Context-based sentiment analysis on customer reviews using machine learning linear models

Anandan Chinnalagu [Corresp., 1] , **Ashok Kumar Durairaj** [1]

[1] Computer Science, Government Arts College (Affiliated to Bharathidasan University, Tiruchirappalli), Kulithalai, Karur, Tamil Nadu, India

Corresponding Author: Anandan Chinnalagu
Email address: anandanc@hotmail.com

Customer satisfaction and their positive sentiments are some of the various goals for successful companies. However, analyzing customer reviews to predict accurate sentiments have been proven to be challenging and time-consuming due to high volumes of collected data from various sources. Several researchers approach this with algorithms, methods, and models. These include machine learning and deep learning (DL) methods, unigram and skip-gram based algorithms, as well as the Artificial Neural Network (ANN) and bag-of-word (BOW) regression model. Studies and research have revealed incoherence in polarity, model overfitting and performance issues, as well as high cost in data processing. This experiment was conducted to solve these revealing issues, by building a high performance yet cost-effective model for predicting accurate sentiments from large datasets containing customer reviews. This model uses the fastText library from Facebook's AI research (FAIR) Lab, as well as the traditional Linear Support Vector Machine (LSVM) to classify text and word embedding. Comparisons of this model were also done with the author's a custom multi-layer Sentiment Analysis (SA) Bi-directional Long Short-Term Memory (SA-BLSTM) model. The proposed fastText model, based on results, obtains a higher accuracy of 90.71 % as well as 20% in performance compared to LSVM and SA-BLSTM models.

1 # Context-based sentiment analysis on customer
2 # reviews using machine learning linear Models
3

5 Anandan Chinnalagu[1], Ashok Kumar Durairaj[2]
6
7 [1,2] Computer Science, Govt Arts College (Affiliated to Bharathidasan University,
8 Tiruchirappalli), Kulithalai, Karur, TN, India
9
10 Corresponding Author:
11 Anandan Chinnalagu[1]
12 3120 De La Cruz Blvd, Santa Clara, CA, 95054, USA
13 Email address: anandanc@hotmail.com

15 ## Abstract

16 Customer satisfaction and their positive sentiments are some of the various goals for successful
17 companies. However, analyzing customer reviews to predict accurate sentiments have been
18 proven to be challenging and time-consuming due to high volumes of collected data from various
19 sources. Several researchers approach this with algorithms, methods, and models. These include
20 machine learning and deep learning (DL) methods, unigram and skip-gram based algorithms, as
21 well as the Artificial Neural Network (ANN) and bag-of-word (BOW) regression model. Studies
22 and research have revealed incoherence in polarity, model overfitting and performance issues, as
23 well as high cost in data processing. This experiment was conducted to solve these revealing
24 issues, by building a high performance yet cost-effective model for predicting accurate
25 sentiments from large datasets containing customer reviews. This model uses the fastText library
26 from Facebook's AI research (FAIR) Lab, as well as the traditional Linear Support Vector
27 Machine (LSVM) to classify text and word embedding. Comparisons of this model were also
28 done with the author's a custom multi-layer Sentiment Analysis (SA) Bi-directional Long Short-
29 Term Memory (SA-BLSTM) model. The proposed fastText model, based on results, obtains a
30 higher accuracy of 90.71 % as well as 20% in performance compared to LSVM and SA-BLSTM
31 models.
32 **Subjects:** Machine Learning, Deep Learning, Customer Sentiment, Conversational Analytics.
33 **Key Words**: Sentiment Analysis, Natural Language Processing, Text Analytics

35 ## Introduction
36 Today, customer satisfaction plays a major role for a successful business providing products
37 and/or services. An analysis of consumer reviews is crucial to understand what a customer wants
38 in terms of sentiment, (*Duyu, Bing & Ting, 2015*) as well as the betterment of a company or
39 business to grow overtime. The phrase, "what other people think" has importance to a buyer's

decision when purchasing products and services according (*Bo & Lillian, 2008*) to survey. Most companies handle customer service via call-centers with live agents, and over the last few years, the availability of viewing opinions, reviews, testimonies, etc. (*Swapna et al, 2007*) on the web has provided new avenues of research for automatic subjectivity, understanding texts, and sentiment classification. Researchers would use ML and DL models with the use of Natural Language Processing (NLP) techniques to process and classify datasets filled with various reviews in their study. However, the purpose of NLP is to analyze, extract, and present information for better decision-making in businesses. The level of granularity in the process of analyzing controversial texts vary from individual characters to sub-word units or words forming (*Alexis et al., 2017*) a sentence to sentences forming paragraphs. Early research and applied methods in text analysis discriminates at a sentence and phrase level (*Hong & Vasileios, 2003*) between objective and subjective texts. Prime candidates for traditional solutions for a sentence level classification of a document include the bag-of-words approach, SVM's, or the Adaboost classifier. Accurate scores in sentiment, detection of negation and sarcasm, as well as challenges in word ambiguity and multi-polarization were taken into account when making considerations to machine learning algorithms while building a sentiment classifier. The evaluation of the models includes audio transcript, voice and text chats from various internal sources along with publicly available social media data sources. We use unigram, bigram, trigram and n-grams textual features in terms of multimodal sentiment analysis. Our model's training dataset includes customer sarcastic reviews in the context of customer sentiment. In this experiment polarity, negation and sarcasm are considered and classified as positive and negative sentiment.

Our main goal is to build a state-of-the-art learning model that utilizes proven binary and multi-class algorithms, methods, and word embedding techniques for classification. The following below are contributions made towards that goal:

- Solving issues with context-based sentiments through multi-layer SA model with Bi-LSTM, *fast*Text and LSVM. Data pre-processing is customized to fit the model requirement for customer review and speech transcript dataset.
- Solving vocabulary issues with predictions made from datasets containing mixed language texts by adding input and service layers to detect the baseline language, translates them into English, and form a transcript.
- Transcript and Translate service layers are added to model input layer for train and testing the domain-base mixed-language dataset to avoid out of vocabulary (OOV) issues.
- Saving cost by providing new data pipeline techniques while having increased performance to train models with large datasets.
- LSVM, *fast*Text and SA-BLSTM models hyperparameters are fine-tuned based on dataset.

This paper details several contributions from various researchers, such as relating literature, published works, and research papers and are taken into review. The methods used, as well as the pre-processing steps and flow of data are presented. This paper also describes the approach taken for the ML and DL models, as well as the architecture of various algorithms used for the model.

82 Details of the experiment are documented throughout, such as the setup and the results. In the
83 end, the results of the concluding model are shown, as well as a proposal for the future.
84

Related Works

86 Reviewed recent research papers and researchers' contributions towards text classification,
87 sentiment prediction. Our review focused on supervised ML, DL and SA research papers.
88   *Anna et.al., (2020)* published a paper related to COVID19 Cross-language SA of
89 European Twitter messages from Italy, Spain, France, and Germany. Neural network model used
90 with the pre-trained word or sentence embedding. Model constructed with fully-connected
91 Rectified Linear Unit (ReLu) layer to process output from embedded vector and a regression
92 output layer with sigmoid activation. In this experiment the following pre-trained word
93 embeddings models used, a skip-gram version of word2vec (Trained English-language
94 Wikipedia data), and a multilingual version of BERT (Trained on Wikipedia data and 160
95 million Covid-19 tweets key-words). The result showed, based on analysis over-all 4.6 million
96 tweets in which 79,000 tweets contains one keyword of Covid-19. In this paper researchers took
97 geolocation-based data and trends were varying and these researchers concluded, this study will
98 be continued to collect tweet data from other countries and compare the result and also moving
99 from the binary sentiment scale to a more complex model. *Swarn et al., (2021)* published an
100 article related to a machine learning scraping tool for a data fusion in the analysis of sentiments
101 about supporting business decisions with human-centric AI (HAI) explanations. The multinomial
102 Naïve Bayes (NB, k-nearest neighbours (KNN), SVMs and multinomial Bayesian classifiers are
103 used for sentiments analysis. This study results revealed KNN outperformed other models.
104   *Babacar, Dezheng and Aziguli (2021)* published an article related to employee sentiment
105 using employees' reviews. This study used traditional classifiers and vector stochastic gradient
106 descent classifier (RV-SGDC) for sentiment classification. RV-SGDC is a combination of
107 Logistic Regression (LR), Support Vector Machines (SVM), and Stochastic Gradient Descent
108 (SGD) model. The study result showed RV-SGDC outperforms with a 0.97% accuracy compare
109 to other models due to its hybrid architecture.
110   *Nalini et.al., (2021)* published an article paper related to SA of Covid-19 tweets using
111 Deep Learning (DL) models. These researchers' motto behind this study to analyze tweets by
112 Indian netizens during the lockdown, collected tweets between 23 March 2020 and 15 July 2020
113 and text has been labelled as fear, sad, anger and joy, analysed data using the new deep learning
114 model name called Bi-Directional Encoder Representation from Transformers (BERT). The
115 BERT model result was compared with traditional Logistic Regression (LR), Support Vector
116 Machines (SVM), Long Short-term Memory (LSTM). The BERT model result shows more
117 accuracy 89%, compare to other model's accuracy LR 75%, SVM 74.75% and LSTM 65%. This
118 experiment classified sentiment into fear, sad, anger and joy based on the key-words.
119   Most recently, *Najla et.al., (2021)* published a research article related to evaluation of SA
120 using Amazon Online Reviews dataset.[3] Researchers evaluated different deep learning
121 approaches to accurately predict the customer sentiment, categorized as positive, negative and
122 neutral. The variation of simple Recurrent Neural Network (RNN) such as Long Short-Term
123 Memory Networks (LRNN), Group Long Short-Term Memory Networks (GLRNN), Gated
124 Recurrent Unit (GRNN) and Updated Recurrent Unit (UGRNN). for Amazon Online Reviews.
125 All evaluated RNN Algothims were combined with word embedding as feature extraction
126 approach for SA including the following three methods Glove, Word2Vec and fastText by Skip-
127 grams. Evaluated combination of five RNN variants with three feature extraction methods,
128 evaluation result measured based on accuracy, recall, precision and F1 score. Found that the
129 GLRNN with fastText feature extraction scored the highest accuracy of 93.75%. Researchers try

to solve programming problem for beginners to code and find next word, used conventional LSTM model with word embedding, dropout layer with an attention mechanism.[1] This model result showed the pointer mixture model succeeded in predicting both the next within-vocabulary word and the referenceable identifier with higher accuracy than the conventional neural language model alone in both statically and dynamically typed languages.

Shreyas (2020) published a paper related to customer churn prediction, the traditional Logistic Regression (LR), Gaussian Naïve Bayes (GNB), Adaptive Boosting (AdaBoost), Extra Gradient Boosting (XGB), Stochastic Gradient Descent (SGD), Extra Trees and SVM classifiers are used for this experiment. Result showed Extra Trees classifier outperformed, SVM and XGB classifier performed well for Telecom (UCI repository) dataset. Many researchers (*Ikonomakis, Kotsiantis & Tampakas, 2005*) have shown combining multiple classifiers improve performance and classification accuracy of model in the context of combining multiple classifiers for text categorization.

Babacar, Dezheng and Aziguli (2021) published an article related to employee sentiment using employees' reviews using the traditional classifiers and vector stochastic gradient descent classifier (RV-SGDC) for sentiment classification. RV-SGDC is a combination of Logistic Regression (LR), Support Vector Machines (SVM), and Stochastic Gradient Descent (SGD). The result showed RV-SGDC outperforms with a 0.97% accuracy compare to other models due to its hybrid architecture.

Ashok & Anandan (2020) presented a research study paper related to Sentiment and Emotion in social media COVD19 Conversations. In this research, used variants of RNN algorithms and evaluated a multi-class neural network model using Bi-directional Long Short-term memory (Bi-LSTM) with additional layers to process the Covid-19 long text social media posting, overcome model outfitting, accuracy and performance problems. Authors experimental result showed SAB-LSTM model outperformed the traditional LSTM, Bi-LSTM models and sentiment prediction was context-based. Authors planned to extend their model for future research with domain-based dataset for customer SA problems, comparing with other models to improve the prediction accuracy and performance.

Kamran et.al., (2019) presented a text classification survey paper. In this paper, researchers discussed about existing classification algorithms, feature extraction, dimensionality reduction, and model evaluation methods, and also addressed critical limitation of each one of these components of the text classification pipeline. The following components are discussed, Algorithms such as Rocchio, bagging and boosting, logistic regression (LR), Naïve Bayes Classifier (NBC), k-nearest Neighbor (KNN), Support Vector Machine (SVM), decision tree classifier (DTC), random forest, conditional random field (CRF), and deep learning. Feature's extraction methods such as Term Frequency-Inverse document frequency (TF-IDF), term frequency (TF), and word-embedding methods such as Word2Vec, contextualized word representations, Global Vectors for Word Representation (GloVe), and fastText. Dimensionality reduction methods such as Principal component analysis (PCA), linear discriminant analysis (LDA), non-negative matrix factorization (NMF), random projection, Autoencoder, and t-distributed Stochastic Neighbor Embedding (t-SNE). Evaluation methods such as Accuracy, Fβ, Matthew correlation coefficient (MCC), receiver operating characteristics (ROC), and area under curve (AUC).

Joulin et.al., (2016) at Facebook's AI research lab (FAIR) released and presented a linear text classifier fastText library a paper. It proved that fastText library can be transformed into a simpler equivalent classifier, and also proved that the necessary, sufficient dimensionality of the word vector embedding space is exactly the number of document classes. Experiment results show that combination of bag of words and linear classification methods fastText accuracy is

178   same or slightly lower than deep learning algorithms, fastText performs well in normal
179   environment setup, even without using high performance GPU servers.
180   *Kamran et.al., (2017)* employed deep learning methods to multi-class documents
181   classifications. The traditional multi-class classification works well for a limited number classes
182   and the performance drops when increasing the number of classes and documents. To solve the
183   performance problems, experimented combination of deep learning, recurrent and convolutional
184   neural network models. This combined neutral networks, hierarchical DL classification model
185   (HiDLTex) result showed more accuracy than traditional SVM and Naïve bayes models.
186   *Lizhen et.al., (2010)* presented a paper at International Conference on computational
187   Linguistic related to Bag-of-Opinions method for review rating prediction from sparse text
188   patterns. Customers are writing their comments with implicitly expressing their opinion polarities
189   as positive, negative, and neutral, and also providing numeric ratings of products. The numerical
190   review rating prediction is harder than classifying by polarity. In this paper discussed about a
191   unigram-based regression model each unigram gets a weight indicating its polarity and strength
192   rating, for e.g. "This product is not very good" Vs "This product is not so bad", in this e.g.,
193   unigram regression model consider weight to "good" as positive and "bad" as negative, and it
194   assigns the strong negative weight to "not", combining this weight, it was not predicted the true
195   intention of opinion phrases. These models are not robust and referred unigram regression model
196   as polarity incoherence. To overcome these two models, introduced a novel kind of Bag-of-
197   opinion (BoO) with approach of cumulative linear offset (CLO) model representation, where an
198   opinion, within a review consists of the following three components, a root word, a set of
199   modifier words from the same sentence, and one or more negation words. For a phrase e.g., "not
200   very helpful" has opinion root word "helpful", modifier word "very" and a negation word "not".
201   Enforced polarity coherence by the design of a learnable function that assigns a score to an
202   opinion by ridge regression, from a large, domain-independent corpus of reviews. All Amazon
203   reviews dataset used for BoO model training and testing regardless of domains.
204   *Xiang and Yann (2016)* published a paper to determining the explicit or implicit meaning
205   of words, phrases, sentences and paragraphs, and making inferences about these properties such
206   as words and sentence of these texts has been traditionally difficult because of the extreme
207   variability in language formation. The text understanding is another area of research to
208   understand the text formed in natural languages such as English, Chinese, Spanish and others. To
209   solve text understanding problem convolutional networks (ConvNet) models were used for
210   research studies. For English text understanding, model was built using these 70 characters,
211   including 26 English letters, 10 digits, new line and 33 other characters.
212

Methodologies and Process Flow

214   Recently, researchers are used deep learning and neural network models for SA problems,
215   however neural network approach cost more compare to traditional baseline methods for both
216   supervised and unsupervised learning. The combination of right methodologies and text
217   classification algorithms are contributed to overcome SA models accuracy and performance
218   problems. The following process flow diagrams Fig.1. shows the steps followed for this
219   experiment. In general, the data pre-processing step, data obtain from publicly available
220   customers review data is very often incomplete, inconsistent and filled with a lot of noise and It's
221   likely contained errors not suitable for training and testing the machine learning models.
222   The following minimal syntactical data preprocessing steps of lowercasing all words, removing
223   new lines, punctuation, special characters and stripping recurring headers are needed for Neural

224   networks and embedding models. To improve data quality, introduced   additional steps which
225   includes stop words removal, text standardization, spelling correction, correcting the negation
226   words, tokenization, stemming, and Exploratory Data Analysis (EDA). Fig.2. shows the three
227   proposed models input and output data flow. These models are customized to fit the dataset. In
228   this experiment, added additional layers SA-BLSTM to handle large volume of customer reviews
229   and speech transcript data. *fast*Text and SVM are linear models. All these pre-trained models and
230   developed algorithms can be used for production purpose on real-time SA business applications.
231

Proposed Models

233   LSVM, *fast*Text and SAB-LSTM models are used for this experiment, before building the
234   models, reviewed algorithms towards solving the short and long text classification and SA
235   problems. In these following sections, explained all these three custom model architectures.
236

Support Vector Machine (SVM)

238   SVM is used for text classification problems, this algorithm is viewed as a kernel machine, and
239   the kernel functions can be changes based on the problem. Definition of Support Vector
240   Machines (SVM): It performs classification by finding the hyperplane that maximize the margin
241   between the two classes. The support vector is the vectors that define the hyperplane. For
242   instance, given pictures of apples and oranges, state whether the object in question is an apple or
243   an orange. Equally well, predict whether a customer is satisfied or not satisfied given customers
244   positive and negative sentiment data. SVM performs classification by finding the hyperplane that
245   maximize the margin between the two classes. In other words, SVM is that the partition which
246   segregate the classes. Fig.3. shows an example and definition of a point and vector, plotted a
247   point $A(4,2)$ and any point,

248    $$x = (x_1, x_2), x \neq 0 \qquad\qquad (1)$$

249   A vector is an object that has both a magnitude and a direction. In geometrical term, a
250   hyperplane is a subspace whose dimension is one less than that of its ambient space. If a space is
251   in 3-dimensional then it's a plane, if a space is in 2 dimensions, then it's a line, if the space in
252   one dimension, then it's a point. The following Fig.4. shows the hyperplane definition.
253   The linear and non-linear classifier (*Kamran, 2019*) data separation shown in the Figure.5.
254   for the 2-dimensional dataset. If the dataset is separable then linear kernel works well for
255   classification. Because of the following reasons, the linear kernel is used for the text
256   classification. The most text categorization problems are linearly separable and the linear kernel
257   works well with lot of features and also less parameters to (*Thorsten, 1998*) optimize for training
258   the model.
259   The following Fig.6. shows the linear kernel (*Sven et al., 2015*) SVM model. There many
260   kernel functions have been developed over the years; a kernel is a function, that returns the result
261   of a dot product performed (*Alexandre Kowalczyk, 2017*) in another space.
262   The linear kernel is simplest kernel function $k(x,y)$, it is given by the inner product $<x,y>$,
263   and an optional constant c.

264    $$k(x,y) = x^T + c \qquad\qquad (2)$$

265   Linear kernel is used in this experiment for text classification. The Polynomial kernel, RBF
266   kernel and String kernel functions can be used for other classifications problems.
267
268

fastText

270 Facebook AI research (FAIR) lab release an open-source free library called fastText for text
271 representation and classification. It's a lightweight method and work on standard generic
272 hardware with multicore CPU. *fast*Text approach evaluated for tag prediction and sentiment
273 analysis by FAIR. *fast*Text experiments (*Joulin et al., 2016*) show that it is often on par with
274 recently proposed DL methods in terms of accuracy, performance, faster for training and
275 evaluation. Facebook allows research community to build the models on top of the *fast*Text
276 open-source code. *fastText* introduce a new word embedding approach an extension of the
277 continuous skipgram and Continuous Bag of Words (CBOW) model like word2vec, where each
278 word is represented as a bag of character *n*-grams. The original version of *fast*Text is trained on
279 (*Matthias & Stephan 2019*) Wikipedia and its available for 294 languages. The main difference
280 between word2vec and *fast*Text is that *fast*Text sees words as the sum of their character n-grams
281 and it treats a vector representation is associated to each character *n*-gram and words being
282 represented (*Piotr et al., 2017*) as the sum of these representations. This new approach has clear
283 advantages, as it can calculate embeddings even for out-of-vocabulary (OOV) words. However,
284 word2vec embedding approach treat word as the minimal entity and try to learn their respective
285 embedding vector, in case if the word does not appear in the training corpus, then it fails to get
286 word vector representation. Fig.7. shows the CBOW and Skip-gram model architecture, The
287 CBOW is the distributed representation of context model, it predict the words in middle of a
288 sentence based on surrounding words. However, the Skip-gram predicts context within a
289 sentence (*Tomas, Quoc & Ilya, 2012*).
290

291 The Skip-gram maximize the average log probability for input of training words w,

292 $$w_1, w_2, w_3 ....., w_T \tag{3}$$

293 In the following Eq. (4) used for computing probability.

294 $$\frac{1}{T}\Sigma_{t=1}^{T}\left[\sum_{j=-k}^{T} \log p(w_{t+j}|w_t)\right] \tag{4}$$

295 Here k represents the size of the training window and function for w_t word in the middle.

296 -k to k is the representation inner summation and it computes the log probability of the word
297 w_{t+j} prediction for word in the middle w_t. The outer summation words are based on the corpus
298 used for model training.
299   The skip-gram model consists of input u_w, and output v_w, vectors associated with each word w.
300 The following probability Eq. (5) is used to predict the word u_i, from w_j.

301 $$p(w_i|w_j) = \frac{\exp(u_{w_i}\top v_{w_j})1}{\Sigma_{l=1}^{V}\exp(u_{w_l}\top v_{w_j})} \tag{5}$$

302 Her, the total number V of words in the given vocabulary.

303   CBOW and skip-gram models are obtaining the semantic information during the large
304   datasets used for training. The words which are closely related has the similar vector
305   representation the words. For example, school, college, university, education words are having
306   similar context, similarly orange and apple are having similar context representations.
307

308   The Fig.8. shows the architecture (*Joulin et al., 2016*) of a simple linear model of *fast*Text
309   with N gram features.

$$x_{1,} x_2,.....x_N \tag{6}$$

311   The features are embedded and averaged to form the hidden variable. This model is a simple
312   neural network with only one layer (*Maria, 2018*). The bag-of-words representation of the text is
313   first fed into a lookup layer, where the embedding is fetched for every single word. Then, those
314   word embedding are averaged, so as to obtain a single averaged embedding for the whole text.
315   At the hidden layer we end up with n words x dim number of parameters, where dimension is
316   the size of the embedding and n words is the vocabulary size. After the averaging, we only have
317   a single vector which is then fed to a linear classifier: we apply the softmax over a linear
318   transformation of the output of the input layer. The linear transformation is a matrix with
319   dimension1 $x_N$output, where N output is the number output classes. The following Eq. (7) is the
320   negative log likelihood function of fastText model.

$$-\frac{1}{N} = \sum_{n=1}^{N} y_n \log \left(f\left(BAx_n \right) \right) \tag{7}$$

322   Here, x_n represents the n-gram feature of the word,

323   A represents the lookup matrix of the word embedding,

324   B represents the linear output of the model transformation,

325   f represents the softmax function.

326

327   The softmax function calculates the probabilities distribution of the event over n different
328   events. The softmax takes a class of values and converts them to probabilities with sum 1. So, it
329   is effectively squashing a k-dimensional vector of arbitrary real values to k-dimensional vector
330   of real values within the range 0 to 1. The following Eq. (8) is the softmax function f of fastText.

$$\text{softmax}(z) = \frac{\exp(z)}{\sum_{k=1}^{K} exp(z)} \tag{8}$$

332

333   The fastText has the following tuning parameters: *Epoch:* By default, the model is trained on
334   each example for 5 epochs, to increase this parameter for better training specify the number of
335   epoch argument. *Learning rate (lr):* The learning rate controls how "fast" the model updates
336   during training. This parameter controls the size of the update that is applied to the parameters of
337   the models (*Maria, 2018*). Changing learning rate implies changing the learning speed of our
338   model is to increase (or decrease) the learning rate of the algorithm. This corresponds to how
339   much the model changes after processing each example. A learning rate of 0 would means that
340   the model does not change at all, and thus, does not learn anything. Note that this calculation of

341   the best model is going to be quite expensive. There is no magic formula to find the
342   hyperparameters for the best model. Just taking one hyperparameter, the learning rate, would
343   make the calculation impractical. This is a continuous variable and it would need to feed in each
344   specific value, compute the model, and check the performance. *Loss function:* In this we are
345   using softmax as loss function. The most popular methods for learning parameters of a model are
346   using gradient descent. Gradient descent is basically an optimization algorithm that is meant for
347   minimizing a function, based on which way the negative gradient points toward. In machine
348   learning, the input function that gradient descent acts on is a loss function that is decided for the
349   model. The idea is that if we move towards minimizing the loss function, the actual model will
350   "learn" the ideal parameters and will ideally generalize to out-of-sample or new data to a large
351   extent as well. In practice, it has been seen this is generally the case and stochastic gradient,
352   which is a variant of gradient descent, has a fast-training time as well. Since it needs to obtain the
353   posterior distribution of words, the problem statement is more of a multinomial distribution
354   instead of a binary.
355   The following Fig.9. shows the text documents process flow using fastText linear model.
356   In this model text classification pipe-line, raw text documents are processed using data pre-
357   processing steps described in Fig.1. and processed text data tested using fastText model, the
358   model output classified the output into two classes (Satisfied and Not-Satisfied). *fast*Text is a
359   classification algorithm and C++ used to compile fastText mode. It provides high accuracy as
360   well as good performance (*Vladimir & David, 2017*) during training and testing the model.
361
362   **SA-BLSTM Model**

363   Authors SA-BLSTM is a sequence processing model (*Ashok & Anandan, 2020*). The
364   Bidirectional LSTM is used in this model. The extended LSTM architecture is shown in Fig.10.
365   with Input, Output, multiplicative Forget gates and all the gates are using (*Tomas et al., 2013*)
366   activation function *f* sigmoid.
367   The Constant Error Carousels (CEC) is the central feature of LSTM (*Ashok & Anandan,*
368   *2020*) and it solves the vanishing problem. CECs back flow is constant when there are no input
369   or error signals to the cell. Input and output gates protect CECs error flow from forward and
370   backward activation. If the gates are closed or the gates activation is around 0 then the irrelevant
371   input will not enter the cell. The Forward pass LSTM computes the output of the network given
372   the input data, the Backward pass LSTM computes the output error with respect to the expected
373   output and then go backward into the network and update the weights using gradient descent. To
374   compute (*Tomas et al., 2013*) the network weight for a single input to output network, the back
375   propagation (BP) uses the loss function to compute the gradient. The following are the equations
376   for gates and activation functions. LSTM Gates are the activation of sigmoid function, between 0
377   and 1 is output value of the sigmoid. When the gates are blocked the value is 0 and when the
378   value is 1 then gates allow the input to pass through.

379   $$\text{sig}(t) = \frac{1}{1 + e^{-t}} \tag{9}$$

380   Here are the equations for all three gates.

381   Input Gate $$i_t = \sigma(\omega_i[h_{t-1}, x_t] + b_i) \tag{10}$$

382   Output Gate $\qquad\qquad o_t = \sigma(\omega_f[h_{t-1}, x_t] + b_o)$ (11)

383   Forget Gate $\qquad\qquad f_t = \sigma(\omega_o[h_{t-1}, x_t] + b_f)$ (12)

384   σ Represents sigmoid function, x_t Represents input at current timestamp. h_{t-1} Represents
385   LSTM block output of previous state at timestamp t-1. ω_i, ω_f, and ω_o are represents weight
386   for the input, forget and output gates. b_i, b_o and b_f are represents bias for input, output, forget
387   gates. The following are equations for cell state for gates (*Ashok & Anandan, 2020*).

388   Cell State of Input gate $\tilde{c}_t = tanh\,(c\omega[h_{t-1}, x_t] + b_c)$ (13)

389   Cell Sate of Output gate $c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ (14)

390   Cell Sate of Forget gate $\quad h_t = o_t * tanh\,(c^t)$ (15)

391    $\tilde{c}_t$ Represents input gate cell state at timestamp (*t*). c_t Represents memory cell state at timestamp
392   (*t*). h_t Represents cell state of final output at timestamp (*t*).

393   This model (*Ashok & Anandan, 2020*) consists of Input, Language detection and translation,
394   embedded, Bi-Directional LSTM neutral network layer, dropout, dense and output layers. The
395   input layers process the multilingual mixed customer reviews and speech transcript dataset and
396   vectorizing the data using word embedding technique, each post has one or more sentences, and
397   each sentence is composed with n number of words sequence. Here x representing input during
398   the language detection, language translation process.

399                  $x = x_1, x_2, x_3, ...x_T$ (16)

400   In the detection layer, input text processed to detect the non-English text, here d represents
401   input to detection layer.

402                  $d = d_1, d_2, d_3, ...d_T$ (17)

403   If the input text identified as non-English text, then the language translation layer converts the
404   text to English, here t represents the input of translation layer.

405                  $t = t_1, t_2, t_3, ...t_T$ (18)

406   The output of the translation layer processed by embedding layer, each input word converted
407   to vector, here S represents vector value.

408                  $S = w_1, w_2, w_3, ...w_n$ (19)

409   The Bi-Directional LSTM (BLSTM) is a sequence processing model, it consists of two
410   LSTM units, (*Karthik & Fathi, 2020*) one unit taking the input in a forward direction and other
411   unit taking the input in a backward direction. It effectively processes the input and context
412   available to the network. Fig.11. shows the mixed-language data processing flow, the language
413   detection and translation layers convert the non-English to English and then it's embedding the
414   words.

415   Input layer fed the embedded dataset to Bi-LSTM model and it processes vector output of the
416   embedded layer. The following Fig.12. shows the SA-BLSTM model architecture. This model
417   used for both binary and multiclass classifications and SA applications.
418
419 **Experiment and Results**

420   For models training and testing, used Windows 64-bit Operating System with Intel core i7
421   processor, 16 GB Memory, and on-board GPU NVIDIA MX150 server environment. Developed
422   models using Python, Jupyter, Anaconda IDE and used Python libraries, Pandas for data load
423   processes, Numpy for mathematical operations, Seaborn and Matplotlib for plotting graph,
424   NLTK Tool kit, Wordcloud and Sklearn used for build, train and test the models. This novel
425   method uses a memory caching technique with automated custom Python scripts to speed-up
426   data preprocessing tasks during model training on large volume dataset.
427
428
429   To compare various traditional models results for Twitter dataset, collected related (*Abhilash &*
430   *Sanjay, 2019*) experimental results. The following Table 1. shows the various models accuracy.
431
432   The following Fig.13. shows experiment result of fastText model with bigram.
433
434   Dataset collected from publicly available Twitter, IMDB movie review, Amazon product
435   review and Yelp sentiment analysis data source from kaggle.com, Total 778631 dataset, 70%
436   (545041) of data used for Training and 30% (233590) of data used to test the models which
437   includes Kaggle.com sentiment datasets, chat conversations from chat application and transcript
438   of sample audio files. These data sources details and URLs are listed in Table 2.
439
440   Designed and developed fastText and Linear SVM (LSVM) models, used the linear kernel
441   setting for LSVM model and for fastText model used the epoch=10, lr=0.01 and loss=softmax
442   parameters. Both the models trained with unigram(n=1), bigram(n=2) and trigram(n=3) and n-
443   grams parameters. The unigram (n=1) model result shows the polarity incoherence (*Lizhen,*
444   *Georgiana & Gerhard, 2010*) due to unigram model gets a weight indicating its polarity and
445   strength, for e.g. *not so good Vs. not so bad*, the fundamental problem arises in unigram model
446   when assign the weight to *not*. Analyzed the training result, 3-gram showed the better
447   performance of LSVM and fastText model for this dataset. Tested with authors Pre-trained SBA-
448   LSTM model using the same dataset. We also combined the datasets of Amazon, Yelp, Twitter,
449   IMDB, Chat and audio transcripts for the LSVM and SA-BLSTM models.
450   During the model testing, captured the test results and tested n-gram features on both the
451   models. The result shows for example, customer wrote the following review about phone
452   purchase experience. e.g., "*Even after three working days of phone purchase. Noticed that phone*
453   *Service was not good.*"

454   Based on the unigram method the above text considers only one word for instance, in this
455   case the above sentence "*phone Service was not good*", it can be written a

456   Probability(P) P("*phone Service was not good*") = P("*phone*") * P("*Service*") * P("*was*
457   ") * P("*not*") * P("*good*")   (20)

458   From this equation here unigram n=1 matches pattern word by word in this case *"good"*
459   gets more weight considering it is an individual word so the SVM unigram predicts 32% as
460   positive and fastText unigram predicts 19.37% as positive. So, to avoid this problem, used
461   Bigram n=2, where the algorithm considers the "not good" as a single word while learning the
462   pattern, the probability of whole sentence can be written as follows;

463
464   Probability(P) P("*phone Service was not good*") = P("*Service*
465   "|start of sentence) * P("*Service | phone*") * P("*was | Service*") * P("*not|is*") * P("*good |*
466   *not*")   (21)

467
468   As per maximum likelihood estimation, the condition probability of something like P("*good*
469   *| not*") can be given as the ratio of count of the observed occurrence of "not good" together by
470   the count of the observed occurrence of "not". These models can predict new sentences.
471   Similarly, trigram n=3 to learn the probability of occurrence of pattern so that model becomes
472   more accurate. The trigram n=3 results showed more accurate compare to other unigram and
473   bigram testing parameters for both the models LSVM and fastText. The following metrics are
474   used to evaluate the models training performance matrix based on True Positive (TP), True
475   Negative (TN), False Positive (FP) and False Negative (TN).

476    $$Precision = \frac{tp}{tp + fp}$$ (22)

477    $$Recall = \frac{tp}{tp + fn}$$ (23)

478    $$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$ (24)

479    $$Accuracy\ \% = \frac{tp + tn}{tp + fp + tn + fn} * 100$$ (25)

480   The following Table 3 shows the model training performance for 3-gram method for both
481   fastText and LSVM. This result shows both linear models were performed well during the model
482   training, fastText performed slightly better than LSVM.
483   The following Table 4 shows n-grams performance measures of model's accuracy, Recall,
484   Precision and F1 Score.
485
486   The LSVM and fastText are showing similar model accuracy results. SAB-LSTM shows less
487   accuracy. The following Table 5. shows the % conversations express the positive and negative
488   score of customer sentiment.
489

490

491

Conclusions

492

493 The results from training the model revealed that *fast*Text performed exceedingly compared to
494 the LSVM and SA-BLSTM models, and that fastText is much more suitable with large datasets
495 within a server that has minimal configuration. The results revealed that fastText provided a
496 more accurate response within a small duration of time compared with the other two models,
497 obtaining a 90.71 % rate in comparison to LSVM and SA-BLSTM models. The authors
498 concluded that the n-gram method had better compatibility with both fastText and LSVM for the
499 type of dataset used for the experiment, and noticed that training domain-specific datasets
500 improves the accuracy of the sentiment score when tested with a particular domain. fastText
501 shows much better performance during model training compare to LSVM and SA-BLSTM
502 models. The fastText works well with large dataset within a minimal configuration of server
503 infrastructure setting. Experiment result shows fastText model training time duration is less and
504 it gives more accuracy and response time is faster than LSVM and SA-BLSTM models. The n-
505 gram method works better for both fastText and LSVM, especially trigram n=3 for this dataset.
506 Noticed that a domain specific dataset training improves the accuracy of the sentiment score
507 when it tested with particular domain. There is research that is highly essential in the future to
508 explore a framework to build generic models that would be beneficial for industries such as
509 healthcare, retail, and insurance. The SA-BLSTM model the authors have built has the ability to
510 integrate fastText for representation of words to provide increased performance, and has the
511 ability to be pre-trained for such industries that could benefit from this. However, improvements
512 should be made for the quality of audio text files, as well as the use of automation scripts to
513 correct text errors in conversations.

514

515

516

References

517

518 **Abhilash Mittal, Sanjay Patidar, 2019.** Sentiment Analysis on Twitter Data: A Survey, Delhi
519   Technological University, New Delhi, India. ACM. © 2019 Association for Computing
520   Machinery. ACM ISBN 978-1-4503-7195-7/19/07.
521 **Alexis Conneau, Holger Schwenk, Loic Barrault, and Yann Lecun, 2017.** Very Deep
522   Convolutional Networks for Text Classification, arXiv:1606.01781v2[cs.CL].
523 **Alexandre Kowalczyk, 2017.** Support Vector Machine Succinctly, Syncfusion Inc,
524   https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/
525   (Accessed 5 October 2021).
526 **Anna Kruspe, Matthias Haberle, Iona Kuhn, Xiao Xiang Zhu, 2020.** Cross-language
527   sentiment analysis of European Twitter messages during the COVID-19 pandemic,

528   German Aerospace Center (DLR) Institute of Data Science, Technical University of
529   Munich (TUM). arXiv:2008.12172v1 [cs.SI].

530 **Ashok Kumar D, Anandan Chinnalagu**, **2020.** Sentiment and Emotion in social media
531   COVD19 Conversations: SAB-LSTM Approach in *proceedings of the SMART–2020,*
532   *IEEE Conference ID: 50582 9th International Conference.*
533   *DOI:10.1109/SMART50582.2020.9337098*

534 **Babacar Gaye, Dezheng Zhang and Aziguli Wulamu, 2021**. Sentiment classification for
535   employees reviews using regression vector-stochastic gradient descent classifier (RV-
536   SGDC). *PeerJ Comput. Sci. 7:e712 http://doi.org/10.7717/peerj-cs.712.*

537 **Bo Pang, and Lillian Lee, 2008.** Opinion mining and sentiment analysis, [Foundation and
538   Trends in Information Retrieval vol 2].

539 **Duyu Tang, Bing Qin and Ting Liu, 2015.** Document modeling with gated recurrent neural
540   network for sentiment classification, Harbin Institute of Technology, Harbin, China.
541   [*2015 Conference on Empirical methods in Natural Language processing*], DOI:
542   10.18653/v1/D15-1167.

543 **Hong Yu and Vasileios Hatzivassiloglou, 2003.** Towards Answering Opinion Questions:
544   Separating facts from opinions and identifying the polarity of opinion sentences,
545   *Conference on Empirical Methods in NLP*.

546 **Ikonomakis M, S. Kotsiantis S and V. Tampakas V, 2005.** Text classification using machine
547   learning techniques, WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4.

548 **Joulin A, Grave E, Bojanowski P, and Mikolov T, 2016.** Bag of tricks for efficient text
549   classification, Facebook AI Research arXiv:1607.01759v3 [cs.CL].

550 **Karthik Gopalakrishnan and Fathi M. Salem, 2020**. Sentiment Analysis Using Simplified
551   Long Short-term Memory Recurrent Neural Networks, Department of Electrical and
552   Computer Engineering, Michigan State University", Michigan, USA.
553   arXiv:2005.03993v1[cs.CL].

554 **Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura**
555   **Barnes, and Donald Brown, 2019,** Text Classification Algorithms: A Survey, DOI:
556   10.3390/info10040150.

557 **Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi,**
558   **Matthew S. Gerber and Laura E. Barnes, 2017.** HDLTex: Hierarchical Deep Learning
559   for Text Classification, University of Virginia, Charlottesville, VA, USA.

560 **Kenta Terada and Yutaka Watanobe**, **2021,** Code Completion for Programming Education
561   based on Deep Learning (2021), *Intl. J. Computational Intelligence Studies* Vol.10, Nos
562   2/3.

563 **Lizhen Qu, Georgiana Ifrim and Gerhard Weikum, 2010.** The Bag-of-Opinions method for
564   review rating prediction from sparse text patterns, *23rd International Conference on*
565   *computational Linguistic*, Beijing, China.

566 **Maria Mestre, 2018,** FastText: stepping through the code,
567   https://medium.com/@mariamestre/fasttext-stepping-through-the-code-259996d6ebc4,
568   (Accessed on 12 November 2021)
569 **Matthias Nitsche, and Stephan Halbritter, 2019.** Comparison of Neural Document
570   Classification Models, Hamburg University of Applied Sciences, Department of
571   Computer Science, Hamburg, Germany.
572 **Najla M. Alharbi, Norah S. Alghamdi, Eman, H. Alkhammash and Jehad F. Ali Amri**,
573   **2021**, Evaluation of Sentiment Analysis Via Word Embedding and RNN Variants for
574   Amazon Online Reviews, Hindawai, Mathematical Problems in Engineering. Volume
575   2021. Article ID 5536560.
576 **Nalini Chinatalapudi, Gopi Battineni, and Francesco Amenta, 2021.** "Sentimental Analysis
577   of COVID-19 Tweets Using Deep Learning Models", Copyright @2021 by the authors.
578   MDPI, Basel, Switzerland.
579 **Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov, 2017.** "Enriching
580   Word Vector with Subword Information". Facebook AI Research. arXiv:1607.04606v2
581   [cs.CL].
582 **Swapna Somasundaran, Theresa Wilson, Janyce Wiebe, and Veselin Stoyanov, 2007.** QA
583   with Attitude: Exploiting Opinion Type Analysis for Improving Question Answering in
584   On-line Discussions and the News, ICWSM, Boulder, Colorado, USA. Year 2007.
585 **Sven F. Crone, Stefan Lessmann, Rober Stahlbock, 2015.** Artificial Neural Network – New
586   Potential in Data Mining for Customer Relationship Management? Lancaster University,
587   Humboldt-University, University of Hamburg.
588 **Shreyas Rajesh Labhsetwar, 2020.** Predictive Analysis of Customer Churn in Telecom
589   Industry Using Supervised Learning, Department of Computer Engineering. Fr. C.
590   Rodrigues Institute of Technology, India. *ICTACT Journal on Soft Computing*, January
591   2020, Vol:10, Issue:3.
592 **Swarn Avinash Kumar, Moustafa M. Nasralla, Ivab Garcia-Magarino, and Harsh Kumar,**
593   **2021.** A machine-learning scraping tool for data fusion in the analysis of sentiments
594   about pandemics for supporting business decisions with human-centric AI explanations.
595   *PeerJ Comput. Sci. 7: e713 DOI 10.7717/peerj-cs.713.*
596 **Thorsten Joachims, 1998.** Text categorization with Support Vector Machines: Learning with
597   many relevant features, *International Conference on Machine Learning ICML.*
598 **Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, 2013.** Efficient Estimation of
599   Word Representations in Vector Space, Google Inc.
600 **Tomas Mikolov, Quoc V. Le and Ilya Sutskever, 2012.** "Exploiting Similarities aiming
601   Languages for Machine Translation", Google Inc.
602 **Vo Thi Ngoc Chau and Nguyen Hua Phung, 2020.** Enhanced CNN Models for Binary and
603   Multiclass Student Classification on Temporal Educational Data at the Program Level.
604   Vietnam Journal of Computer Science. Vol 8. No 2 (2021) 311-335.

605 **Vladimir Zolotov and David Kung, 2017.** Analysis and optimization of fastText linear text
606   classifier, IBM T.J.Watson Research Center. Yorktown Heights, NY,
607   USA.arXiv:1702.05531v1 [cs.CL].
608 **Xiang Zhang and Yann LeCun, 2016.** Text understanding from Scratch, New York University,
609   arXiv:1502.01710v5 cs.LG.

Figure 1

Methodology and Process Flow Diagram.
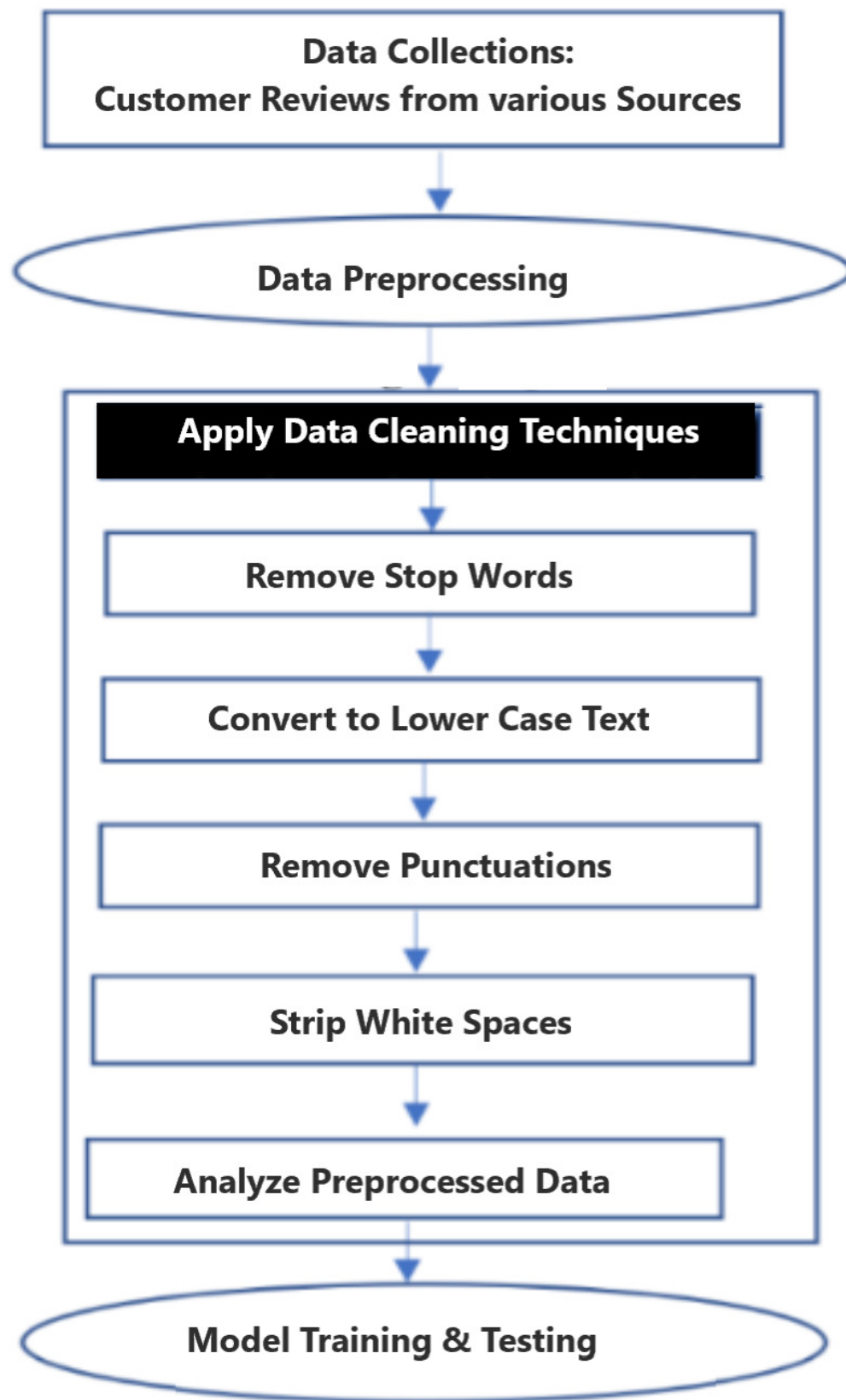
Figure 2

Input and model output data flow.
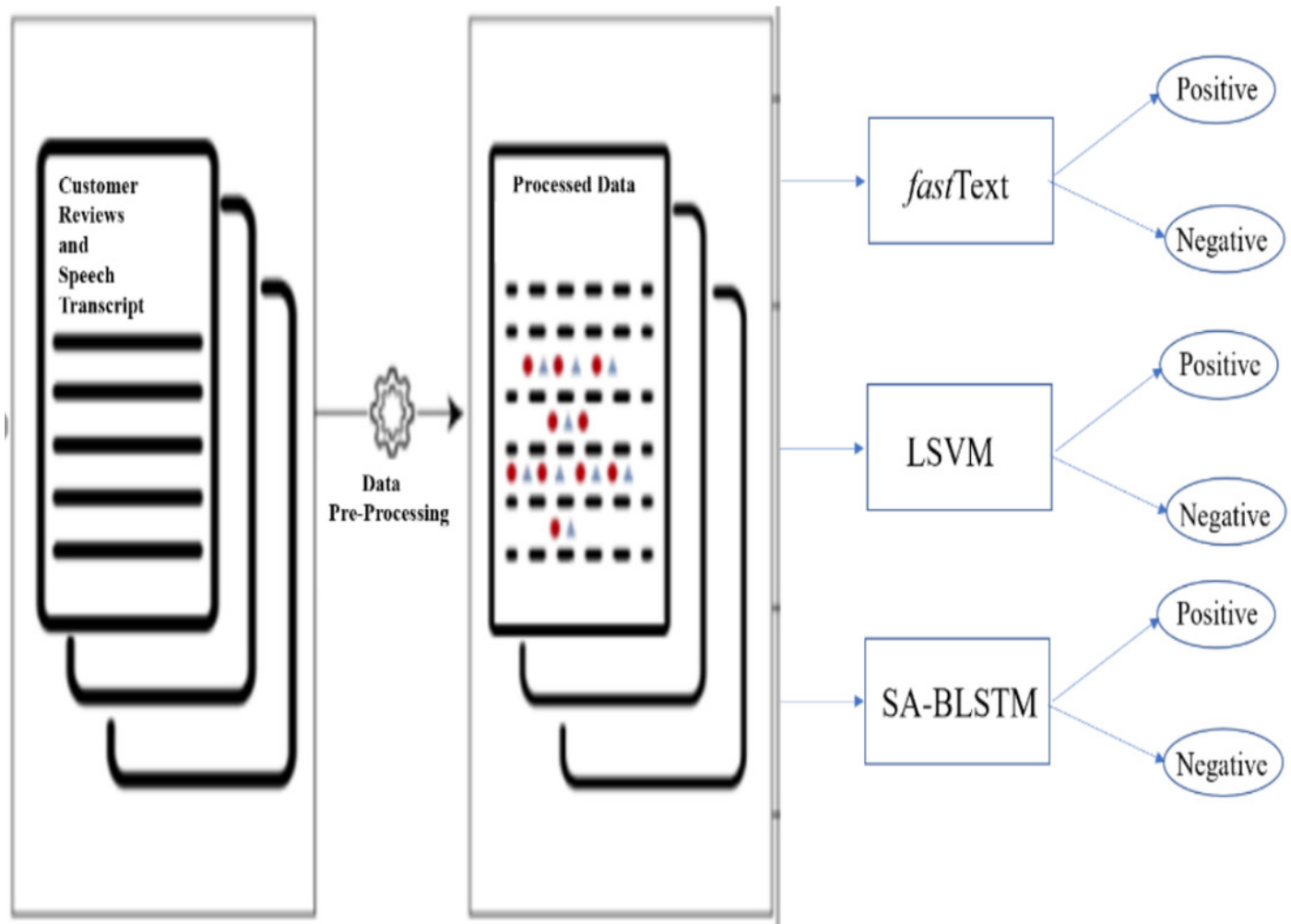
Figure 3

Figure. 3.Representation of Vector
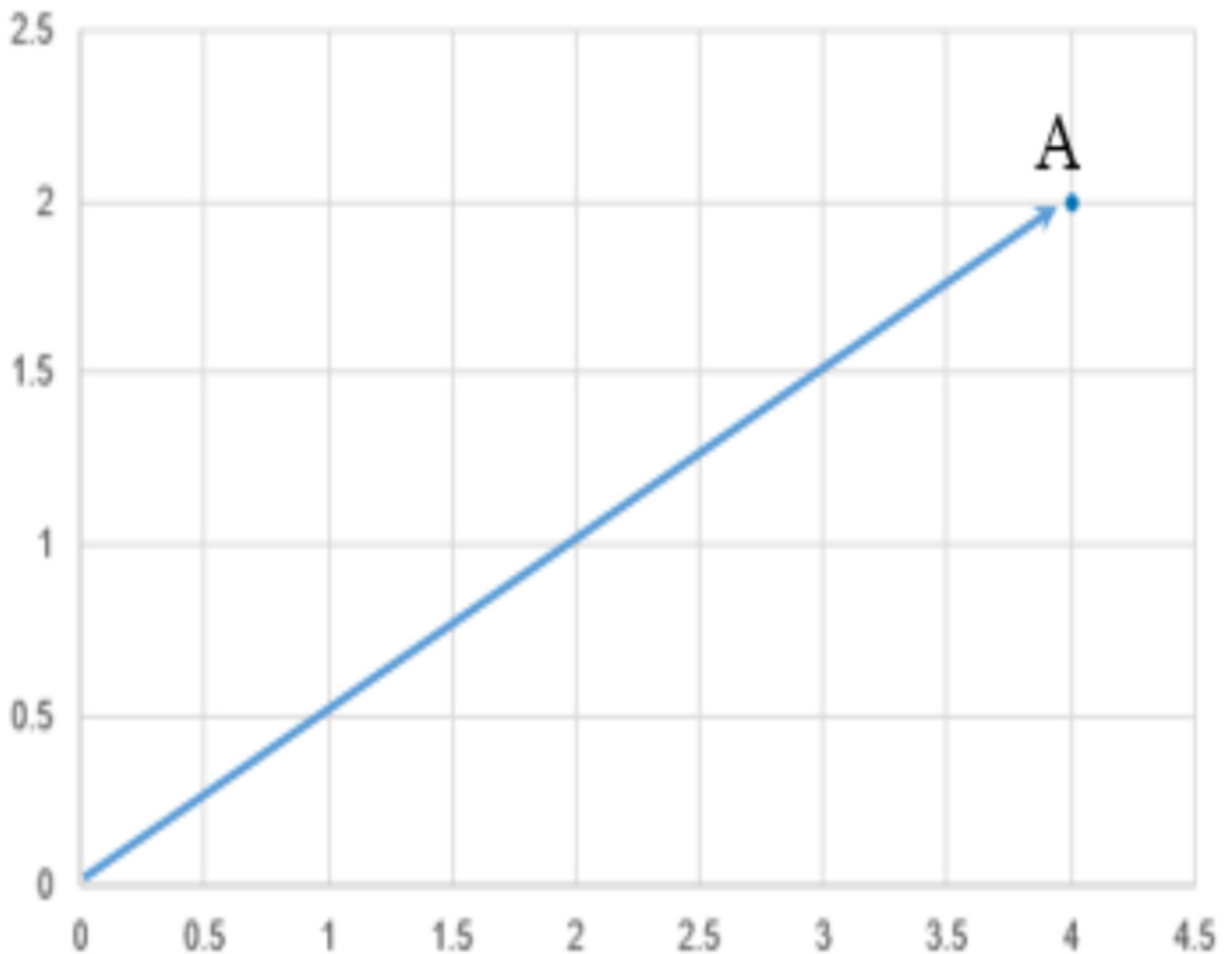
Figure 4

Hyperplane.

Define the hyperplanes H such that:
$w \bullet x_i + b \geq +1$ when $y_i = +1$
$w \bullet x_i + b \leq -1$ when $y_i = -1$

H_1 and H_2 are the planes:
$H_1: w \bullet x_i + b = +1$
$H_2: w \bullet x_i + b = -1$
The points on the planes H_1 and H_2 are the tips of the Support Vectors
The plane H_0 is the median in between, where $w \bullet x_i + b = 0$

     d+ = the shortest distance to the closest positive point
     d- = the shortest distance to the closest negative point
     The margin (gutter) of a separating hyperplane is d+ + d–.
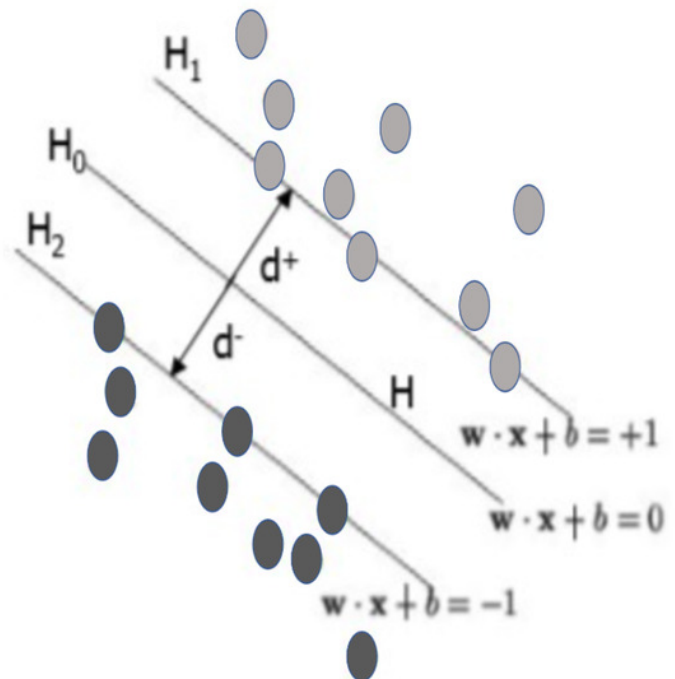
Figure 5

Data Separation linear and non-linear.

A) Linear Separation   B) Non-linear Separation
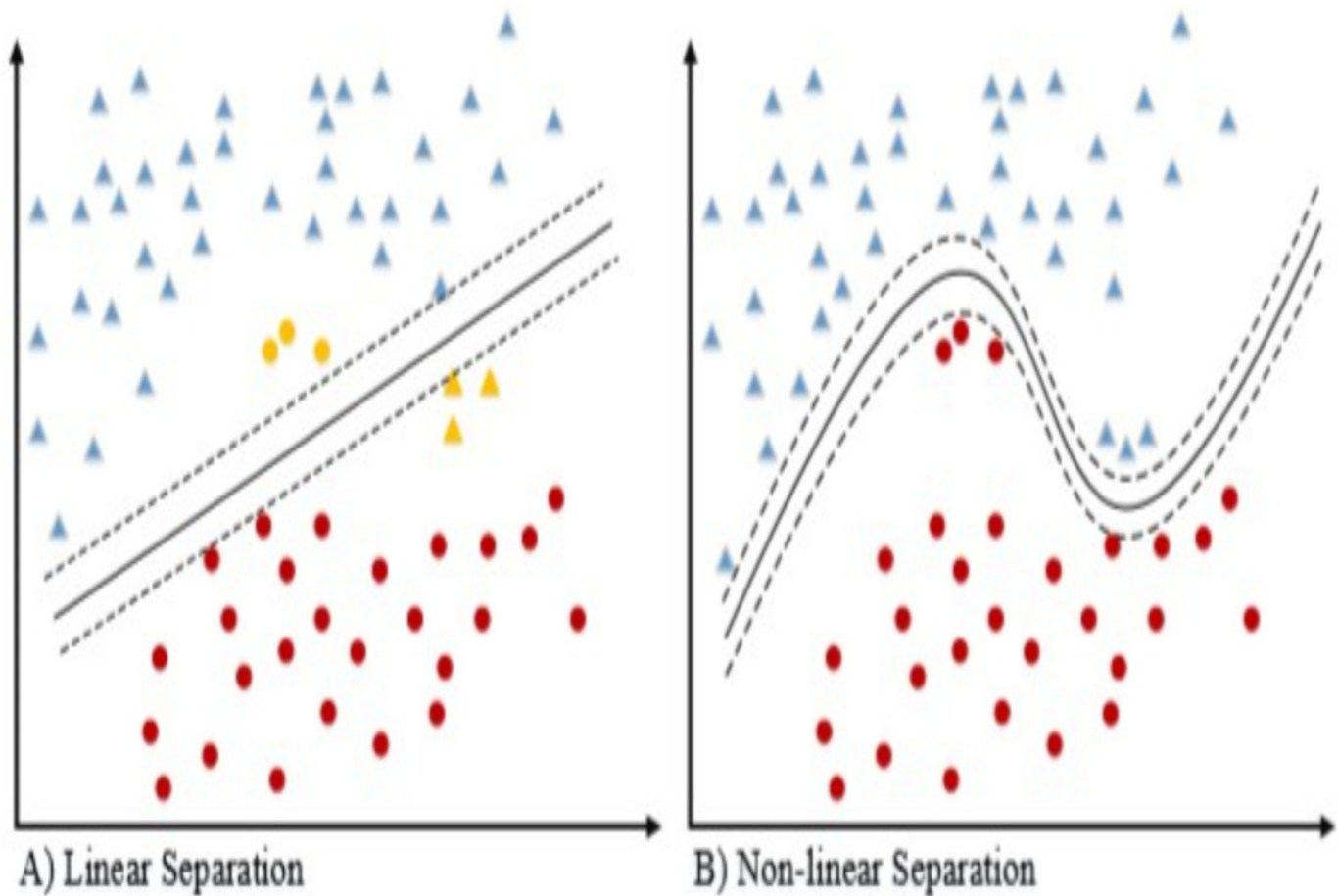
Figure 6

Linear Support Vector Machine.
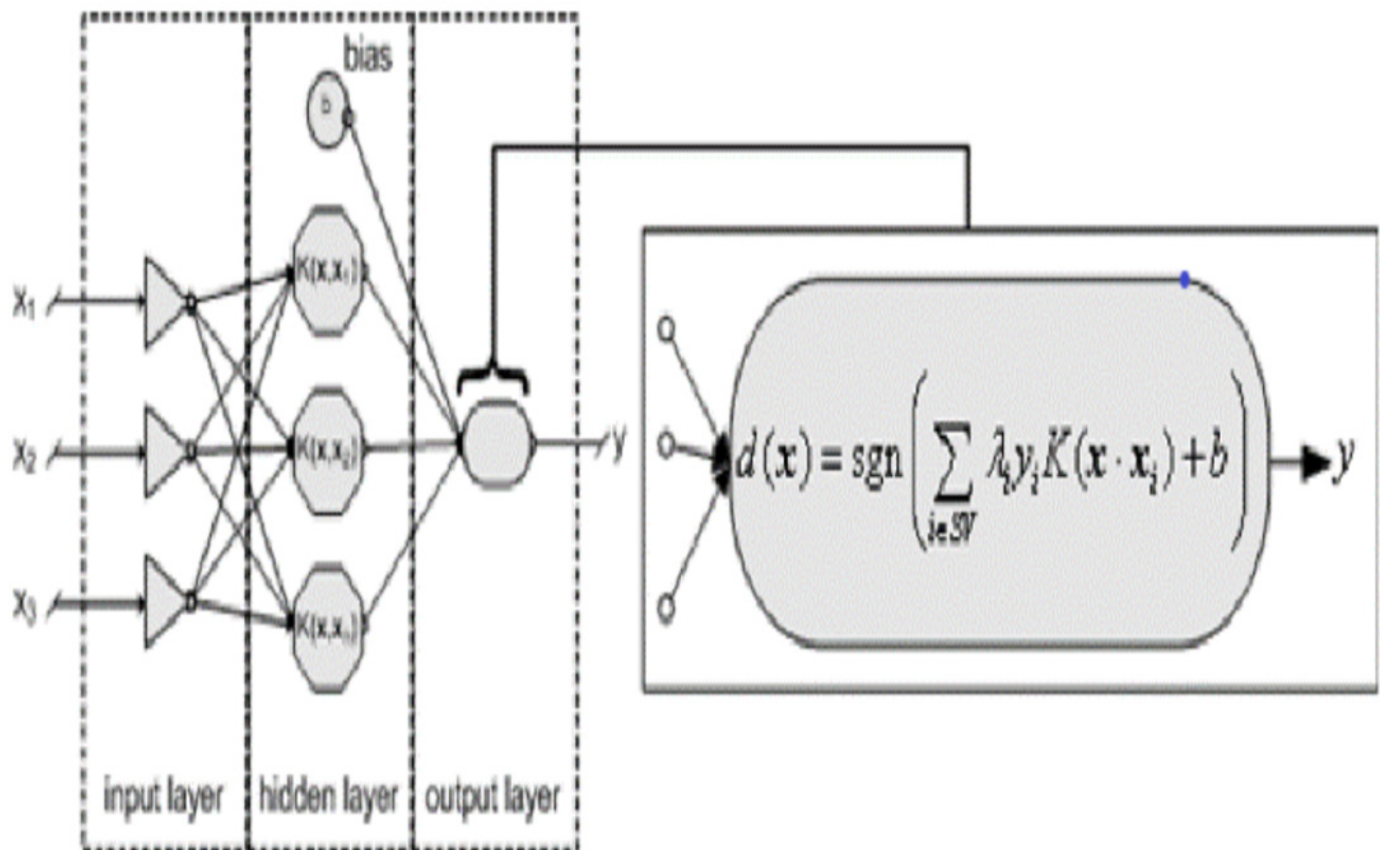
Figure 7

Graphical representation of the CBOW and Skip-gram model.
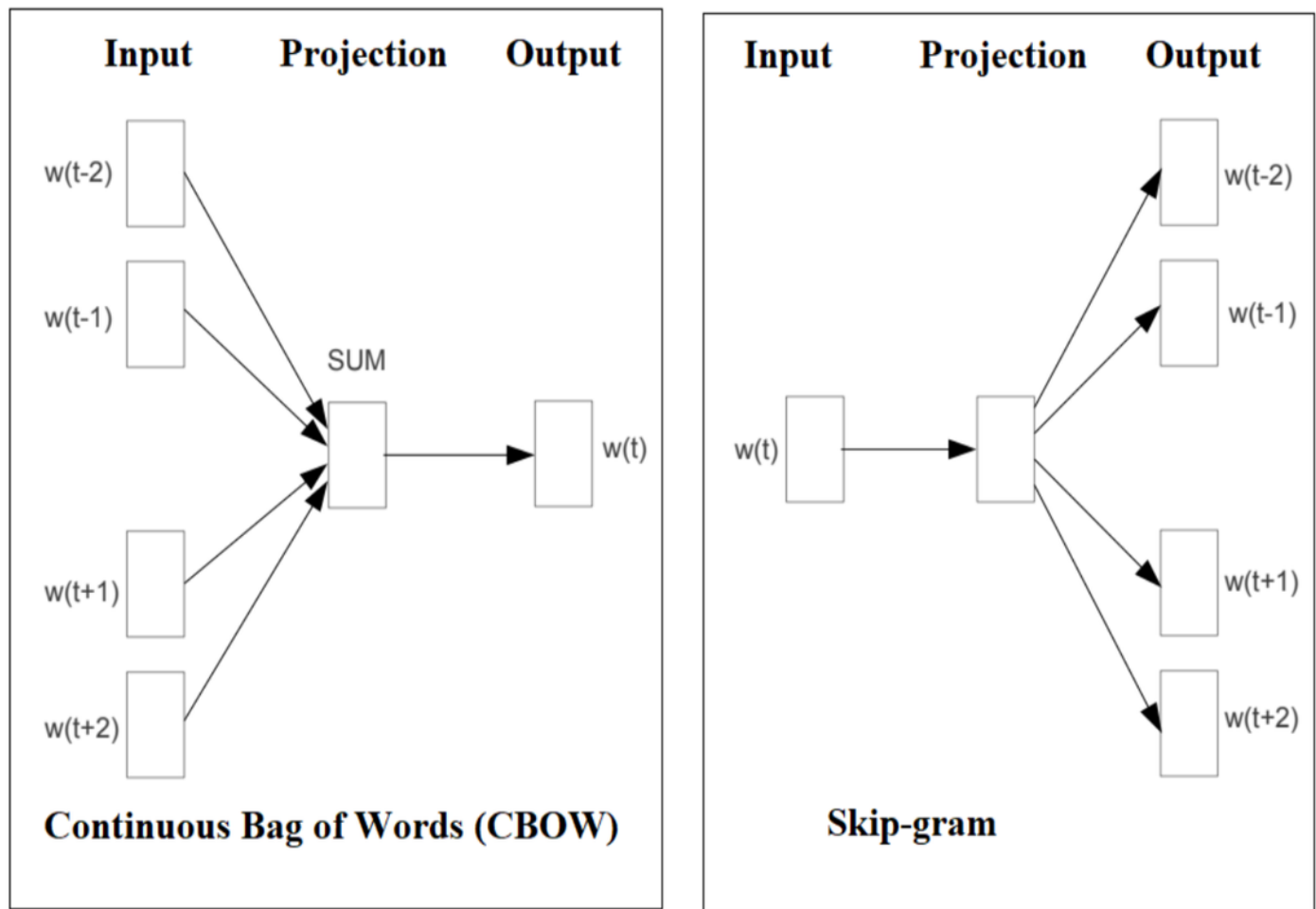
Figure 8

A Simple Linear fastText Architecture.
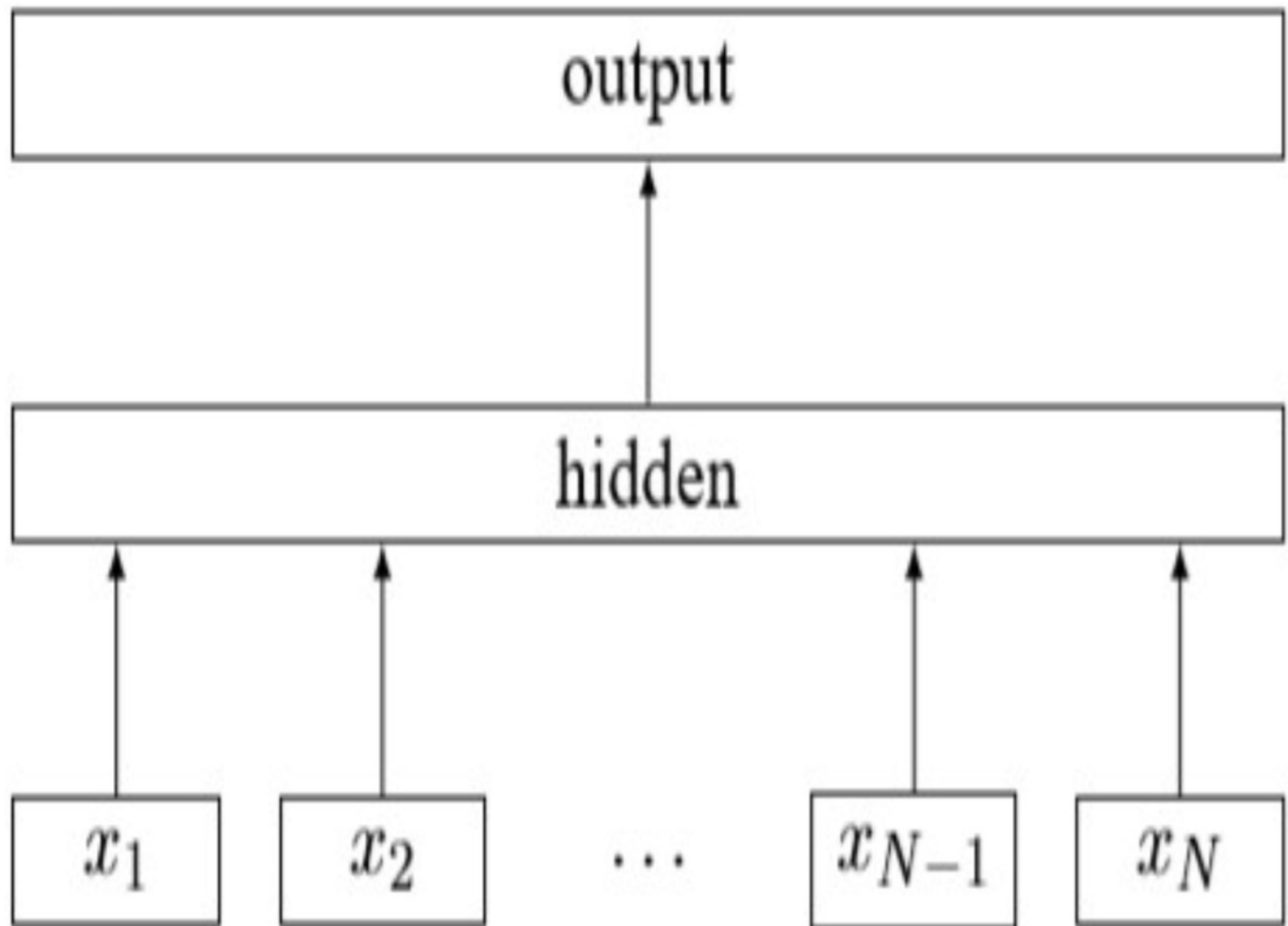
Figure 9

*fast*Text model text document process flow.
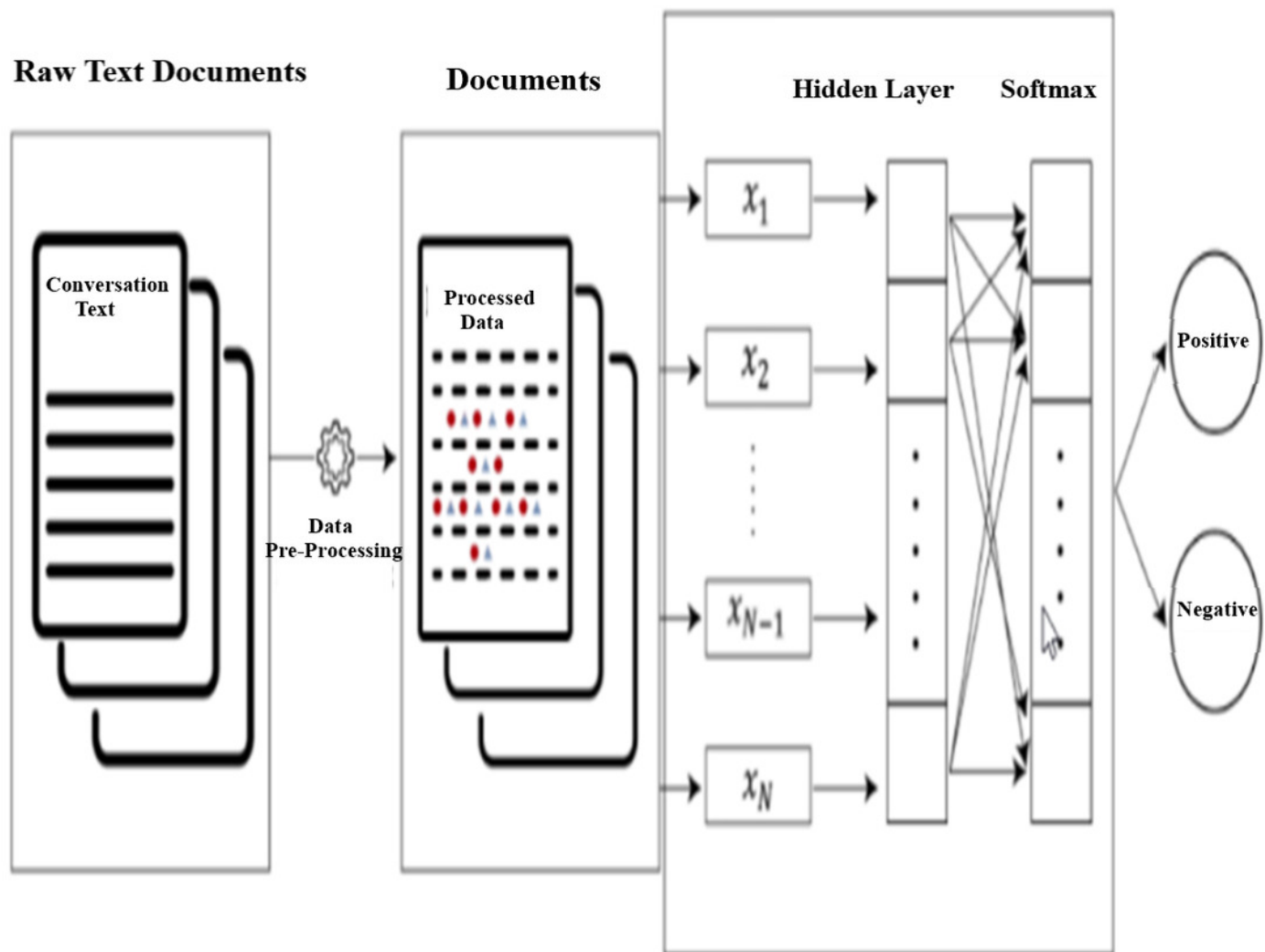
Figure 10

Extended LSTM with multiplicative Forget gate.
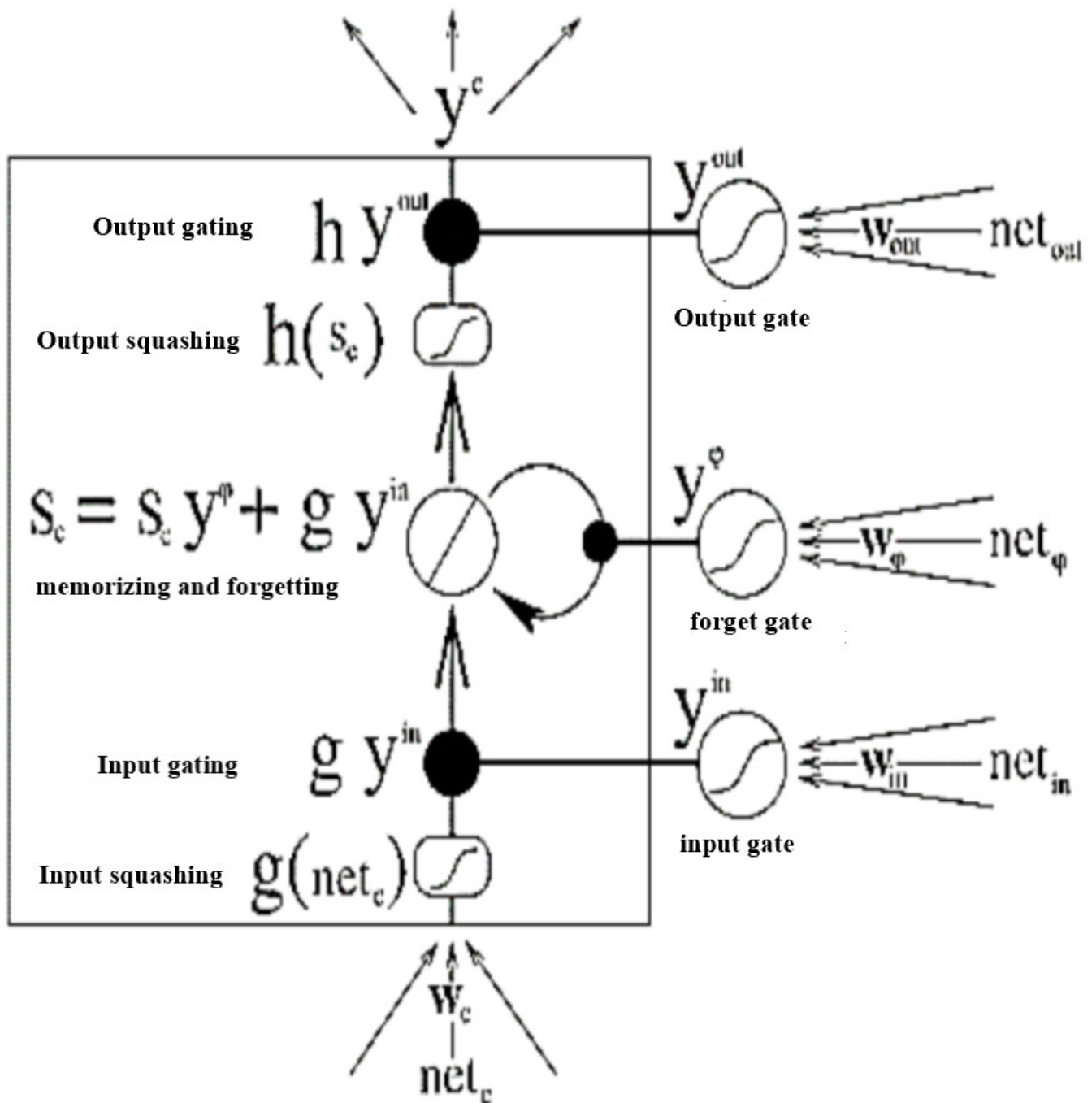
Figure 11

SA-BLSTM mixed-language text data process flow.
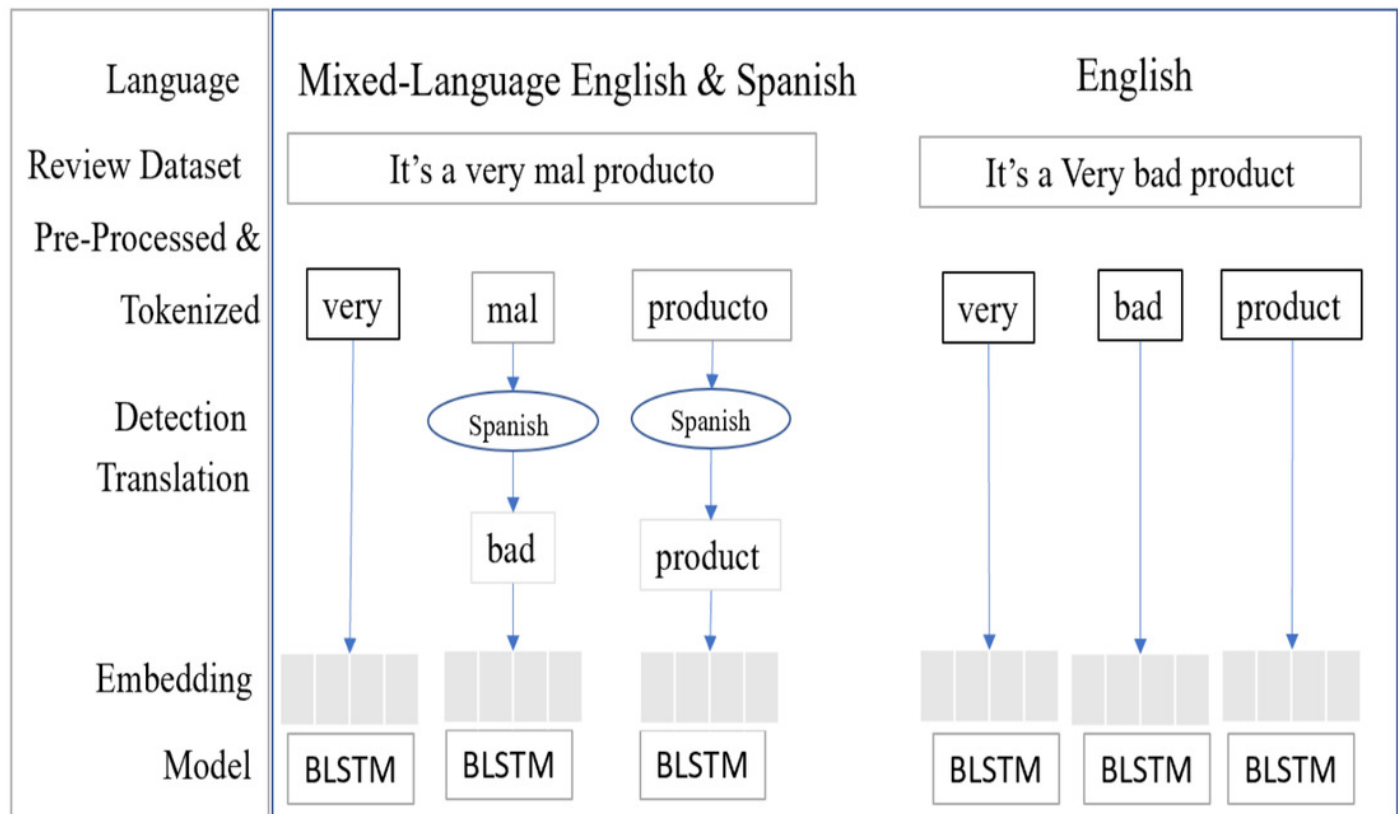
Figure 12

Sentiment Analysis Bi-Directional LSTM Model (SA-BLSTM).
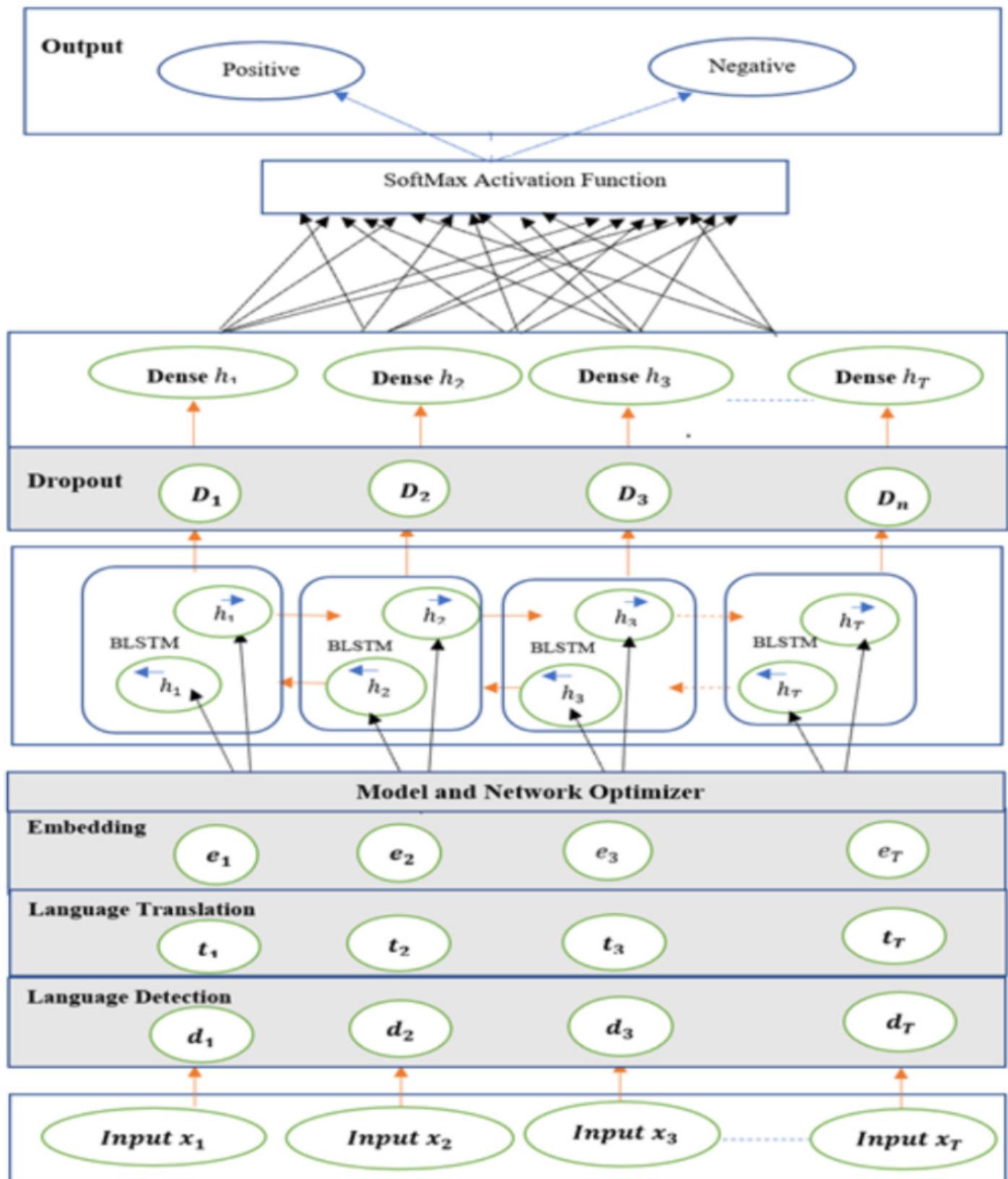
Figure 13

Comparison of related Dataset.

Model	AG	Sogou	DBP	Yelp P.	Amz. P.
BoW (Zhang et al., 2015)	88.8	92.9	96.6	92.2	90.4
ngrams (Zhang et al., 2015)	92.0	97.1	98.6	95.6	92.0
ngrams TFIDF (Zhang et al., 2015)	92.4	97.2	98.7	95.4	91.5
char-CNN (Zhang and LeCun, 2015)	87.2	95.1	98.3	94.7	94.5
char-CRNN (Xiao and Cho, 2016)	91.4	95.2	98.6	94.5	94.1
VDCNN (Conneau et al., 2016)	91.3	96.8	98.7	95.7	95.7
fastText, $h = 10$	91.5	93.9	98.1	93.8	91.2
fastText, $h = 10$, bigram	92.5	96.8	98.6	95.7	94.6

Comparison of related works.

1 **Table 1:**
2 **Comparison of related works.**

Et.al	Dataset	Model	Accuracy %
Geetika Gautam	Twitter Customer Review	SVM Max Entropy Naïve Bayes Semantic Analysis (WordNet)	85.5 83.8 88.2 89.9
Seyed-Ali Bahrainaian	Twitter Data on Smart phones	Hybrid Approach-Unigram, Naive Bayes, MaxEnt, SVM	89.78
Neethu M.S	Twitter data on electronic products	Naïve Bayes SVM Max Entropy Essemble	89.5 90 90 90
Dhiraj Gurkhe	Twitter Data	Unigram Bigram Unigram-Bigram	81.2 15 67.5

Data Sources and URLs.

1   **Table 2:**

2   **Data Sources and URLs.**

Sentiment analysis datasets	Data Source URL
Twitter	https://www.kaggle.com/c/twitter-sentiment-analysis/data
IMDB Movie Review	https://www.kaggle.com/columbine/imdb-dataset-sentiment-analysis-in-csv-format
Yelp	https://www.kaggle.com/omkarsabnis/sentiment-analysis-on-the-yelp-reviews-dataset/data
Amazon	https://www.kaggle.com/saurav9786/amazon-product-reviews
Chat	Our internal team chat conversations dataset
Audio Transcript	Our internal team audio recordings dataset

Models Training performance, tested the pre-trained model for the following parameters setting n=3-gram, epoch=10, lr=0.01 and loss=softmax for fastText model. For LSVM model, kernel=linear, n=3 gram.

1 **Table 3:**
2 **Models Training performance, tested the pre-trained model for the following**
3 **parameters setting n=3-gram, epoch=10, lr=0.01 and loss=softmax for fastText**
4 **model. For LSVM model, kernel=linear, n=3 gram.**

Models	Training duration
LSVM	1.2040 Minutes to train 545041 text documents
fastText	0.7699 Seconds to train 545041 text documents
SA-BLSTM	0.9093 Seconds to train 545041 text documents

Models test and performance measures results with various parameters.

1 **Table 4:**

2 **Models test and performance measures results with various parameters.**

Model	Parameters	Accuracy	Recall	Precision	F1
LSVM	Unigram, Bigram Trigram Kernel=linear	87.74 % 89.96% 90.11%	0.88 0.90 0.90	0.88 0.90 0.90	0.88 0.895 0.896
fastText	Unigram, Bigram Trigram epoch=10, r=0.01, loss=softmax	88.23% 90.55% 90.71%	0.876 0.896 0.896	0.886 0.907 0.910	0.868 0.901 0.902
SA-BLSTM	epoch=10,lr=0.01, loss=softmax	77.00%	0.74	0.79	0.76

Models Sentiment Score.

1 **Table 5:**

2 **Models Sentiment Score**.

Models	Positive Sentiment	Negative Sentiment
LSVM	48.31%	51.67%
fastText	48.49%	51.49%
SA-BLSTM	44.67%	55.31%