.in

Received: 27 Mar 2020 | Revised: 20 Apr 2020 | Accepted: 05 May 2020

The term Sentiment Analysis was first coined in [11]. It has been used extensively in various domains like politics, media, medicine, science, films etc. Ample research has been done on how sentiment analysis can be done. Using Sentiment Analysis, the text can be classified as follows:

- polarity of the sentiments i.e. positive, negative, and neutral
- agree or disagree [11]
- good or bad [12]
- support or opposition [13; 14]
- pros and cons [15]

This paper focuses on polarity of the sentiments to ana- lyze them. It experiments various machine learning mod- els like Naive Bayes (NB), Support Vector Machine (SVM) and Recurrent Neural Network (RNN) for performing Sen- timent Analysis. The performance of the three models is compared by evaluating their accuracy, time and precision. The dataset is extracted using the python client for official TwitterAPI called ”tweepy”. Using the TwitterAPI, senti- ment analysis has been done on tweets related to android and ios mobile phones.

The dataset consists of a mixture of words, emoticons, web links and references to people. Web links and references to people do not contribute to understanding the sentiment of a text and hence those can be ignored. But words and emoticons help in analyzing the sentiment of the text. Thus they act as the features to the machine learning models. The commonly used words in the samsung (mobile phone) dataset is as shown in Figure 1.

II LITERATURE REVIEW

Twitter or any social networking sentiment analysis has been done by people so that they can understand how a product is performing. [1] uses hadoop to automate the prediction of sentiments of the tweets posted by people on social network. It considers real time tweets as it is the latest data for doing analysis. The disadvantage here is they used unigram model instead of n-gram, which means it does not take into account the semantic meaning of the words. The preprocessing also is quite minimal in this approach

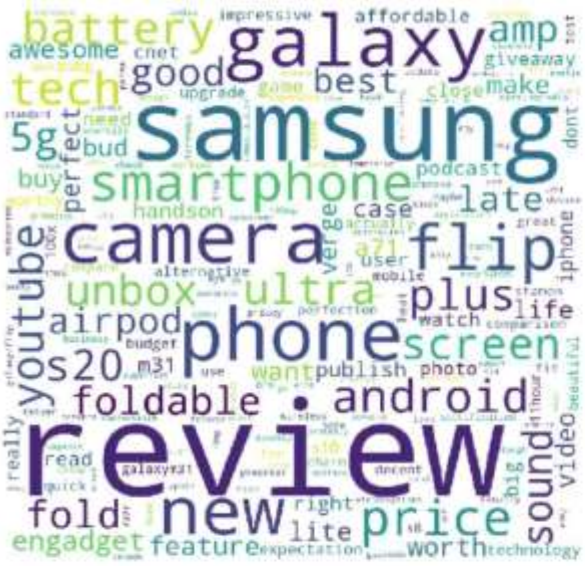

Figure 1: Features extracted from tweets

The main purpose of [2] is to automate the sentiment analy- sis by not involving humans in it and get information related to the trending topics in the order which is most popular. The sentiments are analyzed by interpreting the reactions in the tweets. They experiment UP-growth algorithm which is a popular mining algorithm. This algorithm consumes lot of time, is prone to errors and is difficult to automate. How- ever they have used Emoticon dictionary which contributes to the analysis effectively.

The approach discussed in [3] deals with automated opin- ion based analysis of topics on social network. It proposes the use of Support Vector Machine [17] (SVM) and Ada-boosted Decision Tree algorithms separately and then combining the two to get a hybrid approach. They are using 3 datasets - Stanford Sentiment, Polarity Dataset, University of Michi- gan with pre-processing. The accuracy of this approach is around 84%. Some drawbacks of it are its only used for En- glish Language, semantic meaning is neglected and sarcasm is ignored.

In [4], they compared the accuracy of different algorithms. Out of all the algorithms that were experimented, Recurrent Neural Network using LSTM had the highest accuracy with 93%, while the Naive Bayes had the lowest accuracy of 60%. The second highest accuracy of 91% was depicted by Sup- port Vector Machine. Decision Tree had accuracy of 86% and K-Nearest Neighbor had 85%. The drawback of this system is it uses BOW technique using CountVectorizer.

In [5], again the approach is using BOW with CountVector- izer showing the perfomance of SVM as better than Naive Bayes. In [6], discusses a lexical transformation approach used to correct the various shortcuts used in tweeting and texting and they have concluded that most of the words converted were irrelevant and is insignificant for analysis.

In all of the above, we notice the usage of unigram model and BOW. Also, the overall sentiment is classified only based on the tweet data. The likes and retweets are not considered.

III EXPERIMENTAL SETUP

Our proposed solution consists of 4 phases as illustrated by Figure 2. The first phase involves data collection, followed by pre-processing the data and extracting the features. Once the required features are available, they are given to the ma- chine learning classifier which classifies the sentiments based on polarity (positive/negative/neutral). Finally, the results are displayed using report visualizer. Our approach differs from the approaches suggested in the literature reviews:

- Additional Preprocessing steps
- TFIDFVectorizer with ngram instead of CountVector- izer
- Variations in the algorithm parameters
  - Multinomial Naive Bayes with alpha parameter as 1
  - SVM with rbf as kernel and gamma parameter as 10
  - RNN with activation function as softmax and Op- timiser as RMSProp

The data set we have used is obtained from the Twitter API and in particular, the following data is of importance to us:

- Tweets: On which sentiment analysis is performed.

- Favorite count: identify how many like this.
- Retweet count: identify how many have retweeted the same

Twitter dataset is collected using the Tweepy module which is the python client for the official Twitter API[16]. The tweets can be fetched form Twitter API, by registering the App through the twitter account. Once the app is created, there will be ‘Consumer Key’, ‘Consumer Secret’, ‘Access token’ and ‘Access Token Secret’ which will be used to get the data set. We query the data by giving keyword, time since we need the data and filter out the retweets. Once the data is fetched its loaded into a flat file for further processing. Twitter API has the option of extended entities which gives us the link to any other media posted such as images. The various attributes that will be extracted from Twitter API are as follows:

- Tweet text
- Favorite count
- Retweet count
- Source
- Place
- Created At
- Media url

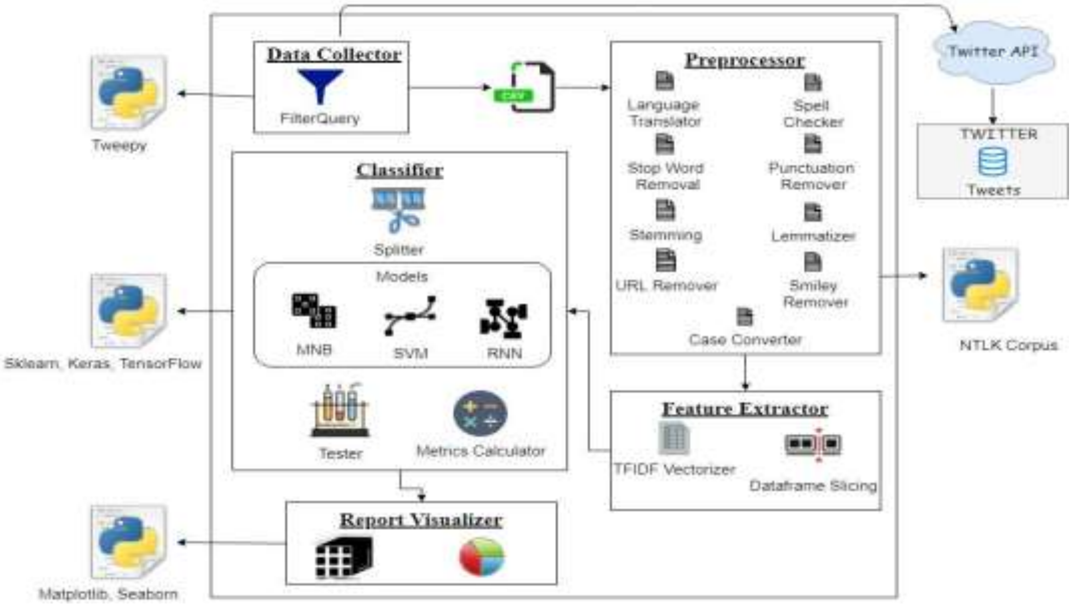

Figure 2: Sentiment Analysis Framework

Before doing any analysis on the data, it needs to be pre- processed as it makes the data ready for analysis. The aim of this step is to clean irrelevant text such as punctuation, special symbols, numbers, and terms which are not very important for sentiment analysis. Usually the data collected from twitter is noisy data. Thus various preprocessing operations as depicted in Figure 3 is performed. Also, NLTK which is an impressive library to process natural language is used to extract better features.

The text available as tweets will always consist of upper- case and lower-case. As the analysis is case-sensitive which means ”Best” and ”best” are considered as different words, it is a good practice to convert the text to lower-case.

Twitter being popular all across the world, the tweets can be posted in various languages. In order to generalize the processing methodology, the tweets are first converted into English language.

Making error is human and those errors can be found in the tweets in the form of spelling errors. Hence, they need to be corrected before doing analysis.

As the classifier is based on words, eliminating the words which will not help in analysis and which does not give any semantic information will classify the sentiments with better accuracy. Some of the examples of stop words are - ”I”, ”me”, ”myself”, ”is”, etc.

Remove all the punctuation marks which will not be used for analysis. One of the approach to remove the smiley would be to re- place it with its equivalent meaning so that it also con- tributes in sentimental analysis. Stemming will change the word into its stem form mainly by removing suffixes and prefixes like ’s’, or ’ed’ which might not return an actual word. This also returns the base form of the word but its main difference from stemming in the way it generates an actual word.

TFIDF stands for Term Frequency-Inverse document frequency which as the name suggests associates a score to each item based on its frequency as well as considers how rare its occurence is. Term frequency times Inverse document frequency help in identifying those rare features which will contribute to the sentiment analysis. TFIDF supports n gram.

$$TF(w) * IDF(w) = \left(\frac{n_w}{n_d}\right) * log_e(\frac{N}{df_w})$$

where nw - Number of times term w appears in a document nd - Total number of terms in the document

N - Total number of documents dfw - Number of documents with term w in it.

Naive Bayes is a simple model which can be used for text classification. We use the Multinomial Naive Bayes algo- rithm with TFIDFVectorizer. The overall text is tokenized and the word frequency is considered. The reason we are go- ing ahead with Multinomial Naive Bayes algorithm instead of Naive Bayes, as it is considered that Multinomial Naive Bayes algorithm gives better accuracy than Naive Bayes for text classification. In Naive Bayes, its only the occurrence of the words that is considered in calculation whereas in Multinomial Naive Bayes it considers the frequency of the occurrence of the words as well. Laplace smoothing is used to handle the zero probability. (alpha=1).

Naive Bayes Formula:

$$P(\frac{sentiment}{sentence}) = \frac{P(\frac{sentence}{sentiment}) * P(Sentiment)}{P(Sentence)}$$

$$P(sentence) = P(word_1) * P(word_2) * ..P(word_n)$$

$$P(\frac{sentence}{sentiment}) = P(\frac{word_1}{sentiment}) * P(\frac{word_2}{sentiment}) * ..P(\frac{word_n}{sentiment})$$

Support Vector Machine (SVM) is a binary linear classifier which is non-probabilistic. For training data (xi , yi) where x is the feature vector and y is the class, we want to find the maximum-margin hyperplane that divides the points with yi = 1 and yi = 1.

The equation of the hyperplane is w * x – b = 0 We want to maximize the margin, denoted by γ, as follows:

$$\max w, \gamma s.t. \gamma <= y_i \, (w * x_i - b)$$

we use TFIDF Vectorizer with the kernel function as rbf. The kernel is selected as ’rbf’ as it works better when we don’t have much past knowledge of data. It uses Euclidian distance. We use the gamma parameter as 10.

The third algorithm we use is Recurrent Neural Network (RNN) with 3 Long short-term memory (LSTM) layers and Activation function as softmax and Optimizer as RMSProp. We consider 5 epoch with batch size as 100. LSTM performs better for sequence of data. Hence accuracy is improved using this approach.

We use matplotlib and seaborn libraries to show the various findings and results. Overall Sentiment is calculated by the (sentimentvalue + maximumlike + retweet). This will show the classification report and confusion matrix and overall sentiment pictorially for easier understanding for the user.

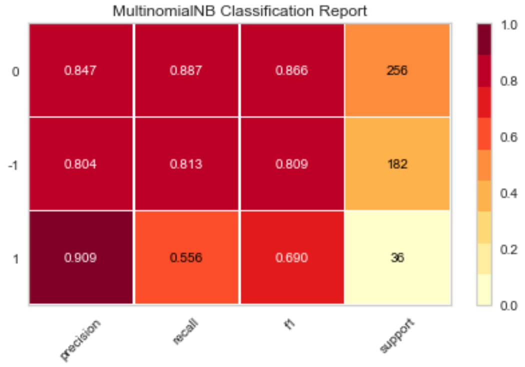

Figure 3: Multinomial Naive Bayes Classification Report

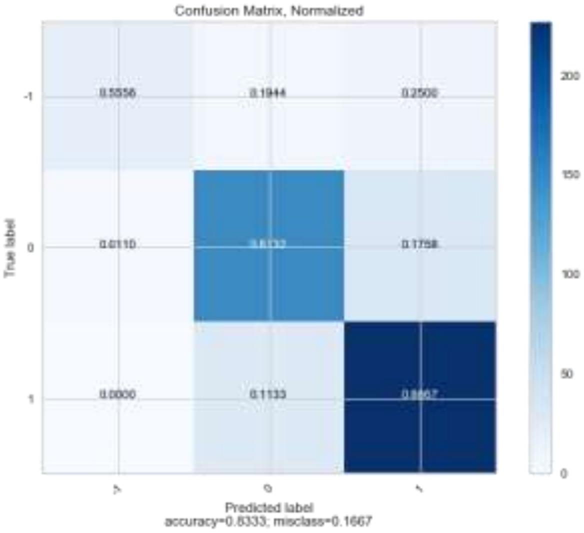

Figure 4: Multinomial Naive Bayes Confusion Matrix

Received: 27 Mar 2020 | Revised: 20 Apr 2020 | Accepted: 05 May 2020

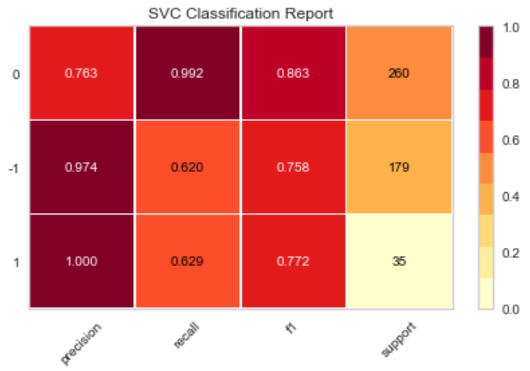

Figure 5: Support Vector Machine Classification Report

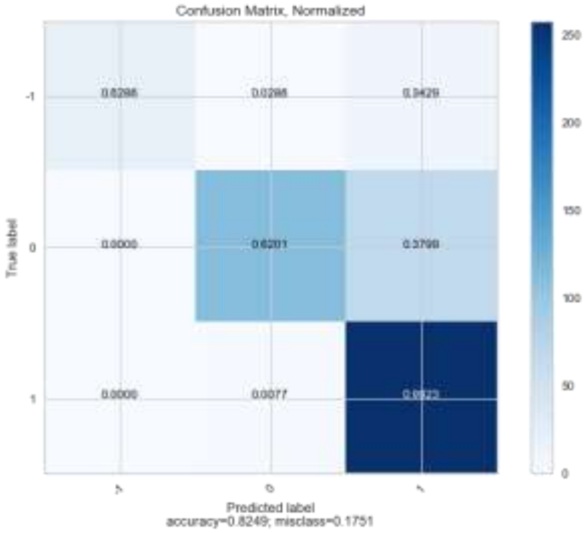

Figure 6: Support Vector Machine Confusion Matrix

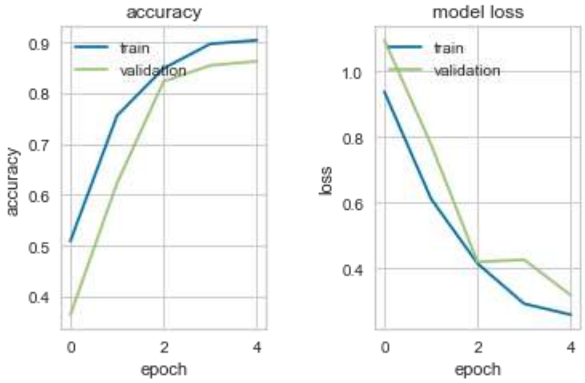

Figure 7: Recurrent Neural Network Accuracy / Loss

To observe the performance of the machine learning models, accuracy measure is used as shown in table 1.

Table 1: Accuracy of models

Machine Learning Models	Accuracy
Multinomial Naive Bayes	83.33%
Support Vector Machine	82.49%
Recurrent Neural Network	93.4%

Received: 27 Mar 2020 | Revised: 20 Apr 2020 | Accepted: 05 May 2020

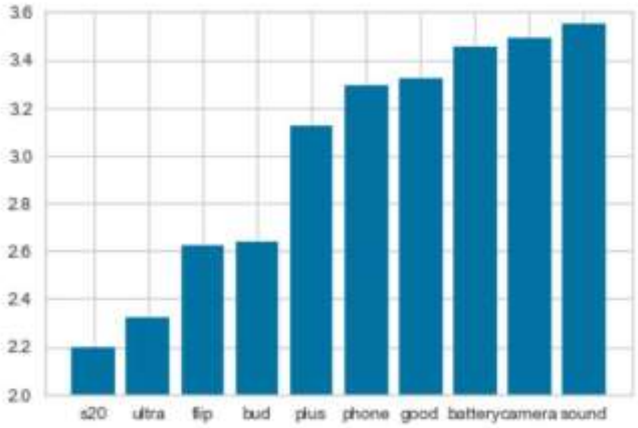

Figure 8: Framework Top 10 words in Tweets

Out of the total tweets in the dataset, the distribution of positive, negative and neutral reviews is as shown in Figure 12 and in the table 2.

Table 1: Sentiment Type Distribution

Sentiment	Percentag
Positive	55.67%
Negative	14.85%
Neutral	29.46%

Today, people are saying that Galaxy Z Flip is Samsung's worst phone.
Samsung Galaxy Z Flip Review: 48 hours later
Samsung Galaxy Z Flip review: The best foldable around

Figure 9: Sentiment Analysis Framework

Some examples of positive, negative and neutral tweets from the dataset are depicted in Figure 13.

IV CONCLUSION

This paper concludes that we can improve the accuracy of the various classifiers by the above mentioned pre-processing techniques - to convert the sentences into relevant words and by adjusting the various parameters mentioned for each of the model ensuring they don't over-fit the model. On com- paring these models as expected, Recurrent Neural Network performs the best on Twitter data set for the sentiment anal- ysis of tweets. Also we notice that the likes and retweets affect the overall sentiment classification and cannot be ignored. The approach followed in this paper for twitter sentiment analysis can be scaled to include other businesses as well.

Considering tweets from a recent time period can help track performance of the product and compare it with its past performance and its competitors, this eliminates the problem of using tweets that might not be relevant any longer. Also, there is a lot of work that needs to be done in getting the content in tweets which are in other media formats like audio and video. The other enhancement would be identify the subjectivity and truth in the tweets

REFERENCES

1. Sneh Paliwal, Sunil Kumar Khatri, Mayank Sharma, "Twitter Sentiment Analysis using Deep Neural Network," *International Conference on Inventive Research in Computing Applications*, 2018.
2. Subramaniam.G, Ranjitha.M, "User Emotion Analysis using Twitter Data", *IFET COLLEGE OF ENGINEERING.*
3. M.Trupthi, Suresh Pabboju, "SENTIMENT ANALYSIS ON TWITTER USING STREAMING API" *IEEE 7th International Advance Computing Conference,* 2017.
4. Yaser Maher Wazery; Hager Saleh Mohammed ; Essam Halim Houssein, "Twitter Sentiment Analysis using Deep Neural Network," *2018 14th International Computer Engineering Conference,* 2018.
5. Pitiphat Santidhanyaroj, Talha Ahmad Khan, "A SENTIMENT ANALYSIS PROTOTYPE SYSTEM FOR SOCIAL NETWORK DATA," *Faculty of Engineering and Applied Science University of Regina Regina.*
6. Stephan Gouws, Donald Metzler, Congxing Cai and Eduard Hovy., "Contextual Bearing on Linguistic Variation in Social Media" , *Workshop on language in social media*, 2011
7. Megha Rathi, Aditya Malik, Daksh Varshney, Rachita Sharma, Sarthak Mendiratta, "Sentiment Analysis of Tweets using Machine Learning Approach", *Jaypee Institute of Information Technology*.
8. Thorsten Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features".
9. Amit G. Shirbhate1, Sachin N. Deshmukh, "Feature Extraction for Sentiment Classification on Twitter Data" *International Journal of Science and Research (IJSR).*
10. Nasukawa ,T. Yi, J, "Sentiment analysis: capturing favorability using natural language processing," *In Proceedings of the 2nd international conference on Knowledge capture,* pp. 70–77, 2003.
11. Balahur, A., Kozareva, Z., Montoyo, A. "Determining the polarity and source of opinions expressed in political debates", 2009.
12. Ku, L.W., Li, L.Y., Wu, T.H., Chen, H.H., "Major topic detection and its application to opinion summarization". *In Proceedings of the ACM Special Interest Group on Information Retrieval*, 2005.
13. Bansal, M., Cardie, C., Lee, L., "The power of negative thinking: Exploiting label disagreement in the min-cut classification framework," *In Proceedings of the International Conference on Computational Linguistics,* pp.15-18, 2008.
14. Terveen, L., Hill, W., Amento, B., McDonald, D., Creter, J. "A system for sharing recommendations," *Communications of the Association for Computing Machinery*, vol. 40, no. 3, PP. 59-62.
15. Kim, S.M. Hovy, E. "Automatic identification of pro and con reasons in online reviews," *In Proceedings of the COLING/ACL Main Conference Poster Sessions,* pp. 483-490, 2006.
16. *A. Green, Twitter API Engagement Programming, Adam Green Press, 2013*

17. Srinivasan, C., Suneel Dubey, and T. R. Ganeshbabu, “Complex Texture Features For Glaucoma Diagnosis Using Support Vector Machine,” *International Journal of MC Square Scientific Research*, vol. 7, no. 1, pp. 81-92, 2015.