International Journal of Formal Sciences: Current and Future Research Trends (IJFSCFRT)
ISSN (Print) , ISSN (Online)

https://ijfscfrtjournal.isrra.org/index.php/Formal_Sciences_Journal/index

Automatic Student Affective State Detection from Plain Text

Dan O. Anne[a]*, Agnes Chepkemoi[b], Elizaphan Maina[c]

[a]*Kenyatta University, P. O. Box 7936-00100, Nairobi, Kenya*
[b,c]*Kenyatta University, Nairobi, Kenya*
[a]*Email: danoyuga@gmail.com*
[b]*Email: chepkemoiagnes@gmail.com*
[c]*Email: maina.elizaphan@ku.ac.ke*

Abstract

We explore the concept of automatic detection of affective state of a learner in an e learning environment. We propose a model to detect the emotion from learners' text. We employ machine learning algorithms with ISEAR data and twitter data from Kaggle data repository. We follow the conventional steps of natural language processing; text preparation, feature extraction and emotion detection and classification. For text preparation we use processes of tokenization and segmentation, noise removal and segmentation. We extract features using count vectors and term frequency -inverse document frequency. For classification compare varied machine leaning algorithms. Results show that Linear SVM using Count Vectors accuracy gave an accuracy of 79% which is encouraging. We deduce that we can extract the affective states of the learners automatically from text during their interaction with e-learning environment. This will help in understanding the learners needs and help in enhancing adaptability

Keywords: Affective state; e-learning; ISEAR data; Machine Learning.

1. Introduction

In e- learning characteristics of the learner are quite important so as to determine the effectiveness of the learning process employed [1, 2] cites learner characteristics as major components in designing e-learning platforms. Moreover, learning theories places learner characteristics at the center of learning[3].

----

* Corresponding author.

The exploration of learner characteristics and incorporating it into e-learning platform is to enable adaptability and enhance nontraditional classroom learning environment[4, 5]In their study on the learner characteristics, they looked on multi-dimensionality aspect of learner characteristics and placed the learner characteristics as construct of social presence. They found out that learner characteristics have a strong impact on social presence. One of the major learning characteristics being studied in relation to e- learning environment is the emotional state of the learner as the learner interacts with the learning environment. Reference [6] in reviewing emotion aware systems for e- learning emphasizes the importance of learner emotions in e-learning environment and notes that there are inadequate researches on the topic and advances for more work on emotion aware e-learning platforms. Many studies have modeled emotion extraction techniques [7,8,9,10, 11, 12] all use facial expression and user movements to detect emotion of an individual. Reference [13] proposed text based intelligent emotion detection using the dominant meaning technique to user's emotion in any typical environment. In their experiment they used the ISEAR data to classify to create emotion trees then. Reference [14] explored empirical machine learning techniques in emotion detection from text. Their focus was on children literature and their objectives were to help in rendering the literature. Learning theories; constructivism, behaviorism and Cognitivism [15] advocates for adaptively of learning systems. By getting emotional state of learner we will be advancing the theories of learning

2. Related Work

2.1 Text preparation

Natural language processing uses text data for computation processes. Use of raw data in scientific computing can pose a great challenge if the data to be processed is not well cleaned and presented in format that can be processed [16]. There are quite a number of processed that can be used to clean data which are guided by the dataset to be used for training and the actual data expected during the real time processing of the text[16]. The process of data preparation[17] aims at transforming text into a input parameter of palatable form for effective and efficient performance of machine learning algorithms acting on it [18].

Tokenization and segmentation

Tokenization also referred to as lexical analysis is a process that splits larger texts into words while segmentation is the splitting larger texts into smaller e.g. paragraphs into sentences[19]. Sentence boundary detection techniques[20] and application of openNLP software [21] software are major processing steps that have been elevated in tokenization and segmentation of plain text. result.

Stemming

A normalization process of producing morphological variants of a root word devoid of its derivational and inflectional affixes[22]. The process transforms a word to its base or root form. The end goal of stemming is to identify word forms which differ in irrelevant ways and merge them so as to be treated as equivalent. Stemming is one of the approaches of achieving conflation which is a process where two distinct words, expressions or phrases are treated as semantically equivalent. Quite a number of algorithms have been proposed over time to implement stemming. Some of the algorithms include the Paice/Husk stemmer, Lovins' stemmer, stemmer KSTEM and Martin Porter's algorithm. All the stemmers, the algorithms that implement stemming, operations are subjected to acceptability constraints which ensures that indicated action in algorithm is only taken after satisfying specified conditions.

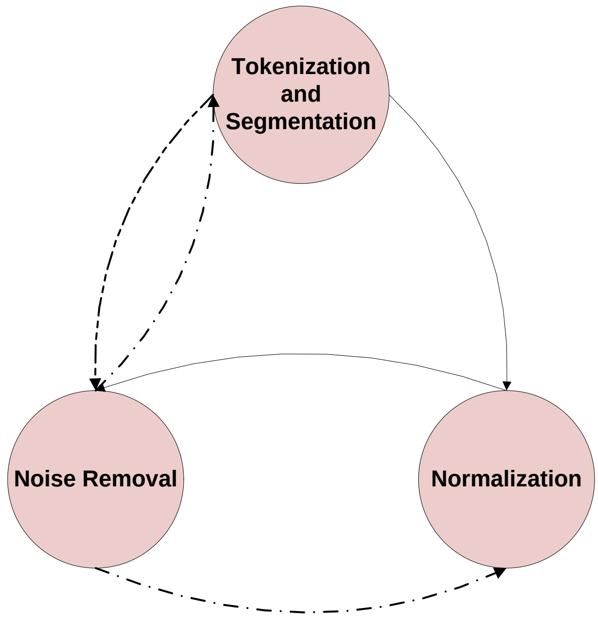

Figure 2.1: Data Preparation Process

Normalization

This involves leveling all the text to same footing to allow uniform processing during computational processes [23]. Normalization activities encompasses converting all characters of given word to its lower-case equivalent, removing punctuation marks, Lemmatization- The process of reforming a given or specific word to its root or lemma[24, 24], Stemming-The processes of removing affixes in a given word, converting numbers to their equivalent words amongst others [25, 26].

Noise Removal

Noise removal involves elimination of unnecessary characters in the texts to help in cleaning up such texts e.g. removal of extra whitespaces, html tags, special characters amongst others

Parts of Speech (PoS) Tagging

Part-of-speech(Pos) tagging is a disambiguation process that involves assignment of parts of speech maker to each word in a tokenized sequence of words and a tag set text [27]. The most used approaches for PoS tagging are generative approaches like Hidden Markov Models tagging and discriminative approaches like Maximum Entropy Markov Models and Neural approaches. In this research we employ Hidden Markov Pos Tagging

2.2 Feature Identification and Extraction

We deploy both the Term frequency- Inverse Document Frequency (TF-IDF) & Count Vectors for extracting the affective state features from a given text. These are chosen because we need numeric data for computational purposes

Term Frequency- Inverse Document Frequency

Term frequency–inverse document frequency(TF-IDF), is a weighting numerical statistic method employed in information retrieval process and is intended to reflect the importance of a given specific word from a document in a collection or corpus[28]. The use of TF-IDF which is a feature extraction algorithm, begins with calculation of the term frequency (TF) which is the word frequency followed by calculation of IDF which is the inverse document Frequency[29] hence Term frequency–inverse document frequency(TF-IDF) The variants involved include calculating the how count of term t in a document. According to [30], the more a term/word appears is directly proportional to its importance. Term Frequency is expressed as the count of occurrence of term t occurs in document d. The raw count is given by $f_{t,d}$, therefore tf scheme is given by $tf(t,d) = f_{t,d}$ [31].

N-Grams

We apply the TF-IDF to word n-gram. The n-grams are created to predict the probability of the next word given the previous word [32]. This in reference to Markov's Assumption. [33] explains Markov models as a class of probabilistic models, models that mathematical probabilities to express uncertainty and noise associated with it. With this assumption we can predict the probability of some future unit without looking too far into the past. We do this by generalizing the bigram, looking of one word into the past[34] to the trigram, looking for two words into the past and thus to the n-gram which looking n-1 words into the past [35] We compute $P(w|h)$, the probability of a word w given some history h. take history h is "*I lost so much money that*" and we want to know the probability that the next word is *the*:

$$P(the|I\ lost\ so\ much\ money\ that).$$

In estimating the probability, we get relative frequency counts: take a very large corpus, count the number of times *I lost so much money* that appears, and compute the number of times this is followed by the, giving the history h appearance count, the count of times was it followed by word w", and represented as below

$$P(the|\ I\ lost\ so\ much\ money\ that) = C(I\ lost\ so\ much\ money\ that\ the)\ C(I\ lost\ so\ much\ money\ that)$$

2.3Affective state detection and Classification

Reference [36] refers to affective state classification as the classification of emotional state feature of a given plain text. A number of classifications algorithms have been put forth to do classification of text emotionality and have displayed varied results as put forth in[37,38]. The results are most of the time dependent on text being analyzed the features in that text. The emotional state of text is measured using the emotional state adjectives like sad, happy, glad, joy amongst others[39]. These emotional adjectives most of the times are classified in three broader categories namely positive, negative and neutral. Reference [40] enlisted current researches on extraction of emotions from text and found out that machine learning algorithms are the best approaches for emotion detection. This was also a firmed by[41] who grouped the approaches of emotional text detection into the machine learning approach, the rule construction approach and the hybrid approach which combines both approaches. The most commonly used machine learning algorithms include neural network and fuzzy logic[42], Vector space model[43], Support Vector Machine (SVM) and Long-Short Term Memory (LSTM) [44, 44,45], Naïve-Bayes[46,47,48] and Random Forest Classification [49,50] . AI techniques have been used in two ways; one is for classifying learners into groups to provide adaptation to those particular groups, two is for diagnosing the learner characteristics as learners learn so as to adjust the instruction method. Fuzzy logic is seen as an extension of set theory, Fuzzy logic is usually used to assess learning and knowledge of the learner. It has been used in several studies to make adaptation based on learner's knowledge. Reference [51] used fuzzy logic to extract rules from learner data so that they could tell the knowledge needs of learners [52] , use fuzzy logic to automatically generate the domain model of the adaptive e-learning systems [53], use a type-2 fuzzy logic in order to learn the preferred knowledge delivery method of learners so that the content can be adapted to suit the learners knowledge delivery method [54], uses fuzzy logic to identify and update the learner's knowledge level so that they could adapt the kind of advice given to learners based on their knowledge levels. Though fuzzy logic has been used in many adaptive systems, its application is limited to assessing, updating learner knowledge. Bayesian networks are directed acyclic graphs which are usually used for modelling variables probabilistic dependencies[55]. Bayesian networks have been used in adaptive systems in order to provide adaptive instruction. For instance [55], uses Bayesian networks to assess the learner knowledge and provide instruction as per the learner knowledge [56], uses Bayesian network to classify users based on their navigation habits and then suggest content based on the classification [57], uses Bayesian network to provide learning path adaptability by first constructing the domain module using a Bayesian network [58], uses Bayesian network to provide motivational messages based on the learner logs. Bayesian networks are only applicable in specific and conditional places where the situations can be converted into probabilistic reasoning. Bayesian networks are also rule based; they therefore depend on the ability to be able to generate accurate learning rules; Hidden Markov models have been used in adaptive e-learning systems. For instance [59] used the K-means algorithm together with the Hidden Markov models to cluster learners into different learning styles and adapt content to suit the learner learning style [60], used fuzzy petri nets and hidden markov model to adapt learning content to each learner in accordance with the learner's learning path. Hidden markov models have the disadvantage of having low accuracy for the low order ones and requiring complex computation for the high level ones [61].One of the major uses of ANN is pattern recognition, this ability of neural networks being able to model human conduct makes it suitable to be used in modelling learners.

3. Methodology

Our proposed model of auto extraction of affective state from text is divided in four components; the text preparation, feature extraction, model training, and classification as shown in figure 3.1.

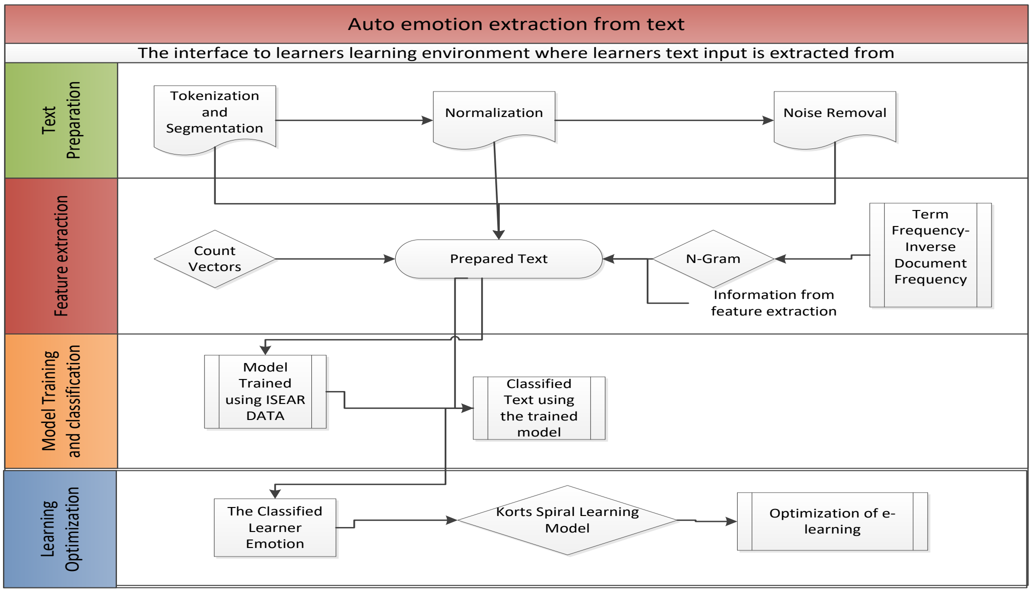

Figure 3.1: Auto Affective state extraction from text

Dataset

Our dataset had 40,000 tweets in total, labeled into seven different human emotions. The emotions included Anger, Disgust, Fear, Guilt, Joy, Shame and Sadness. We considered two broad sentiments in our experiment for affective state extraction; happiness and sadness. These two sentiments constituted 10,000 tweets from the entire data sample**.**

3.1 Text preparation

We deploy a number of data preparation processes to bring out some uniformity to the text by making everything lowercase, removing punctuation, and stop words (like prepositions).

Tokenization and segmentation

We applied word tokenization and to resolve the Out of Vocabulary (OOV) problem we used Byte Pair Encoding (BPE) which segments OOV into sub words and then represent the word into these sub-words

Steps in word tokenization using BPE

Begin

Take a corpus:

Step 1. Add </w> at the end of every word in the corpus

Step 2 Tokenizing words to characters:

Step 3. Assign vocabulary to Initial values:

Loop One:

Step 4. Calculate the frequency:

Step 5. Merge the highest frequency combination:

Step 6. Save the best combination:

Repeat steps 4-6 for every Loop.

loop Two:

Step4. Calculate frequency:

Step 5. Merge the highest frequency combination:

Step 6. Save the best combination:

end

Stemming

To achieve conflation, we use Porter's algorithm which by design reflects suffixes linguistic structure as defined in English words. We examine inflexions of verbs, plurals and words ending with 'y', and explore derivational endings and finally remove 'e' and final'-ll' singling.

Stemming Rules

Rules are of the form ***condition (St1 → St2),***

St1 and St2 -> are suffixes, considering a give set of rules, the longest matching suffix St1 applies.

Conditions in Stemming:

- $\boldsymbol{m \rightarrow measure\ of\ the\ stem\ m\ =\ k\ or\ m\ >\ k},$
  - $\boldsymbol{where\ k\ is\ an\ integer}$
- $\boldsymbol{2.*\ x \rightarrow the\ stem\ ends\ with\ a\ given\ letter\ x}$
- $\boldsymbol{3.*\ v * \rightarrow the\ stem\ contains\ a\ vowel}$
- $\boldsymbol{4.*\ d\ \rightarrow the\ stem\ ends\ in\ double\ consonant}$
- $\boldsymbol{5.*\ o\ \rightarrow the\ stem\ ends\ with\ a\ consonant - vowel - consonant\ sequence, where\ the\ final\ consonant\ is\ not\ w, x\ or\ y, (e.g., will, hop)}$

Porters Algorithm Steps

Each step corresponds to a set of rules. The rules in a step are examined in sequence, and only one rule from a step can apply

{

- $\boldsymbol{step1(word)}$;
- $\boldsymbol{step2(stem)}$;
- $\boldsymbol{if\ (the\ second\ or\ third\ rule\ of\ step\ 2\ was\ used)}$
  - $\boldsymbol{step21(stem)}$;
- $\boldsymbol{step3(stem)}$;
- $\boldsymbol{step4(stem)}$;
- $\boldsymbol{step5(stem)}$;
- $\boldsymbol{step6(stem)}$;
- $\boldsymbol{step7(stem)}$;
- $\boldsymbol{step8(stem)}$;
- }

Normalization

We convert all characters of given word to its lower-case equivalent, removing punctuation marks,

- $> tokens\ =\ ['House','Visitor','Center']$
- $> normalized\_tokens\ =\ [x.lower()\ for\ x\ in\ tokens]$
- $> print(normalized\_tokens)$
- $> ['house','visitor','center']$

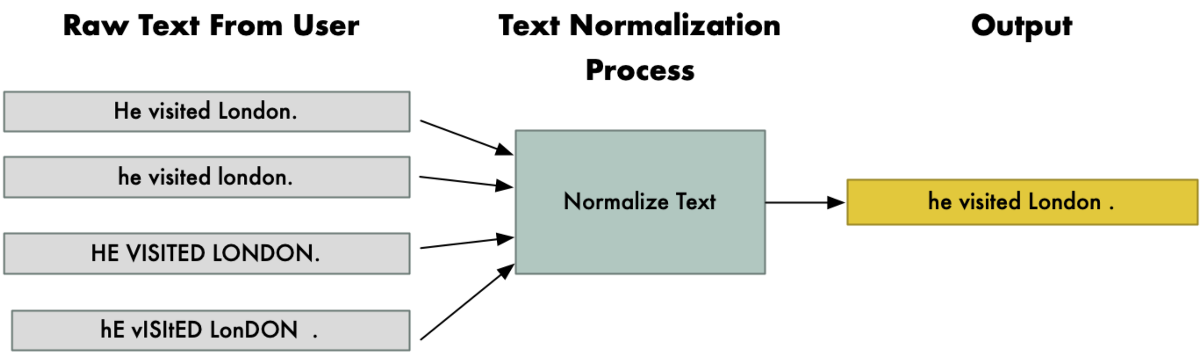

Figure 3.2: Normalization process

Noise Removal

we of extra whitespaces, html tags, special characters amongst others

$$Example\ 1$$

$$\begin{aligned}
&text\ =\ " \quad This\ is\ a\ paragraph\ \ "\\
&result\ =\ re.sub(r'<.?p>', '', text)\\
&print(result)\\
&\#\quad This\ is\ a\ paragraph
\end{aligned}$$

Parts of Speech Tagging (POS Tagging)

We employ generative approach, Hidden Markov Models tagging and discriminative approach, Maximum Entropy Markov Models and Neural approaches.

3.2 Feature Extraction

We deploy both the Term frequency- Inverse Document Frequency (TF-IDF) & Count Vectors for extracting the affective state features from a given text. These are chosen because we need numeric data for computational purposes

Term Frequency- Inverse Document Frequency

The term frequency scheme is given by $tf(t,d)\ =\ f_{t,d}$

We perform normalization to get tf :

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

Where $f_{t,d}$ is the count occurrences of term t in a given document d, and $f_{t',d}$ is the count of occurrences of every term in document d.

To get the Inverse Frequency Document (IDF),

$$idf(t,D) = log\,\frac{N}{|\{d \in D\}: t \in D|} \tag{2}$$

where

N: count of documents in corpus $N = |D|$

$|(d \in D: t \in D)|$: count of occurrences of documents where the term *t* appears (i.e.$tf(t,d) \neq 0$).

N-Grams

We apply the TF-IDF to word n-gram. By generalizing the bigram, we compute $pr(i|j)$, probability of a word i given some history j. take history j is "*I lost so much money that*" and compute the probability pr that the next word is *the*:

$pr(the|I\ lost\ so\ much\ money\ that)$.

In estimating the probability pr, we get counts of relative frequency rf: take a very large corpus, count the number of times *I lost so much money that* appears, and count the number of times this is followed by *the*. This gives, out of the times the history i appeared, how many times was it followed by the word *j*", and represented as below

$$pr(the|\ I\ lost\ so\ much\ money\ that) \\ = C(I\ lost\ so\ much\ money\ that\ the)\ C(I\ lost\ so\ much\ money\ that)$$

For joint probability of an entire sequence of words like *so much money was lost*, we find "out of all possible sequences of five words, how many of them are *so much money was lost*?" by getting the count of *so much money was lost* and divide by the sum of the counts of all possible five word sequences. To get this we employ the use of chain rule. We formalize the notation equation[62]. The probability pr of random variable x_i taking the value "the", or $pr(xi\ =\ "the")$, $pr(the)$. Taking Sequence of N words $(\ i_1, i_2, \ldots, i_{n-1})$. For the joint probability $pr(x = i_1, Y\ = i_2, Z\ = i_3, \ldots, i\ = i_n)$ we use$pr(i_1, i_2, \ldots, i_n)$.

Therefore, the complete probability is given by $pr(i_1, i_2, \ldots, i_n)$?

Using chain rule to decompose we get

$$pr(x_1 \ldots x_n) = pr(x_1)pr(x_2|x_1)pr(x_3|x_1^2) \ldots pr(x_n|x_1^{n-1})$$

$$= \prod_{k=1}^{n} pr\left(x_k \mid x_1^{k-1}\right) \tag{3}$$

$$pr(i_1^n) = pr(i_1)pr(i_2|i_1)pr(i_3|i_1^2) \ldots pr(i_n|i_1^{n-1})$$

$$= \prod_{k=1}^{n} pr\left(i_k \mid i_1^{k-1}\right) \tag{4}$$

Given a corpus of previous words, we use the chain rule to compute the joint probability of a specific identified word and its conditional probability in relation to the corpus In equation 4, we compute the product of conditional probabilities to estimate the joint probability of the entire sequence. This however has limitations brought about by the creativity of language making it difficult for a particular context to occur. We have no way of computing the exact probability of a given word from a long sequence of proceeding words, $pr(i_n|i_1^{n-1})$ and we can use estimation and indicated above. The limitation above is cured by the use of n-gram model theory. Using the n-gram model we approximate the history by use of a few last words in the word sequence, $pr(x_1 \dots x_n)$.

In bigram model, it gets the approximated probability pr of a given word provided all the previous words $pr(i_n|i_1^{n-1})$ by using only the conditional probability of the preceding word $pr(i_n|i_{n-1})$.

i.e we don't calculate the probability

$pr(the|\ I\ lost\ my\ so\ much\ of\ my\ mother's\ money\ that)$

instead we approximate it with the probability

$pr(the|that)$

In use of bigram, we therefore make the following approximation in calculating the conditional probability of the next word;

$$pr(i_n|i_1^{n-1}) \approx pr(i_n|i_{n-1}) \tag{5}$$

We can now get the general n-gram approximation equation as

$$pr(i_n|i_1^{n-1}) \approx pr(i_n|i_{n-N+1}^{n-1}) \tag{6}$$

Substitute 5 into equation 4 for the probability of complete word sequence

$$pr\left(i_1^n\right) \approx \prod_{k=1}^{n} pr\left(i_k \mid i_{k-1}\right) \tag{7}$$

To estimate n-gram probabilities, we employ maximum Likelihood Estimation by getting Counts C from a corpus and Normalizing C to lie between 0 and 1

$$pr(i_n|i_{n-1}) = \frac{C(i_{n-1}i_n)}{C(i_{n-1})} \tag{8}$$

We Explored both TF-IDF & Count Vectors (This is for numeric data for the math!). Split the data into training and testing parts before performing feature extraction in the ration of 90:10.

3.3 Affective detection and classification

We used varied algorithms to train the models. These include Multinomial Naive Bayes Classifier, Linear SVM Classifier, Logistic Regression Classifier and Naive Bayes Classifier.

4. Experiment and results

Table 4.1: Data Preprocessing /Preparation sample

Lower case	text = "Data processing begins with preprocessing of data. An example is converting data into lower case" output: data processing begins with preprocessing of data. an example is converting data into lower case
Tokenization Removal of stop words	Output: ['Data', 'Processing', 'begins', 'with', 'preprocessing', 'of', 'data', '.', 'An', 'Example', 'is', 'converting', 'is', 'converting', 'data','into', 'lower', 'case'] *output:* ['data', 'preparation', 'process', '.', 'making', 'texts', 'ready', 'machine', 'learning', 'algorithms', 'processing']
Lemmatization	*output:* ['data', 'preparation', 'process', '.', 'making', 'text', 'ready', 'machine', 'learning', 'algorithm', 'processing']
Stemming- Taking the word to its root- Use of snowball stemmer algorithm	*output:* ['data', 'prepar', 'process', '.', 'make', 'text', 'readi', 'machin', 'learn', 'algorithm', 'process'] *output:* [('process', 2), ('data', 1), ('prepar', 1), ('.', 1), ('make', 1)]
POS TAG Deduce part of speech of words, eg. Nouns, verbs etc	*output:* [('data', 'NNS'), ('prepar', 'NN'), ('process', 'NN'), ('.', '.'), ('make', 'VB'), ('text', 'JJ'), ('readi', 'NN'), ('machin', 'NN'), ('learn', 'VBP'), ('algorithm', 'NN'), ('process', 'NN')]

Results

The Characteristics of ISEAR Dataset used for both training and Validation The ISEAR dataset has approximately 7600 statements from college students representing seven emotional states: Anger, Disgust, Fear, Sadness, Shame, Joy and guilt.

Table 4.2: ISEAR Data Description

Affective State (Emotion)	Count
Anger	1096
Disgust	1096
Fear	1095
Guilt	1093
Joy	1094
Sadness	1096
Shame	1096

Table 4.3: ISEAR Data Description

	Affective State	Description
0	joy	On days when I feel close to my partner and ot...
1	fear	Every time I imagine that someone I love or I ...
2	anger	When I had been obviously unjustly treated and...
3	sadness	When I think about the short time that we live...
4	disgust	At a gathering I found myself involuntarily si...
...	...	...
7511	shame	Two years back someone invited me to be the tu...
7512	shame	I had taken the responsibility to do something...
7513	fear	I was at home and I heard a loud sound of spit...
7514	guilt	I did not do the homework that the teacher had...
7515	fear	I had shouted at my younger brother and he was...

7516 rows × 2 columns

Table 4.4: ISEAR Data Description

	Affective State	Description
count	7516	7516
unique	8	7449
top	joy	When my grandfather died.
freq	1092	8

For models using the TF-IDF features-

Table 4.5: Classification with TF-IDF feature extraction

Classifier	Accuracy of Classification
Naive bayes- TF-IDF	0.529
SVM- TF-IDF	0.544
Random Forest- TF-IDF	0.539

From the results on table 4.5, Support Vector Machine emerges top with accuracy of 54.4 %. This is hardly a meaningful classification as it is just an average classification result. The error margin here is large such that we can deduce that the classification is unreliable. Further comparative analysis revealed that this might be because of the complexity and the nature of the textual, and presentation format of the dataset in use. The Fact that the algorithms used, Naïve Bayes, SVM and Random Forest are machine learning algorithms, the results could also be poor because of the sentences complexity in a given text.

For models using count vectors features-

Table 4.6: Classification with Count Vectors feature extraction

Classifier	Accuracy of Classification
Multinomial Naive Bayes - Count vectors	0.776
Linear SVM using Count Vectors accuracy	0.793
Logistic Regression- Count Vectors	0.785
Random Forest -Count Vectors	0.752

Use of Count Vectors for features extraction and Linear Support Vector Machine for classification yields 79.3% of accuracy as indicated in Table 4.6. This is an impressive improved which is safe enough to qualify as valid sentimental analysis result. This result however could be argued from the point of the nature and specificity of the data set used and a factor of dependency of emotional text being dependent on the presence of categorically significant adjectives within the corpus. The result could also change depending on the linguistic structure and the complexity of sentence in a given text.

5. Limitations of research

All the classifiers used in this research; Naïve Bayes, Logistic regression, Support Vector Machine and Random forest are algorithms based on machine learning and their accuracy largely depends on the training dataset. Determination of the training dataset is therefore paramount but was not explored in this research. Neither was the determination of amplification factor as in case of Naïve Bayes explored. These are topics for future research areas. The issue of complex sentences in a given text was also not fully addressed, yet it is important in determination of quality. ISEAR data set is based of research within the European region students. In plain texts, the sentence structures and body composition may differ linguistically based on continents and this needs to be factored in future research

6. Conclusion

In this paper we advocate for strong consideration of emotional aspects of learners by providers of e-learning platform. We provide a model of extracting the affective states of the learners automatically from the text they write when interacting with the e-learning platform then weigh it against the Korts Spiral learning model and though this the optimality of learning can be realized

References

[1] M. Nakayama, H. Yamamoto, and R. Santiago, “Investigating the impact of learner characteristics on blended learning among Japanese Students,” Proc. Int. Conf. e-Learning, ICEL, vol. 2006-Janua, no. 3, pp. 361–370, 2006.

[2] B. Ghirardini, E-learning methodologies: A guide for designing and developing e-learning courses. 2011.

[3] M. (Dalton S. C. Zhou and D. (Dalton S. C. Brown, Educational learning theories (Second Edition). 2015.

[4] A. Learning, A. I. Method, and O. Teaching, “Special Issue on Innovation 2018 Adaptive Learning : An Innovative Method for Online Teaching and,” vol. 56, no. 5, 2018.

[5] D. Mykota, “The impact of learner characteristics on the multi-dimensional construct of social presence,” Turkish Online J. Educ. Technol., vol. 16, no. 2, pp. 137–144, 2017.

[6] M. Feidakis, “A Review of Emotion-Aware Systems for e-Learning in Virtual Environments,” Form. Assessment, Learn. Data Anal. Gamification ICT Educ., no. March, pp. 217–242, 2016, doi: 10.1016/B978-0-12-803637-2.00011-7.

[7] M. Saneiro, O. C. Santos, S. Salmeron-Majadas, and J. G. Boticario, “Towards emotion detection in educational scenarios from facial expressions and body movements through multimodal approaches,” Sci. World J., vol. 2014, 2014, doi: 10.1155/2014/484873.

[8] N. Bosch, S. K. D’Mello, J. Ocumpaugh, R. S. Baker, and V. Shute, “Using video to automatically detect learner affect in computer-enabled classrooms,” ACM Trans. Interact. Intell. Syst., vol. 6, no. 2, 2016, doi: 10.1145/2946837.

[9] L. B. Krithika and G. G. Lakshmi Priya, “Student Emotion Recognition System (SERS) for e-learning Improvement Based on Learner Concentration Metric,” Procedia Comput. Sci., vol. 85, no. Cms, pp. 767–776, 2016, doi: 10.1016/j.procs.2016.05.264.

[10] M. El Kerdawy et al., “The automatic detection of cognition using eeg and facial expressions,” Sensors (Switzerland), vol. 20, no. 12, pp. 1–32, 2020, doi: 10.3390/s20123516.

[11] N. Bosch et al., “Automatic detection of learning-centered affective states in the wild,” Int. Conf. Intell. User Interfaces, Proc. IUI, vol. 2015-Janua, no. October 2018, pp. 379–388, 2015, doi: 10.1145/2678025.2701397.

[12] F. A. Khan, S. Graf, E. R. Weippl, and A. M. Tjoa, “Integrated Approach for the Detection of Learning Styles & Affective States Previous Research and Our Perspective.”

[13] M. A. Razek and C. Frasson, "Text-Based Intelligent Learning Emotion System," J. Intell. Learn. Syst. Appl., vol. 09, no. 01, pp. 17–20, 2017, doi: 10.4236/jilsa.2017.91002.

[14] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text," no. October, pp. 579–586, 2005, doi: 10.3115/1220575.1220648.

[15] J. Hammad, M. Hariadi, M. H. Purnomo, and N. Jabari, "E-learning and Adaptive E-learning Review," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 2, pp. 48–55, 2018.

[16] T. Jauhiainen, M. Lui, M. Zampieri, T. Baldwin, and K. Lindén, "Automatic language identification in texts: A survey," arXiv, pp. 1–103, 2018, doi: 10.1613/jair.1.11675.

[17] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning Based Text Classification: A Comprehensive Review," arXiv, vol. 1, no. 1, pp. 1–43, 2020.

[18] R. Dipanjan Sarkar, "Text Wrangling & Pre-processing: A Practitioner's Guide to NLP," 2018. https://www.kdnuggets.com/2018/08/practitioners-guide-processing-understanding-text-2.html.

[19] A. Toleu, G. Tolegen, and A. Makazhanov, "Character-based deep learning models for token and sentence segmentation," V международная конференция по компьютерной обработке тюркских языков «TurkLang 2017», no. 1, pp. 100–112, 2017.

[20] S. J. Putra, M. N. Gunawan, I. Khalil, and T. Mantoro, "Sentence boundary disambiguation for Indonesian language," ACM Int. Conf. Proceeding Ser., pp. 587–590, 2017, doi: 10.1145/3151759.3156474.

[21] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. Letraon, "A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate," 2019 6th Int. Conf. Soc. Networks Anal. Manag. Secur. SNAMS 2019, no. September 2020, pp. 338–343, 2019, doi: 10.1109/SNAMS.2019.8931850.

[22] W. Ben and A. Karaa, "A NEW STEMMER TO I MPROVE I NFORMATION RETRIEVAL," Int. J. Netw. Secur. Its Appl. (IJNSA), Vol.5, No.4, July 2013, vol. 5, no. 4, pp. 143–154, 2013.

[23] M. Bollmann, "A large-scale comparison of historical text normalization systems," NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, pp. 3885–3898, 2019, doi: 10.18653/v1/n19-1389.

[24] J. Kanerva, F. Ginter, and T. Salakoski, "Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks," Nat. Lang. Eng., 2020, doi: 10.1017/S1351324920000224.

[25] H. A. Almuzaini and A. M. Azmi, "Impact of Stemming and Word Embedding on Deep Learning-

Based Arabic Text Categorization," IEEE Access, vol. 8, pp. 127913–127928, 2020, doi: 10.1109/ACCESS.2020.3009217.

[26] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," IEEE Access, vol. 7, pp. 32664–32671, 2019, doi: 10.1109/ACCESS.2019.2903331.

[27] S. G. K. and S. S. Nandyal, "Survey on Parts of Speech Tagger Techniques," Int. Conf. Curr. Trends Towar. Converging Technol. (ICCTCT), Coimbatore, 2018, pp. 1-6, doi 10.1109/ICCTCT.2018.8550884., 2018.

[28] A. N. K. and I. Agustina, "Implementation of Information Retrieval Using Tf-Idf Weighting Method On Detik.Com's Website," 12th Int. Conf. Telecommun. Syst. Serv. Appl. (TSSA), Yogyakarta, Indones. 2018, pp. 1-4, doi 10.1109/TSSA.2018.8708744., 2018.

[29] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," Int. J. Comput. Appl., vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.

[30] N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm," Procedia Comput. Sci., vol. 161, pp. 509–515, 2019, doi: 10.1016/j.procs.2019.11.150.

[31] H. Mohammadhassanzadeh, I. Sketris, R. Traynor, S. Alexander, B. Winquist, and S. A. Stewart, "Using Natural Language Processing to Examine the Uptake, Content, and Readability of Media Coverage of a Pan-Canadian Drug Safety Research Project: Cross-Sectional Observational Study," JMIR Form. Res., vol. 4, no. 1, p. e13296, 2020, doi: 10.2196/13296.

[32] J. Piskorski and G. Jacquet, "TF-IDF Character N-grams versus Word Embedding-based Models for Fine-grained Event Classification: A Preliminary Study," Proc. Work. Autom. Extr. Socio-political Events from News 2020, no. May, pp. 26–34, 2020.

[33] S. G. & A. K. Bhavya Mor, "A Systematic Review of Hidden Markov Models and Their Applications," Arch. Comput. Methods Eng., 2020.

[34] L. Lopes, R. Vieira, M. José, and D. Martins, "Extracting compound terms from domain corpora," no. February 2015, 2010, doi: 10.1007/s13173-010-0020-4.

[35] Y. Gao, C. M. Meyer, and I. Gurevych, Preference - based interactive multi - document summarisation, vol. 23, no. 6. Springer Netherlands, 2020.

[36] R. Rajendran, S. Iyer, and S. Murthy, "Personalized affective feedback to address students frustration in ITS," IEEE Trans. Learn. Technol., vol. XX, no. c, pp. 1–12, 2018, doi:

10.1109/TLT.2018.2807447.

[37] A. Ansani et al., “Symposium on Emotion Modelling and Detection in Social Media and Online Interaction,” no. April, 2018, [Online]. Available: http://aisb2018.csc.liv.ac.uk/PROCEEDINGS AISB2018/Emotion Modelling and Detection in Social Media and Online Interaction - AISB2018.pdf.

[38] A. Wahid and F. Rasheed, “Emotion Detection Using Unobtrusive methods: An Integrated Approach,” 2019 2nd Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2019, no. April 2020, pp. 1166–1171, 2019, doi: 10.1109/ICICICT46008.2019.8993118.

[39] F. Calefato, F. Lanubile, and N. Novielli, “EmoTxt : A Toolkit for Emotion Recognition from Text,” vol. 1, no. July, pp. 3–4, 2017.

[40] K. Sailunaz, M. Dhaliwal, J. Rokne, and R. Alhajj, “Emotion detection from text and speech: a survey,” Soc. Netw. Anal. Min., vol. 8, no. 1, pp. 1–8, 2018, doi: 10.1007/s13278-018-0505-2.

[41] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, “Text-based emotion detection: Advances, challenges, and opportunities,” Eng. Reports, vol. 2, no. 7, pp. 1–24, 2020, doi: 10.1002/eng2.12189.

[42] N. Kanger and G. Bathla, “Recognizing Emotion in Text using Neural Network and Fuzzy Logic,” vol. 10, no. March, pp. 10–15, 2017, doi: 10.17485/ijst/2017/v10i12/100526.

[43] V. V. Ramalingam, A. Pandian, A. Jaiswal, and N. Bhatia, “Emotion detection from text,” J. Phys. Conf. Ser., vol. 1000, no. 1, 2018, doi: 10.1088/1742-6596/1000/1/012027.

[44] P. Saigal and V. Khanna, “Multi-category news classification using Support Vector Machine based classifiers,” SN Appl. Sci., vol. 2, no. 3, pp. 1–12, 2020, doi: 10.1007/s42452-020-2266-6.

[45] S. N. Devi M.D., “Misogynous Text Classification Using SVM and LSTM.,” Garg D., Wong K., Sarangapani J., Gupta S.K. Adv. Comput. IACC 2020. Commun. Comput. Inf. Sci. vol . Springer, Singapore. https//doi.org/10.1007/978-981-16-0401-0_26, vol. 1367, 2021.

[46] J. Kolluri, S. Razia, and S. R. Nayak, “Text Classification Using Machine Learning and Deep Learning Models,” SSRN Electron. J., no. January, 2020, doi: 10.2139/ssrn.3618895.

[47] P. Di and L. Duan, “New naive Bayes text classification algorithm,” Shuju Caiji Yu Chuli/Journal Data Acquis. Process., vol. 29, no. 1, pp. 71–75, 2014, doi: 10.11591/telkomnika.v12i2.4180.

[48] X. Luo, “Efficient English text classification using selected Machine Learning Techniques,” Alexandria Eng. J., vol. 60, no. 3, pp. 3401–3409, 2021, doi: 10.1016/j.aej.2021.02.009.

[49] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, “A semantics aware random forest for text classification,” Int. Conf. Inf. Knowl. Manag. Proc., no. November, pp. 1061–1070, 2019, doi:

10.1145/3357384.3357891.

[50] M. Savargiv, B. Masoumi, and M. R. Keyvanpour, “A new random forest algorithm based on learning automata,” Comput. Intell. Neurosci., vol. 2021, 2021, doi: 10.1155/2021/5572781.

[51] K. Almohammadi and H. Hagras, “An adaptive fuzzy logic based system for improved knowledge delivery within intelligent E-learning platforms,” IEEE Int. Conf. Fuzzy Syst., pp. 1–8, 2013, doi: 10.1109/FUZZ-IEEE.2013.6622350.

[52] A. Aajli and K. Afdel, “Generation of an adaptive e-learning domain model based on a fuzzy logic approach,” Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA, pp. 1–8, 2017, doi: 10.1109/AICCSA.2016.7945708.

[53] K. Almohammadi and H. Hagras, “An Interval Type-2 Fuzzy Logic Based System for Customised Knowledge Delivery within Pervasive E- Learning Platforms,” pp. 2878–2885, 2013, doi: 10.1109/SMC.2013.490.

[54] K. Chrysafiadi and M. Virvou, “Fuzzy logic for adaptive instruction in an e-learning environment for computer programming,” IEEE Trans. Fuzzy Syst., vol. 23, no. 1, pp. 164–177, 2015, doi: 10.1109/TFUZZ.2014.2310242.

[55] Z. Liu, Z. Wang, and Z. Fang, “An agent-based e-learning assessing and instructing system,” Proc. - 2006 10th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2006, pp. 1414–1419, 2006, doi: 10.1109/CSCWD.2006.253194.

[56] A. A. Firte, C. V. Bratu, and C. Cenan, “Intelligent component for adaptive e-learning systems,” Proc. - 2009 IEEE 5th Int. Conf. Intell. Comput. Commun. Process. ICCP 2009, pp. 35–38, 2009, doi: 10.1109/ICCP.2009.5284788.

[57] M. Guan, J. Jia, Y. Yang, and Q. Chen, “Research on adaptive e-Learning system using technology of learning navigation,” Proc. 8th Int. Conf. Comput. Sci. Educ. ICCSE 2013, no. Iccse, pp. 24–29, 2013, doi: 10.1109/ICCSE.2013.6553877.

[58] M. Ueno and T. Okamoto, “Bayesian agent in e-learning,” Proc. - 7th IEEE Int. Conf. Adv. Learn. Technol. ICALT 2007, no. 1, pp. 282–284, 2007, doi: 10.1109/ICALT.2007.82.

[59] B. Deeb, Z. Hassan, and M. Beseiso, “An adaptive HMM based approach for improving e-Learning methods,” 2014 World Congr. Comput. Appl. Inf. Syst. WCCAIS 2014, 2014, doi: 10.1109/WCCAIS.2014.6916638.

[60] M. Rani, R. Vyas, and O. P. Vyas, “OPAESFH: Ontology-based personalized adaptive e-learning system using FPN and HMM,” IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2017-

Decem, pp. 2441–2446, 2017, doi: 10.1109/TENCON.2017.8228271.

[61] K. Colchester, H. Hagras, and D. Alghazzawi, “A Survey of Artificial Intelligence Techniques Employed for Adaptive Educational Systems within E-Learning Platforms,” J. Artif. Intell. Soft Comput. Res., vol. 7, no. 1, pp. 47–64, 2017, doi: 10.1515/jaiscr-2017-0004.

[62] H. Ahmed, I. Traore, and S. Saad, “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10618 LNCS, no. October, pp. 127–138, 2017, doi: 10.1007/978-3-319-69155-8_9.