# Predicting Sentiments to an accuracy matching the gesture recognized for the specially-abled

Jaganath Prasad Mohanty , Ayas Kanta Swain, Kamalakanta Mahapatra
Department of Electronics and Communicaton Engineering
National Institute of Technology, Rourkela, India
vlsi.nitrkl@gmail.com

***Abstract*—Sentiment analysis being a polarity based detection technique, is going around in the academic research and industry domain for quite a long time, not only seeing the economic benefits to both fields alike, but also the significance in understanding human behavior. In this work, Natural Language Processing (NLP) is done to accurately recognize the sentiment behind the gestures of a specially abled individual, to analyse their behavior in real time considering different application platforms.**

***Keywords*-Sentiment Analysis; Natural Language Processing; Gesture Recognition; Sign Language; Artificial Intelligence.**

## I. Introduction

Specially abled individuals, specifically the deaf and dumb public, have a way of communication, which is usually hand gestures that necessitates interpretation for people who do not understand the format known as Sign language. Consumer Industry [8] which provides services on-demand basis can target these public by considering their requirements, online and/or offline though Sign Language Interpreters (SLI). Special purpose cameras like Kinect [4] were used for assisting in interpreting actual gestures, but with the advent of neural networks [1] and machine learning [2] capabilities, even a web camera can be of optimal usage for predicting keywords portraying dialect relevant to the actual word.

Once words are selected, next step is to join them in a meaningful context that accurately matches the sentiment of the speaker. This is where Natural Language Processing (NLP) [6] comes into picture. NLP is of paramount significance aiding in accurately representing the words in a statement empathizing the speaker. Opinion can be classified as positive, negative or neutral [7].

The interpretation and classification of emotions in textual format using analysis techniques to identify customer satisfaction through services feedback which aids towards brand reputation is sentiment analysis. User generated opinionated data through social media platforms are the main sources of input information for analyzing human behavior in context to a particular topic. Public opinion is used by corporations to improve their services in real time. Machine learning algorithms like, artificial neural networks (ANN), k-nearest neighbors (kNN), naive Bayes (NB), and random forest (RF) as well as lexicon-based methods are few used to relate texts to sentiments [3].

Instead of the input tweets as texts for analyzing the sentiments, when gesture recognized keywords are studied to format and construct empathizing statements, opinions of a deaf and dumb individual can be matched to a high degree of accuracy, as shown in figure 1. With this working direction, the presented work is focusing towards achieving a target of improved accuracy in structuring sentences from keywords generated by the gesture recognized images.

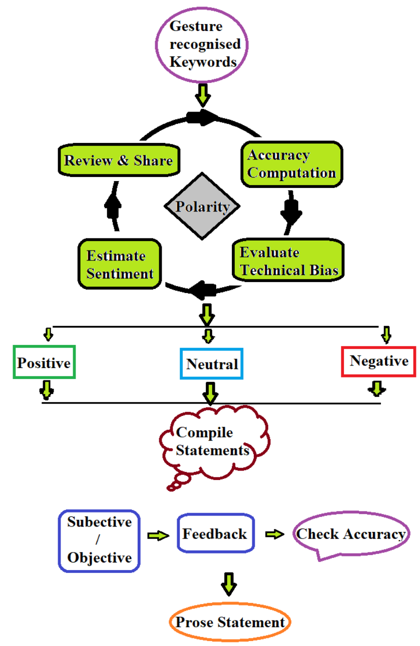



Fig. 1: Gesture Recognized Sentiment analysis.

Dynamic images are converted to texts in real time with flexible accuracy, and a database is created to make statement of the corpus analyzing behavior of words by classifying into positive, negative or neutral features. The problem space lies in the course of accuracy, considering the vernacular linguistics involved. With the aid of Naïve Bayes binary classification, conditional probabilities and log likelihood, an amount of comparatively fine accuracy in predicting the actual sentiment behind the recognized keywords have been attained in the presented work, though a lot is left to be accomplished. Quite a lot of work has been done in the pretence of sentiment analysis [10], but rarely any for the specially abled assistance, hence

978-0-7381-4264-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IEEE-iSES50453.2020.00027

the necessity of research in this direction.

The paper is structured as follows. Section II represents the background work going around in this domain. The implementation aspects of the sentiment analysis, initiating from pre-processing, to stemming, tokenizing, and extracting features are discussed in section III along with finalizing logical regression for predicting sentiment to a certain level of accuracy. Section IV shows the intermittent results of predicting sentiments with graphs. Section V concludes the work in progress and trails interim future directions targetted in this direction.

## II. Background

Customer's opinions matters significantly to service industry for maintaining their merit on the long run belonging to business factors. Sentiment analysis using Artificial Intelligence (AI) based infrastructure has actuated a higher economic value for the industry to an extent never achieved before. Consumer electronics industry has gained a huge advantage due to the inclusion of this AI based technology. It has not only understood the habits of an individual through their shared data, but has also begun to convince them of thier needs and requirements as per the goods available at the moment [12].

Gesture recognition is a field of research much anticipated for Brain Machine Interface in technology domain. Smart Internet of Things(IoT) electronic devices make use of this feature for hassle free communication between human and machines. The same technology can be used for improving the livelihood of billions of specially abled individuals. A few start ups have considered the case for their business. This work is based on recongising dynamic images taken from a camera in real time, and converting to Convolutional Neural Network (CNN) based data set for generating respective keywords relevant to certain gestures.

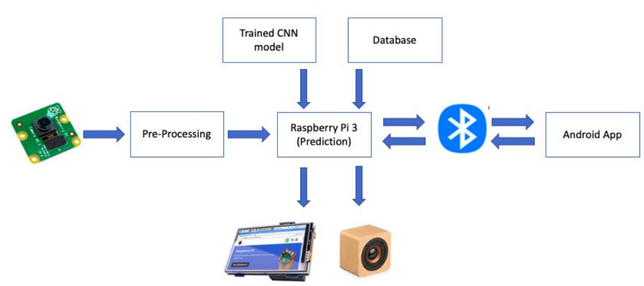



Fig. 2: Block diagram for sign language interpretation.

The general idea behind the real time analysis of the work can be understood from figure 2. The shown SLI system is based on deep CNN training architecture, performed on Google cloud NVIDIA Tesla T4 with 2560 cores. The image dataset was necessary to make CNN user-independant. A raspberry pi 3 has been actuated as the main controller whose camera is used for capturing input hand gestures. The voice output uses a speaker and the developed graphical user interface takes the aid of a LCD screen integrated with a Bluetooth media for communication with specially abled individuals.

Initial steps incliude the prediction of signs related to numbers and alphabets. This is pre-processed using filters to remove noise. The keywords are predicted in real time using the database trained and stored in the pi memory following an algorithm shown in figure 3. Maximum occurence approach of an alphabet is taken into consideration for proper identification and framing of the keywords. Initially these keywords are displayed in the LCD and fed as acoustic outputs to the speaker.

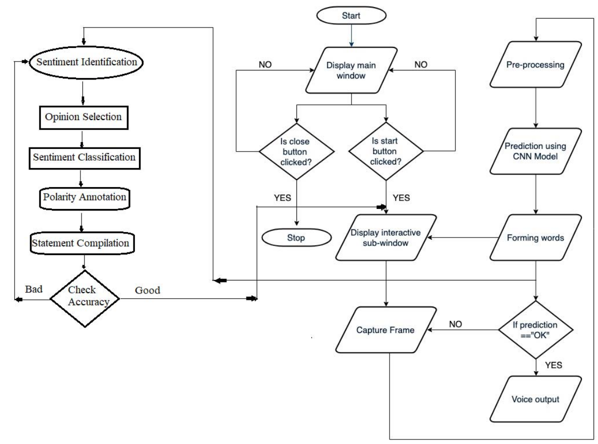



Fig. 3: Proposed sign Language recognition with sentiment polarisation.

Using the keywords, for interim results, a database was created estimating predefined statements fed in the pi memory. This was used as an intermediary for the acoustic signals as communication. But this became a glitch in many a platforms, as estimatng the actual opinions and sentiments of specially abled customers in real time was difficult. Citing this problem statement, a proposed framework was configured. Using the analysis of customer opinions shared in popular social media platforms, business holders improve services by estimating the sentiments through AI. The same idea can be reengineered to predict sentiments and further prose statements matching accurately the sentiments of gesture based keywords. An overview of the work behind this propsed system can be viewed in the figure 3.

A similar concept has been actuated by Wang and Masaki [11], where visual sentiments are predicted using hand-craft and CNN features. But to the best of our knowledge, no methodology has been proposed for SLI based gesture recongnised images and respective built-up keywords. Furthermore, prosing accurate statements relative to the gestures is a mined work scarcely available in literature. Hence, the next step is to analyse these keywords for their polarity and prose statements with improved accuracy. A detailed description of the ongoing proposal is done in the next section. The depiction is a conceptualized work in the direction of assisting the specially abled individual, assuming their understading of sign language to be naive.

## III. Contextual Procedure

This section describes in detail the technical experiments carried out to classify a set of keywords as input, and relatively predict their sentiments to certain accuracy, based on the contexts available. With the aid of Jupyter platform, a python code was inscribed using NLTK, traditional NLP library to categorize an input data as positive or negative sentiment which can be to a calculated accuracy [9]. In the present context, imported inputs are processed statements, keywords chosen randomly, from gesture based recognition procedure, which will be predicted for their sentiments. Certain stop words are used to differentiate among the linguistics and segregate them.

Initially a database of statements are formed with specific keywords which will actuate the sentiments. An array of frequency of occurrence of the words are build up. The building blocks is made using the $build\_freqs()$ function in python. This is the function that creates the dictionary containing the word counts from each corpus. Each key is a 2-element tuple containing a '(word, y)' pair. The 'word' is an element in a processed keyword while 'y' is an integer representing the corpus: '1' for the positive and '0' for the negative keywords. The value associated with this key is the number of times that word appears in the specified corpus. Post making the building blocks nearly, 4000 samples of data is segmented as positive and negatives each, for training approximations. Preprocessing of inputs using tokens and stopwords are vital part of the process.

### A. Stop words

This is a list of stop words generally used in vocabulary ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', … ]

*1) Punctuation:* This is a list of Punctuation used

! " # $ & ' ( ) / % * + - , : ; ¡ = ¿ @ [ ] ^_ { — }

*2) Adding stop words and punctuations:* The next step is to attach stop words and punctuation. Stop words are words that don't add significant meaning to the text. The list provided by NLTK when the cells are run is shown. The stop words list contains some words that could be important in some contexts. For the punctuation, it was seen earlier that certain groupings like ':)' and '...' should be retained when dealing with keywords because they are used to express emotions. In other contexts, like medical analysis, these should also be removed.

*3) Stemming:* Stemming is the process of converting a word to its most general form, or stem. This helps in reducing the size of the vocabulary. NLTK has different modules for stemming; the PorterStemmer module which uses the Porter Stemming Algorithm has been used. Example of a use case.

['beautiful', 'sunflowers', 'sunny', 'friday', 'morning', ':)', 'sunflowers', 'favourites', 'happy', 'friday', '…']

stemmed words: ['beauti', 'sunflow', 'sunni', 'friday', 'morn', ':)', 'sunflow', 'favourit', 'happi', 'friday', '…']

### B. Logistic regression cost function

The intuition behind the cost function is to understand its way of design. A detailed explanation will show what happens when the true label is predicted and when predicted one is wrongly labeled. The equation (1) below shows the cost function, which might look like a big complicated equation, but is actually rather straightforward, once it is broken down into its components.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^i \log h(x^i,\theta) + (1-y^i)\log(1-h(x^i,\theta))] \tag{1}$$

A look at the first part of the equation on right hand side, where a sum over the variable m is found, is just the number of training examples in the training set. This indicates the sum over the cost of each training example. The minus sign ensures that the overall costs will always be a positive number. Inside the square brackets, the equation has two terms that are added together. The first term is the product of y superscript i, which is the label for each training example, most applied by the log of the prediction, which is the logistic regression function applied to each training example represented as h of x superscript i and $\theta$.

From this equation it can be seen that there is one term in the cost function that is relevant when label is 0, and another that is relevant when the label is 1. In each of these terms, log of a value between is 0 and 1, which will always return a negative number, and so the minus sign out front ensures that the overall cost will always be a positive number.

First, the loss when the label is 1 is taken into consideration. In this case J($\theta$) simplifies to just negative log h(x($\theta$)). When prediction is close to 1, the loss is close to 0 because prediction agrees well with the label. And when the prediction is close to 0, the loss approaches infinity, because prediction and the label disagree strongly.

The opposite is true when the label is 0. In this J($\theta$) reduces to just minus log(1- h(x,$\theta$). Now when the prediction is close to 0, the loss is also close to 0. And when prediction is close to 1, the loss approaches infinity. This is how the logistic regression cost function works. A pseudo code is displayed below for getting the gradient descent as a reference

Using Naive Bayes, which is a different type of classification algorithm, allows predicting whether a keyword is positive or negative. Using logistic regression, the training set is subdivided to arrays of fixed length and the corresponding sigmoid is calculated. This is used in iterative calculation of the gradient descent through cost function estimation. This gets the cost function after training and the resulting vector of weights, which can be stored in CSV format for an initial stage of sentiment analysis described in the next section.

**Algorithm 1:** Attaining the gradient descent

**Result:** def gradientDescent(x, y, theta, alpha, $num\_iters$)

get 'm', the number of rows in matrix x;
m = len(y);
**while** $i < num\_iters$ **do**
    z = np.dot(x,theta);
    h = np.exp(z)/ (1+ np.exp(z));
    J = 1/m * np.dot(np.transpose(x),(z - y));
**end**
theta = theta - (alpha * J);
return J, theta;

## IV. Results and Discussion

In this section a table depicting the extracted features are displayed which is then utilized to classify and predict general words as positive or negative and are plotted for semantics. Improving the accuracy of sentiments in a statement needs smoothening of deviations, and reducing the frequency of error, as described in the later part of this section.

Extracting features is an essential part of the process, even though biasing can be a put off at this stage. Table I shows a partial feature extracted table from an inferred data set whose description is provided in this part.

TABLE I: Partial Feature extraction

| | *bias* | *positive* | *negative* | *sentiment* |
|---|---|---|---|---|
| 0 | 1.0 | 3020.0 | 61.0 | 1.0 |
| 1 | 1.0 | 3573.0 | 444.0 | 1.0 |
| 2 | 1.0 | 3005.0 | 115.0 | 1.0 |
| 3 | 1.0 | 2862.0 | 4.0 | 1.0 |
| 4 | 1.0 | 3119.0 | 225.0 | 1.0 |
| 5 | 1.0 | 2955.0 | 119.0 | 1.0 |
| 6 | 1.0 | 3934.0 | 538.0 | 1.0 |
| 7 | 1.0 | 3162.0 | 276.0 | 1.0 |
| 8 | 1.0 | 628.0 | 189.0 | 1.0 |
| 9 | 1.0 | 264.0 | 112.0 | 1.0 |

The creation of the numerical features needed for the Logistic regression model is a key step. In order not to interfere with it, a previously calculated table is adopted and these features are stored in a CSV file for the entire training set. A few of the data entries for a sample input are shown in the table. The vector $\theta$ represents a plane that split feature space into two parts. Samples located over that plane are considered positive, and samples located under that plane are considered negative. Practically a 3D feature space, i.e., each keyword is represented as a vector comprised of three values: [bias, $positive\_sum$, $negative\_sum$], always having bias = 1, is included.

If ignoring the bias term, plotting each keyword in a cartesian plane is possible, using $positive\_sum$ and $negative\_sum$. In the figure 4, precisely this is shown. Additionally, each keyword is colored, depending on its class. Positive keywords are depicted as green and negative keywords as red. From the plot, it is evident that the features that are chosen to represent keywords as numerical vectors allow an almost perfect separation between positive and negative keywords. Thus a very high accuracy for this model can be expected. A gray line is drawn to show the cutoff between the positive and negative regions.

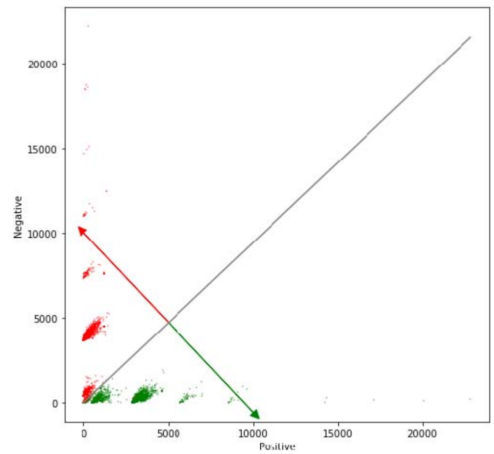



Fig. 4: Feature Extraction in Cartesian plane.

In other words, the gray line marks the line where
z = $\theta * $ x = 0

The red and green lines that point in the direction of the corresponding sentiment are calculated using a perpendicular line to the separation line. It must point in the same direction as the derivative of the Logit function, but the magnitude may differ. It is only for a visual representation of the model.

$$direction = pos * \frac{\theta 2}{\theta_1} \quad (2)$$

In figure 4, the green line in the chart points in the direction where z $>$ 0, and the red line points in the direction where z $<$ 0. The direction of these lines is given by the weights $\theta 1$ and $\theta 2$. Note that more critical than the Logistic regression itself, are the features extracted from keywords that allow getting the right results in this. Further, using the above functions, predicting the sentiment of an input statement let alone a word in the context of a gesture recognized through image, video or real-time is not complex.With the positive and negative classification of the corpus, the steps involved are (1) counting the words in each classifiers, (2) make a total count of all words in each, which is necessary for the conditional probabilities. Dividing the frequency of each word in a class by its corresponding sum of words, and storing them in a table, gives the probabilities of each class noting the number of identical probabilities in it.

$$P(w_i|class) = \frac{freq(w_i, class)}{N_{class}} \quad (3)$$

$class \ \epsilon \ (Positive, Negative)$
N$class$ = frequency of all words in class

After creation of the table of probabilities, Laplacian smoothing technique is used to amplify the weightage of words that contribute less vital to the corpus.

$$P(w_i|class) = \frac{freq(w_i, class) + 1}{N_{class} + V} \tag{4}$$

V = number of unique words in vocabulary The ratio of probabilities with the set of m words depicts positive keywords if the product of the ratios is greater than 1, otherwise neutral or negative.

$$\prod_{i=1}^{m} \frac{P(w_i, pos)}{P(w_i, neg)} > 1 \tag{5}$$

And the log likelihood is used to make the analysis more accurate with less frequency of error. The overall sum of the logarithm of ratio of probabilities with the set of m words depicts positive keywords if the product of the ratios is greater than 0, otherwise neutral or negative.

$$\sum_{i=1}^{m} \log \frac{P(w_i, pos)}{P(w_i, neg)} > 0 \tag{6}$$

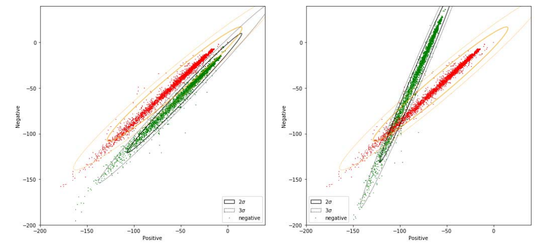
The table II shows a partial dataframe, which is the data structure that simplifies the manipulation of data, allowing filtering, slicing, joining, and summarization.

TABLE II: Dataframe of the few keyword features. Each row represents a keyword

| | *positive* | *negative* | *sentiment* |
|---|---|---|---|
| 0 | -45.763393 | -63.351354 | 1.0 |
| 1 | -105.491568 | -114.204862 | 1.0 |
| 2 | -57.028078 | -67.216467 | 1.0 |
| 3 | -10.055885 | -18.589057 | 1.0 |
| 4 | -125.749270 | -138.334845 | 1.0 |

The visualizing likelihoods of keywords and the confidence ellipses for annotated corpuses are briefly described. The positive and negative likelihoods of a keyword can be interpreted as a pair of numerical features extracted from raw text. The visual inspection of the features that will feed a model allows to better understand the problem. Using the confidence ellipses, the Naïve Bayes model can be interpreted which is better than plotting the points over a cartesian plane, usually with big data sets. These ellipses summarize the information of the dataset with numerical mean of the attributes, variance (height and width, generally specified by a user considering the desired standard deviation amount) of each attribute and the related covariance among attributes.

The figure 5 depicts a population mean of the bivariate distribution of the data set, which gives a high confidence level signifying accuracy of a sentiment analysed for a specific set of keywords. The $n\_std$ is the number of standard deviations used with the data. Interpreting the depiction, when a high number of means are used with better estimation, the confidence ellipse decreases in size which in turn contains less proportion of actual data.



(a) Bivariate disstributions using 2_std and 3_std (b) sentiments with overlapping distributions

Fig. 5: Population mean data set and its corresponding confidence ellipses.

In this case, the data which is in CSV format are normally distributed, which is the reason behind its clarity in classification, but when the data is not asymptotically distributed, the sample size may have an overlapping normal distribution, as is seen in figure 5. In this case the confidence ellipse will converge neutralizing the sentiments at the junction. This is done by manipulating the data set such that the centre of pressure displacement can be observed over the time of a trial. Since the data for the sentiment classification contains sentiment labels, accuracy of the classifiers can be evaluated, with errors that are not significant relatively. As this is a binary classifier, it can be applied to any dataset. Extensive details on the design are available on the GitHub page of the project [5].

A brief comparative analysis with existing state of the art methodology for analysing sentiments, in different platforms for various applications is shown in table III. A lot of experimentation has been explored and large number of approaches have already been proposed for sentiment analysis (BERT, LSTM, Word2Vec), yet it is not saturated as the results even discussed in this manuscript build on different approaches. Although, the proposal that has been applied here is a method to interpreting gestures for sign language, this algorithm does not refrain from being used in different platforms (such as the feature extraction method).The only focus of this paper is on analyzing sentiment from keywords, a topic which has already been well explored, as can be seen in the table.

TABLE III: Comparision with existing state of the art technology

| **Reference** | *Algorithms* | *Analyse* | *Applications* |
|---|---|---|---|
| Our Proposal | Naive Bayes | Positive and Negative | Feature Extraction for gestures |
| [13] | BERT & RRC | Question Answers | Aspect Extraction |
| [14] | LSTM & CNN | Temporal Infomation | Text Processing |
| [15] | Word2vec & BOW | Twitter comments | Consumer Serevices |
| [16] | Deep Learning | Bilingual texts | Website Reviews |

An approach which suits one and all is not the target of any technology enthusiast, as the demand and supply chain runs on the basic foundation of the eager and anxious mind. There is no restrictions on the usage of this methodology proposed, and there is no assurance of this being an all inclusive design. There can be numerous loopholes as it has been designed by ordinary humans, who are prone to error. Yet with repetitive feedback based approach, the technical biasing has been tried to be kept on minimum side.

Part of the work is prosing statements which uses speech synthesis. Specifically Text To Speech (TTS) system is utilised which is out of scope of this paper and out of context to this platform. Though a lot of work is impending related to statement compilation, to check its subjectivity and accuracy to the pretext, numerous efforts are put forward in respective speech processing community. Needless to say, their efforts are umsurmountable to the smart electronics and consumerables society.

In order to make the services of consumer electronics all inclusive, it is required to work out issues faced by public who are less resilient to technology and more open to adapt to change. Any industry ready to take on this opportunity can gain a cutting edge advantage over others in technology competition and make a reputation of being all encompassing and broad in rationality.

## V. Conclusion & Future prospects

In this article, based on prior work in gesture recognitions, sentiment analysis is used to structure polarity based raw data in textual format by automatically tagging the keywords.Using Naive Bayes theorem and logistic regression function, a cost estimation for improvement in accuracy is proposed. A behavior pattern is analysed and tagged as positive or negative based on the training set of recognized keywords, made available for statement formation.

A new paradigm shift in acknowledging the services offered by an all-inclusive world and consumer industry to specially abled public is foreseen in this work. The same approach as described in this article can be followed for providing essential services to specially abled public and making their lives more comfortable. It can be linked to industry inputs to provide services adherent to their needs.

Future work in this direction will include smoothening of classification for improved analysis and creating statements that match with the behavior of the sign language user. As sentiment analysis is a field of research in the text mining field, it can be used for delivering best services to specially abled public and predict their choices by a service provider for better performing business economics.

## Acknowledgment

This work is a continuation of the project on Gesture recognition using general purpose processors conducted at NIT Rourkela, India under the Special Manpower Development Project – Chip to System Design project initiative through Ministry of Electronics and Information Technology (MeitY) and Digital India Program, funded by the Government of India. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Indian Government.

## References

[1] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney,& Richard Bowden; Neural Sign Language Translation", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7784-7793.

[2] Dabre, K. & Dholay, S. (2014). Machine learning model for sign language interpretation using webcam images. International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA) (pp. 317-321). Mumbai: IEEE Xplorer. doi:10.1109/CSCITA.2014.6839279

[3] Hartmann, J., Huppertz, J., Schamp, C., & Heitmann, M. (March 2019). Comparing automated text classification methods. International Journal of Research in Marketing, 20-38.

[4] João, G., Cardoso, T., & Rybarczyk, Y. (2014). Kinect-Sign, Teaching Sign Language to "Listeners" through a Game. Procedia Technology, pp. 384-391. doi:10.1016/j.protcy.2014.10.199

[5] NLP. (2019). Retrieved from GitHub: https://github.com/jpm18/NLP

[6] Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011/7). Semi-supervised recursive autoencoders for predicting sentiment distributions. Proceedings of the 2011 conference on empirical methods in natural language processing, (pp. 151-161).

[7] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013/10). Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the 2013 conference on empirical methods in natural language processing, (pp. 1631-1642).

[8] Yang, J., Wang, Y., Lv, Z., Jiang, N., & Steed, A. (2018). Interaction with Three-Dimensional Gesture and Character Input in Virtual Reality: Recognizing Gestures in Different Directions and Improving User Input. IEEE Consumer Electro. IEEE Consumer Electronics Magazine, pp. 64 - 72. doi:10.1109/MCE.2017.2776500

[9] Younes Bensouda Mourri, Łukasz Kaiser, Eddy Shyu. (2019). Natural Language Processing Specialization. Retrieved from Coursera: https://www.coursera.org/specializations/natural-language-processing.

[10] Walaa Medhat, Ahmed Hassan, Hoda Korashy,Sentiment analysis algorithms and applications: A survey,Ain Shams Engineering Journal, Volume 5, Issue 4, 2014, Pages 1093-1113, ISSN 2090-4479, https://doi.org/10.1016/j.asej.2014.04.011.

[11] W. Fengjiao and M. Aono, "Visual Sentiment Prediction by Merging Hand-Craft and CNN Features," 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, 2018, pp. 66-71, doi: 10.1109/ICAICTA.2018.8541312.

[12] Aihua Li, Jiede Wu, Zhidong Liu, Market Manipulation Detection Based on Classification Methods, Procedia Computer Science, Volume 122, 2017, Pages 788-795, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.11.438.

[13] Hu Xu, Bing Liu, Lei Shu, Philip S. Yu, "BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis," arXiv:1904.02232, 2019.

[14] Shervin Minaee, Elham Azimi, AmirAli Abdolrashidi, "Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models," arXiv:1904.04206v1, 2019.

[15] A. A. Karcioğlu and T. Aydin, "Sentiment Analysis of Turkish and English Twitter Feeds Using Word2Vec Model," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806295.

[16] Yadav, A., Vishwakarma, D.K. Sentiment analysis using deep learning architectures: a review. Artif Intell Rev 53, 4335–4385 (2020). https://doi.org/10.1007/s10462-019-09794-5.