Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020)
IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1

User Behavior Prediction using A Novel Sentence N-Gram Model

B. N. Shankar Gowda
Department of Computer Science & Engineering
Bangalore Institute of Technology
Bengaluru, India
bnsgowda@gmail.com

Dr. Vibha Lakshmikantha
Former Professor, Dept. of CS &E
BNM Institute of Technology
Bengaluru, India
vibhalak@gmail.com

***Abstract*— Humans normally use language contextually and make their decisions. This intelligence if transferred to a computer can assist in effective decision making. There is a need to replicate the contextual behavior of humans using web mining techniques. The aim of this work is to ingest real time data from social media data sources and harvest in a data warehouse. The raw data is preprocessed to eliminate the anomalies present raw data. The data is then parsed for creating user profiles. The N-Grams vocabulary is constructed and mapped with the synonym and abbreviation vocabularies. The user behavior is analyzed contextually using the user comments and their corresponding action in the past. The model is trained using the historical data extracted from the same sources. Various supervised machine learning techniques are used for predicting the succeeding comment and their performance is assessed. Conditional probability is assigned to each N-gram using the frequency of occurrence. The comment that has the highest probability value is predicted as the next comment. The performance analysis demonstrates that the prediction using ensemble approaches yield better results.**

Keywords*— Behavior Analysis, Language Modeling, Lemmatization, Machine Learning Techniques, N-Grams, Real Time Data Ingsetion.

I. INTRODUCTION

Hand held devices have become indispensable and companions of humans at home and work, to socialize, play and do business. Text entry into touch screen devices in particular can be cumbersome as they lack full-size keyboard. Automated text prediction can resolve this by predicting the next word when a text is entered. The ability to predict the succeeding action based on its preceding actions is one of the fundamental tasks of Natural Language Processing (NLP) [1]. The major focus of NLP is to reduce the interaction between human and computers. Machine learning for NLP involves using machine learning algorithms and "narrow" artificial intelligence (AI) to understand the meaning of text documents. The text documents can relate to social media comments, online reviews, survey responses, financial, medical, legal and regulatory documents. Text data requires a special approach to machine learning. Text data consists of many dimensions (words and phrases) but tends to be very sparse. For example, the English language has millions of words in common use but the comments or tweets contains only a few dozen of them.

Machine learning for NLP and text analytics involves a set of statistical techniques for identifying parts of speech, entities, sentiments, emotions, and other aspects of text [2]. The techniques can be expressed as a model that is then applied to other text, known as supervised machine learning. It could be a set of algorithms that work across large datasets to extract their meaning, which is known as unsupervised machine learning. Data sciences are increasingly making use of NLP combined with statistical methods to characterize and leverage the streams of text-based data that are not inherently quantitative [3]. The sequence of words that are used for predicting the next word is referred to as N-Gram, where N is a value which may be a unigram, bigram, trigram and quad gram. Probability is assigned to each N-gram for selecting the next word which has the highest probability value. This probability values are based on the sequence of the previous words. Language modelling (LM) is the process of assigning a probability to sentences in a language [4]. LM's can be classified into two categories as Statistical or count-based models and continuous-space models.

Statistical language models (SLM) are probabilistic models that predict the next action or sentence contextually, using the sequence of N-grams that precede it. The language model learns the probability of occurrence of given sentence based on the historical dataset. The language models mainly operate at the level of words rather than sentence.

A prediction system will choose the most suitable word for the user based on the context of the present sentence using statistical techniques. The key features of this work are

- To transfer real time data from the social media portals into a data warehouse repository.
- To build a dynamic N-Gram vector, synonym and abbreviation vocabulary.
- To contextually classify the social media data (blogs, tweets, comments, reviews) as positive, negative and neutral opinions.
- To analyze the performance of classical ML techniques for accuracy and scalability.
- To build a novel sentence N-Gram model to predict comments or sentences contextually in real time with minimal latency.

II. RELATED WORK

A brief literature review related to the proposed research work is presented below.

Steven Tang et al., in [1] has proposed an N-gram technique and recurrent neural network (RNN) methods to automatically predict the next word using the previous word. This method was experimented on the MOOCs (massive open online courses). Erik Cambria at el., in [2] has presented a review on latest development in the area of natural language processing to view the past, present and the future of the NLP in the new research direction. Here authors have discussed about the three curves of NLP namely syntactic, semantic and pragmatics of the text to be analyzed. Abinash Tripathy et al., in [3] has developed a model for classifying the human sentiment mined from the reviews and blogs of the social networking sites using ML algorithms such as naïve bayes, maximum entropy, stochastic gradient descent and SVM.

Oren Kolodny et al., in [4] have proposed a process level model of language acquisition. The model automatically learns the grammar and constructs the directed weighted graph. It is used for analyzing and predicting the next subsequent words. Er. Jyothi et al., in [5] has dealt with a recommendation system which is based on web usage data mining by using K nearest neighbor classification and ANN. The user's current interest is predicted using the user's previous navigational behavior and browsing activities. A hybrid model using KNN and RNN is proposed here. Pollyanna Gancalues et al., in [6] has compared and combined sentiment analysis methods. The authors have discussed about various sentiment analysis techniques such as lexical-based method and supervised ML methods. Aniello Castiglione et al., in [7] has proposed big data infrastructure to query, browse, analyze and process digital contents using Graph Analysis and ML methods.

Duy Duc An Bui et al., in [8] has discussed about a model for classification of clinical text data by learning the regular expressions. The authors have designed RED algorithm for classifying the clinical text data. Abdul Saboor et al., in [9] has presented a dictionary based three-step web-speller supported with a RDBMS database for word prediction. An accuracy of prediction of 92. 5%. Results depict that the word prediction increases the typing speed. Bo et al., in [10] have built a multi-granular CNN model for user profiling considering the social relations of users. The model extracts hidden semantic features from the message content. Barbara Calabrese et al., in [11] have surveyed various data extraction tools and the current trends. They have proposed a first architecture for behavior analysis integrating those tools.

III. PROBLEM DEFINITION

An important aspect of real time applications is the speed at which the next words are predicted. Some of the popular applications of text prediction are social networking sites and mobile communication. They demand that the load on the system and the speed of prediction has to be optimal.

The objectives are:

- To harvest real time data from social media sources into the data warehouse.
- To construct N-Grams vocabulary for each unique user and map with the synonym and abbreviation mapper.
- To analyze the contextual user behavior and classify the comments.
- To predict the succeeding comment of the web user based on behavior analytics.

A. Assumptions

To achieve a modest speed and high accuracy of prediction, the Sentence N-Gram model is limited to quad-grams and the number of words in a sentence should not be more than five.

B. Abbreviations and Acronyms

ANN: Artificial Neural Networks, DT: Decision Trees, ML: Machine Learning, NLP: Natural Language Processing. RED: Regular Expression Discovery RMSE: Root Mean Squared Error, SVM: Support Vector Machines, UAN: User Action Vector, UCV: User Comment Vector.

IV. ARCHITECTURE AND MODELING

The architecture framework of the prediction system is depicted in Figure 1. It has four phases. The first phase is about data acquisition, the second phase deals with pre-processing, the third phase deals with modelling the training and testing data and last phase yields the inferences.

A. Data Sourcing

The social network represents a potentially infinite source of user data, usable both for scientific and commercial applications [11]. The data source considered for this work is taken from the twitter DB and blog data (moneycontrol.com). A small fraction of large datasets is extracted to avoid hitting memory, performance and processing constraints.

B. Data Acquisition

The live data and also the historical data need to be extracted from the data sources. The historical data is used for training and testing, and live data is used for prediction. Data is ingested by using an automated tool "Apache NIFI" [7]. NIFI is configurable plug and play tool containing list of sources from where the data has to be extracted and the list of destination to where data to be handed over.

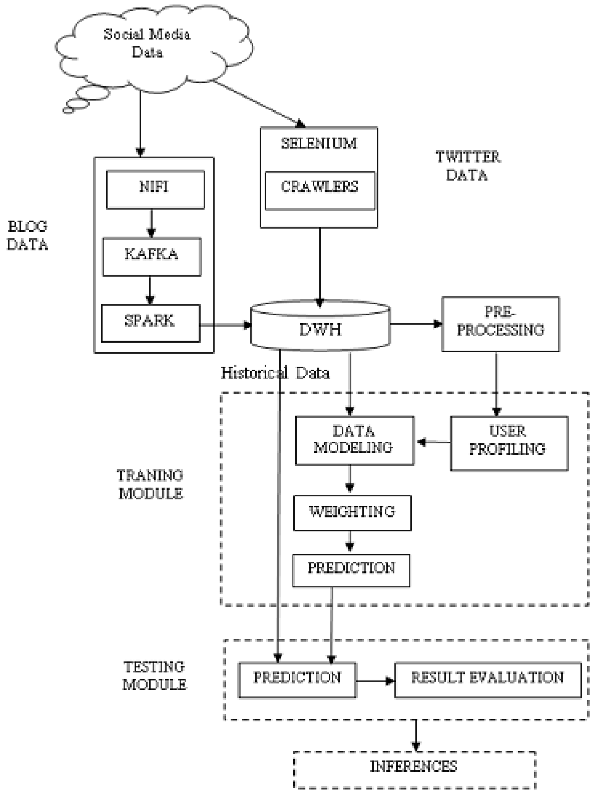

Figure 1: System Architecture

Another important parameter that should be mentioned is the context on which the data is to be extracted. The destination of NIFI could be middle ware like Apache KAFKA using which multiple tenants can access and use the data. Spark streaming mechanism is used to capture and transform the data, to make it compatible for the consecutive steps of the experiment. The transformed data is stored in apache HIVE. Hive is a data ware house on which big data processing techniques can be applied.

C. Data Preprocessing

It is the first step of data exploration; data is cleaned and transformed into data frame. Tokenization is the process of cutting the text into pieces; from body of text into sentences and words. Punctuation characters such as commas, parentheses are removed under the assumption that they have little impact on word order and n-gram composition. Numeric characters that may appear in numbers and dates are also removed. The combination of certain numbers and words can have predictive value, but their frequency might be too low to be useful. Regular Expressions can be used for text classification of the sample training data [8]. It performs the clean-up operation which reduces the size of the sample data. Keep it simple approach has been followed with the assumption that although the rigorous approach will tend to yield better predictive results, but it will consume more computational resources and it may not be significantly better for real time data.

D. User Profiling

The preprocessed data is segregated based on users in the data warehouse. The unique users are identified using IP address in web logs, unique_ID's in social media data web forum data. The comments or blogs posted by the users are influenced by social relations [10]. They are segregated and a vocabulary of N-grams say uni-gram, bi-gram, tri-gram and quad-gram are constructed. The comments are appended to the N-gram vocabulary dynamically and the frequency of occurrence of each N-gram is computed. The user profiles of all the users is stored in the data warehouse.

E. Data Modelling

N-gram techniques are used to generate and map comments and actions of the user describing the behavior or sentiment of the acting member. Map reduce techniques are used to tabulate the frequency of occurrences and classify them as unigram, bigram, trigram and quad-gram. The resultant records are partitioned into user comments and actions. These records act as input for the data modelling phase which are then used for training and testing. The novel predictive model uses single prediction techniques and ensemble prediction techniques.

F. Training and Testing

The resulting dataset is then split into training and testing datasets in the ratio of 80:20. The following exploratory analysis and model training will use the training dataset. The testing set will be reserved for the final model evaluation prior to application deployment. Sample of corpus data used for Training and Testing provides enough term associations that are able to for the next word prediction. To address the trade-off between performance and resource costs, consider only the most frequently occurring words in the corpus and ignoring the words that has less frequency. A corpus, or collection of sample text, is produced with each dataset using a consistent methodology. The real time comment posted by the user is ingested using automated tools. The comment is preprocessed and N-grams tokens are generated. The unique user is identified using their credentials. The length of comment is computed and compared with the corresponding N-grams word in the specific user profile. The comment that has maximum frequency in the action list is the probable next comment.

G. Prediction

Partitioning of records into comments and action to describe the user behavior. N-Gram prediction technique algorithm is used to build two vectors called comment vector and action vector. It considers the user comment as an input and predicts the next action using the historical data. The outcome of N-Gram algorithm predicts the actions based on the user comments. It then maps the comments with actions.

V. Proposed Methodology

Input

The data required for training and testing the model is obtained from blogs and tweets.

Output

Instantaneous prediction of user behavior (next tweet/blog/) with minimal latency.

Process

1. Load the Training dataset
   The system should load the sampled training dataset to the data frame.
2. Data Pre-processing
   Cleanse the data by removing punctuation characters such as commas, parentheses, numeric characters, profanity words and converting the text to lowercase.
3. Create user profiles using their unique ID's.
4. Parse input data and generate tokens *(uni-gram, bi-gram, tri-gram and quad-gram).*
5. Construct synonym and abbreviation vectors.
6. Generate N-gram & Frequency
   The basic N-gram model will take the n-grams of one to four words to predict the next tweet/blog/post/comment.
7. Generate the user comment table and the corresponding action table.
   Sentence N-gram conditional probabilities are computed using the formula for maximum likelihood.
8. Prediction
   Instantaneous prediction of succeeding tweet/blog/post/ comment.
9. Draw Inferences.

V. ALGORITHMS

The various algorithms proposed in this work are discussed in this section.

A. Algorithm for Data Pre-processing

The Data pre-processing algorithm is shown in Table 1. The raw data is extracted from the social media files using automated Data ingestion tools. The incomplete, inconsistent and redundant data is eliminated to obtain a good quality dataset for behavior analysis. The resulting output dataset are saved in the data warehouse.

Table 1: Algorithm for Pre-processing

```
Input: Raw Blog/Tweet data from social media
Output: Cleansed Data in Data warehouse
Method:
For each record in blog/tweet
Read Fields
        If records={*.gif,*.jpg, *.css}
        >> remove record
        Else
        Remove Whitespace, Punctuation, Stop
        words, Emojis, Profanity words & numbers.
        Convert text to lowercase
        >> save records
      Endif
Repeat until no more records
```

B. Algorithm for Uni-Gram Vector Generation

The algorithm for Uni-Gram Vector Generation is shown in Table 2. The input data is read from the preprocessed data in data warehouse. The data is parsed and uni-gram tokens are generated. The unique uni-grams are extracted and appended to the vocabulary.

Table 2: Algorithm for Uni-Gram Vector Generation

Input: Pre-processed data
Output: Uni-gram vocabulary
Method:
DistinctUnigram[ ] = select distinct words from $P1_u$
For each Distinct Unigram Word DUW_i in
DistinctUnigram[ ]
        UnigramFrequency[DUW_i] = 0
End For
For each Distinct Unigram Word DUW_i in
DistinctUnigram[ ]
    For each word W_i in $P1_u$
        If W_i = = DUW_i
        UnigramFrequency[DUW_i] =
UnigramFrequency[DUW_i] + 1
        End if
    End For
End For

C. Algorithm for Tri-Gram Vector Construction

The algorithm for Tri-Gram Vector Generation is shown in Table 3. The input data is read from the preprocessed data in data warehouse. The data is parsed and Bi-gram tokens are generated. The unique Quad-grams are extracted and appended to the vocabulary.

Table 3: Algorithm for Tri-Gram Vector Generation

Input: Pre-processed data
Output: Tri-Gram vocabulary
Method:
/* Quad-gram Word frequency count */
DistinctQuadgram[ ] = select distinct words from $P4_u$
For each Distinct Quadgram Word DQW_i in
DistinctQuadgram[ ]
        QGF [DQW_i] = 0
End For
For each Distinct Quadgram Word DQW_i in

```
DistinctQuadgram[ ]
   For each word Wi in P4u
       If Wi = = DQWi
       QGF [DQWi] = QGF [DQWi] + 1
       End if
   End For
End For
```

D. Algorithm for Synonym Vector Construction

The supervised machine learning techniques analyses behavior using the adjective or adverb in the comment. The classifier will not be able to determine the sentiment of that comment accurately if the dictionary does not have that word. A Dictionary of synonyms is built using online WordNet for each N-Grams keyword identified in the comment. Hence, through synonym mapping, the various inflected forms of a word are grouped together contextually. This process is also called as lemmatization or word stemming in NLP. The algorithm for constructing Synonym vector is as shown in Table 4.

Table 4: Algorithm for Synonym Vector Construction

```
Input: Tokenized uni_grams
Output: Synonym Vocabulary
Method:
/* Build Synonym Vocabulary */
For each unigram Wi in P1u
        Find synonyms using worknik API for the word Wi
        Update the synonym vocabulary SV[ ]
        End For
For each word Wi in P1u
        If Wi is not in DistinctUnigram[ ]
                Find synonyms for Wi using SV[ ]
                Update SV[ ]
        End if
End For
```

E. Abbreviation Vector Construction

Abbreviations are largely used in social media interaction to express sentiment. Since there are large number abbreviations, not all of them will appear as part of the vocabulary. Hence, a vocabulary of abbreviations and slangs are constructed and mapped with the N-grams vocabulary. The algorithm for constructing Abbreviation Vector is as shown in Table 5.

Table 5: Algorithm for Abbreviation Vector Construction

```
Input: Tokenized uni_grams
Output: Abbreviation Vocabulary
Method:
      /* Abbreviation Mapping */
      For each word Wi in P1u
       If Wi is not in wordnik API
          If Wi exists in AB [ ] then Map Wi to AB [i]
           Find synonyms for AB[i] using wordnik API
           Update SV[ ]
         End if
       End if
     End For
```

F. Algorithm for Constructing User Comment Vector

The user comments are read from the data warehouse. The comments are parsed and the length of comments are extracted. If the length is more than 5-grams, the comments are ignored otherwise, they are compared with the comment vector. If the comment is unique, they are appended to the vector, otherwise the frequency of that comment is updated. The algorithm for constructing user comment vector is as shown in Table 6.

Table 6: Algorithm for Constructing User Comment Vector

```
Input: user comment from live data streams
Output: User Comment Vector [UCV]
Method:
Step 1: Read user comment from data warehouse
Step 2: Identify the user posting comment
Step 3: Parse user comment
Step 4: Extract the number of words in the comment.
If the number of words >5
then Remove comment.
Else
        If live_comment € UCV[]
         Frequency = Frequency in UCV[] + 1
         Else
         UCV[] = UCV[] + Live_comment
         Endif
        Endif
```

G. Algorithm for Constructing User Action Vector

The historical user comments are harvested in the data warehouse from the social media data sources. The data so extracted is contextually classified as positive, negative or neutral classes using supervised learning techniques. The $(n-1)^{th}$ comment posted by the user is appended in the User Comment Vector [UCV] and the succeeding n^{th} comment posted by the user is appended in the User Action Vector [UAV]. The frequency of occurrence of $(n-1)^{th}$ comment followed by n^{th} comment is recorded. The comments are then compared with the appropriate N-grams. The succeeding action of the comment with maximum occurrence will be predicted as the next comment.

Table 7: Algorithm for constructing User Action Vector

```
Input: user comment from historical datasets
Output: User Action Vector [UAV]
Method:
Step 1: Read historical comment from DWH
Step 2: Identify the user posting comment
Step 3: Read nth comment
Step 4: Append in the UCV[]
```

Step 5: Read the (n+1)th comment
Append it in the UAV[] corresponding to nth comment.
Update the frequency of occurrence
Frequency = Frequency in UAV[] + 1
Step 6: Repeat for all comments.

VI. PERFORMANCE ANALYSIS

The proposed work aims to effectively predict the succeeding action of the web user by analyzing their behavior. A novel architectural framework is built for ingesting real time data from the social media data sources. The comments of users are pre-processed and then segregated as per the user profiles. Appropriate N-Gram vectors are constructed using the comments posted by users. A synonym and abbreviation vectors are constructed for better coverage and increasing the accuracy of prediction. A novel User comment Vector [UCV] and the corresponding User Action Vectors [UAV] are constructed using the historical data, extracted from the same data sources. The historical dataset is randomly split in the ratio of 80:20. The larger chunk is used for training the model and the remaining 20% is used for testing the model. The frequency of occurrence of the User action for the given user comment is measured and the action that has the maximum likelihood is predicted to be the next comment to be posted by the user.

A. Exploratory Analysis

Data is computed for 1% of the input size which is around 200 MB of Twitter datadase. The sampled output of this uni-gram vocabulary along with its frequencies generated from the "training" dataset is as depicted in Figure 2.

N-Gram	Frequency
again	14895
argree	12542
arrive	9586
bought	9458
bull	9111
welcome	7821
thanks	5631
right	4254
time	4096

Figure 2: Sampled Uni-gram Frequency

B. Language Coverage

The Ratio of number of unique n-grams and the associated total number of N-grams occurrences in each training scenario is depicted in Table 8. The experimental results demonstrate that the language coverage is expected to increase with more words being included in the vocabulary. A low ratio depicts that the frequency of occurrence of N-grams is less in the sampled training set. Ratio is computed using Equation number 1.

Ratio= Total number of instances/ unique N-Grams (1)

The results demonstrate that higher the frequency of n-gram occurrence, the language coverage ratio increases exponentially. As the frequency of occurrence of words become less, the curve tends to be linear.

Table 8: Ratio of Unique N-grams to Total N-grams

Sample set	N-Gram	Unique N-Gram	Total Instances	Ratio %
set1	Uni-Gram	25685	173208	14.8
set1	Bi-gram	25685	173208	14.8
set1	Tri-Gram	25685	173208	14.8
set1	Quad-Gram	25685	173208	14.8

C. Supervised Learning Techniques for Prediction

The Root Mean Square Error (RMSE) of the various single predictor is detailed in Table 9. The Machine Learning model was trained and evaluated with single predictors.

Table 9: Prediction Techniques

Prediction Technique	Quad -Gram	Tri- -Gram	Bi Gram	Uni -Gram
ANN	0.19	0.13	0.09	0.09
DT	0.14	0.12	0.1	0.09
Lasso	0.25	0.19	0.13	0.1
LR	0.12	0.11	0.09	0.08
Logistic	0.16	0.14	0.13	0.11
RF	0.19	0.15	0.14	0.12
SVM	0.19	0.16	0.15	0.11

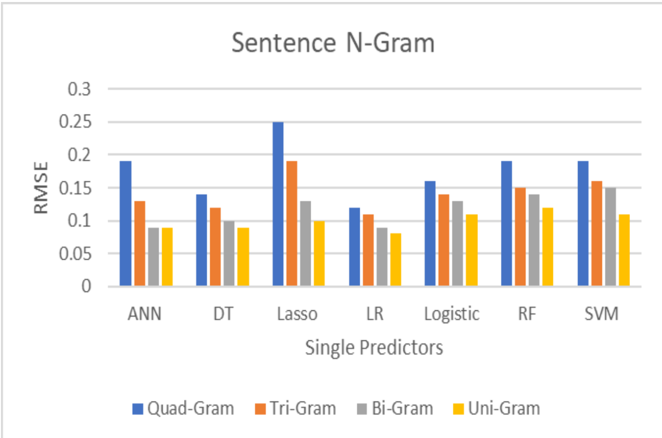

Figure 4: RMSE of Single Predictors

D. Prediction Table

The table 10 depicts the Prediction that includes predictor, frequency and the computed conditional probabilities sorted in descending order of probability. The comment with the highest probability will be predicted as the succeeding action by the user.

Table 10: Prediction table

N-Gram	Frequency	Predicted	Probability
again	14895	again	66.9
agree	12542	agree	65.1
arrive	9586	arrive	59.2
bought	9458	bought	53.1
bull	9111	bull	49.8

E. Root Mean Squred Error (RMSE)

The quality of a model built can be assessed based on the accuracy of prediction value.

Table 11: RMSE

Sample set	Predicted value	Actual value	Accuracy %	Error %
Set 1	135	123	91	9
Set 1	154	135	88	12
Set 2	125	119	95	5
Set 2	357	297	83	17
Set 10	160	137	86	14
Set 10	364	305	89	11

The RMSE is the ratio of correct predictions and total predictions. The test is done for the different training set sample sizes. The model RMSE varies between 6 percent and 11 percent for quad-gram. Varying the training set sample size between 1 and 10 percent does not seem to have a significant effect on prediction accuracy. It is to be investigated whether a larger sample size will increase accuracy.

F. Time Complexity

The applications of behavior analysis and the corresponding action of user are in social networking sites and mobile communication. Hence, the prediction of comments for any real time must have minimal latency. The results of prediction are as shown in Table 12 demonstrates that the time taken to predict the succeeding action of the user comment increases exponentially with higher order N-grams. With the increase in the number of records, the time taken also increases exponentially.

Table 12: Time taken for Prediction

% of records	Uni-Gram	Bi-Gram	Tri-Gram	Quad-Gram
1	9	14	16	18
2	17	26	27	29
5	79	124	145	190
10	184	299	348	447

VII. CONCLUSION

The result shows that N-Grams language prediction model has proven to be efficient for real time applications. It is observed that the accuracy of this model improves as the value of N-Grams increases. An efficient word prediction model can improve the quality and quantity of text generated for people with language impairments and those with learning disabilities.

VIII. REFERENCES

[1] Steven Tang, Joshua C. Peterson and Zachary A. Pardos "Predictive Modelling of Student Behavior Using Granular Large-Scale Action Data", The Handbook of Learning, Berkeley, USA doi:10.18608/hla17.019, 2017.

[2] Erik Cambria and Bebo White, "Jumping NLP Curves: A Review of Natural Language Processing Research", in IEEE Computational Intelligence Magazine, doi: 10.1109/MCI.2014.2307227, 2014.

[3] Abinash Tripathy, Ankit Agrawal and Santanu Kumar Rath, "Classification of Sentiment Reviews Using N-Gram Machine Learning Approach", doi.org.10.1016/ jeswa. 2016.03.028, vol 57, pp 117-126, 2016.

[4] Oren Kolodny, Arnon Lotem and Shimon Edelman, "Learning a Generative Probabilistic Grammar of Experience: A Process-Level Model of Language Acquisition", in Cognitive Science, doi.org/10.1111 /cogs.12140, vol 39, Issue 2, 2015.

[5] S. Kaviarasan, K. Hemapriya and K. Gopinath, "Semantic Web Usage Mining Techniques for Predicting Users' Navigation Requests", in International Journal of Innovative Research in Computer and Communication Engineering", vol. 3, Issue 5, 2015.

[6] Pollyanna Gonçalves, Matheus Araújo, Fabrício Benevenuto and Meeyoung Cha, "Comparing and Combining Sentiment Analysis Methods", in the first ACM conference on Online social networks, ISBN: 978-1-4503-2084-9, pp 27-38, USA, 2013.

[7] A. Castiglione, F. Colace, V. Moscato and F. Palmieri, "CHIS: A Big Data Infrastructure to Manage Digital Cultural Items", in Future Generation Computer Systems, doi.org/ 10.1016/j.future.2017.04.006, vol 86, pp 1134-114, 2018.

[8] Duy Duc An Bui and Qing Zeng-Treitler, "Learning Regular Expressions for Clinical Text Classification", in the Journal of the American Medical Informatics Association, vol 21, Issue 5, pp 850–857, 2014.

[9] Saboor, Abdul, Mihaly Benda, Felix Gembler, and Ivan Volosyak. "Word Prediction Support Model for SSVEP-Based BCI Web Speller" in International Work-Conference on Artificial Neural Networks, pp. 430-441. Springer, Cham, 2019.

[10] Bo Xu, Michael M. Tadesse, Peng Fei, and Hongfei Lin, "Multi-granularity Convolutional Neural Network with Feature Fusion and Refinement for User Profiling", in China Conference on Information Retrieval, pp. 161-172. Springer, Cham, 2019.

[11] Barbara Calabrese, Mario Cannataro, Nicola Ielpo, "Using Social Networks Data for Behavior and Sentiment Analysis", in 8th International Conference, IDCS 2015, pp.285-293, Windsor, UK, 2015.