# A Literature Review of Context-Based Sentiment Analysis:

Sai Madhur Mallampalli
sbm6433@psu.edu
COMP 597, Adv Topics in NLP,
Spring 2023
Department of Computer Science, Penn State Harrisburg

**Abstract -** Context-based sentiment analysis is an emerging research area in natural language processing (NLP) that focuses on understanding the sentiment of a given text by considering the context in which it appears. There has been a growing interest in this field in recent years due to the increasing need for more accurate sentiment analysis in various applications, such as social media monitoring, product reviews, and political analysis. This literature review aims to provide a comprehensive overview of the state-of-the-art techniques and methods used in context-based sentiment analysis in NLP. The survey review covers various context-based sentiment analysis aspects, including different context types, feature extraction techniques, and machine learning algorithms. Overall, this literature review provides valuable insights into the current state of research in context-based sentiment analysis and its potential applications in real-world scenarios.

## Introduction -

Natural language processing (NLP) has been interested in the study of sentiment analysis for a lengthy period to identify emotions, attitudes, and views in textual data. Traditional sentiment analysis methods, however, frequently fall short when accurately capturing the nuanced nature of textual emotion, especially when profoundly influenced by the context in which it is presented. This has led to the development of context-based sentiment analysis, which considers the various contextual factors that can impact the sentiment expressed in text.

This literature survey review examines the current state of research in context-based sentiment analysis, from the early works in the pre-2010 era to the latest research papers published between 2020-23. The review also intends to explore the techniques and methods given in the literature to incorporate the context of sentiment analysis, including contextual embeddings, contextualized language models, and knowledge graphs. The objective is to examine the different types of context that can influence and potentially alter sentiments, such as linguistic context, situational context, and social context, and the challenges associated with each.

In this review, 12 research papers have been selected covering the latest advancements in context-based sentiment analysis including deep learning models for sentiment analysis on customer reviews, the Generative Pre Training model, and BERT-based pre-training for language understanding. Some earlier works have also been included that have paved the way for current research, such as opinion mining from noisy text data, aspect-based sentiment analysis of movie reviews, and automatic sentiment analysis for web user reviews, including legacy models using lexicons that are used to extract the context in sentences and documents and mixed language data that denote the context. This survey review aims to provide an overview of the progress made in context-based sentiment analysis, highlight key research directions and challenges, and suggest potential avenues for future work.

## Pre-2010:

**An Analytical Approach to Assess Sentiment of Text - Mostafa Al Masum Shaikh, Helmut Prendinger, and Mitsuru Ishizuka (2007):**

Mostafa Al Masum Shaikh et al. (2007)[1] describe an approach to gauge the sentiment of a sentence by enforcing a numerical-valance based analysis. To ascertain the proposition, the authors developed SenseNet, a linguistic tool to provide lexical units owing to the semantic verb-frame generated from the input sentence. A numerical value is assigned to the units by the sense affinity, assign the values using rules, and produce a sense-valence for every sentence. The authors discuss the existing systems used to perform sentiment analysis and state that the models are ineffective for sentence-level predictions.

The proposed approach aims to interpret opinions expressed in the text by employing a contextual valence assessment algorithm. The input to the algorithm is a paragraph, and a set of sentences is processed using a semantic parser. In each sentence, Si is represented as a set of m triplets T.

The system also has a knowledge base that is the information source to build a computational data structure to process the input text. The verbs in the knowledge base are classified into two groups: affective verbs (AV) and non-affective verbs (V) group. Verbs with the tag < effect> in the knowledge base are members of the AV group. AV and V groups are further divided into positive (AVpos, Vpos) and negative (AVneg, Vneg) groups based on their prior valences.

Similarly, adjectives (ADJ), adverbs (ADV), and concepts (CON) are also classified into positive and negative groups. The groups are indicated by ADJpos, ADJneg, ADVpos, ADVneg, and CONpos, CONneg, respectively.

For named entities (NE), the system creates three types of lists: ambiguously named entity (NEambi), positively named entity (NEpos), and negatively named entity (NEneg). A named entity with a different sign for the valence of the 'genre' and 'general sentiment' fields is a member of (NEambi).

The authors chose these two datasets, which contain individual sentences classified as positive, negative, or neutral. The first dataset contained 200 sentences from internet sources such as reviews of products, movies, news, and email correspondences which were scored by 20 human judges who scored each sentence in terms of "Sentiment" and "Intensity" by selecting radio buttons. The second dataset is the sentence polarity dataset v1.0, introduced by Pang and Lee (2005)[2]. It contains 5331 positive and 5331 negative classified sentences or snippets, which are only the subjective opinion sentences of movie reviews. The authors also calculated the inter-rater agreement using Fleiss' Kappa statistics, with a Kappa coefficient of 0.782, indicating good reliability of inter-rater agreement. The system also outperforms (Liu et al., 2003)[3] only under simplified conditions, with 82% and 91.53% accuracy on both datasets, as the previous model was not designed for sentiment recognition.

Overall, the paper uses cognitive and common sense knowledge to give the terms a prior valence as the rules are built using heuristics to use language aspects, making the system reliable.

**Opinion mining from noisy text data - Lipika Dey, Sk. Mirajul Haque (2009):**

Lipika Dey and Sk. Mirajul Haque. 2009[5] propose a semi-automated method to extract and consolidate opinions from noisy text data collected from customer reviews in blogs and feedback at several granular layers.

The article begins by introducing the concept of opinion mining and its importance in today's world. The authors then explain the challenges of opinion mining from noisy text data, including irrelevant information, lack of standardization in language and syntax, and the presence of subjective opinions.

According to the authors, opinion mining analyzes sentiment and extracts opinions from unstructured text documents. Opinion extraction and structurization, which can aid in gathering and analyzing ideas on pre-defined issues, are the essential elements of opinion mining. Identifying the opinion holder, the component or aspect of the topic under consideration, and classifying the opinion are all steps in the opinion extraction process. Structuring entails turning the retrieved opinion expressions into structures that can be assimilated and analyzed. The paper presents the design and analysis of a model designed to perform mining of opinions from blogs and other web sources with spelling errors, ad-hoc abbreviations and irregular casing, and grammatically incorrect.

The system uses a linguistic strategy that uses surface dependence principles to identify opinion statements in cluttered text data. From user-generated data, it can extract and condense opinions about specific features, components of a product, or the product as a whole. It can separate different opinion expressions from a single sentence and give each expression its orientation and that of the entire sentence. The system is initially set up with a seed set and then enlarged to learn domain-specific opinion terms by utilizing WordNet[4] instead of working with a fixed collection of opinion words. The process of creating a knowledge base is semi-automated. They create a novel strategy to deal with adverbial modifiers, which can alter the semantic orientation of the term. The extracted opinions are saved as structured templates, which can be combined with domain ontologies to produce statistics with various levels of resolution.

The paper then reviews the existing literature on opinion mining from noisy text data. The authors identify three main approaches to addressing the challenges of noisy text data: pre-processing techniques, feature selection methods, and machine learning algorithms. Pre-processing techniques involve cleaning and normalizing the text data before analysis. Feature selection methods involve selecting the most relevant features for sentiment analysis. Machine learning algorithms involve training models on labeled data to predict the sentiment of new text data.

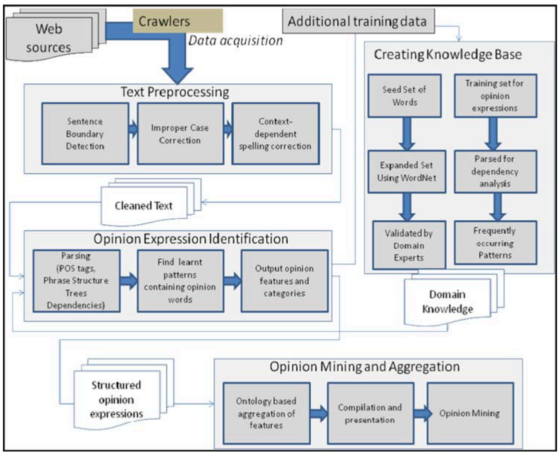



Figure 1: Software architecture of opinion mining system [5]

The data acquisition module has crawlers that crawl websites and store the contents using a pre-defined schema where every blog entry is stored as a data element. The knowledge base module uses a seed set of words, then an expanded one using Wordnet, which domain experts validate. The model also uses a training set for opinion expressions, parsed for dependency analysis to form frequently occurring patterns to formulate the domain knowledge. The noisy text is passed through a pre-processor to be cleaned. Moreover, opinion expression identification is performed by parsing the cleaned text and finding learned patterns with opinion words, and then structured opinion expressions are pushed out as output. In the final phase, the system uses a domain ontology to integrate the extracted opinions and combine the data at various levels of specificity. This enables numerous features like information comparison and information slicing and dicing. They evaluated it on a customers review data set[6] and noisy text from blogs on the Indiacar website[7]. They achieved an accuracy of 0.851 in identifying the opinion of a sentence and 0.898 in opinion orientation at the sentence level.

The paper concludes with a discussion of the limitations of the authors' approach and directions for future research in opinion mining from noisy text data. Overall, the paper provides a comprehensive review of the existing literature on opinion mining from noisy text data and presents a promising approach for addressing the challenges of this task by presenting novel algorithms to determine the orientation of opinion words using WordNet[4].

**Automatic Sentiment Analysis for Web User Reviews - Sun Yueheng, Wang Linmei, Deng Zheng (2009):**

Sun Yueheng et al[8]. propose an approach to conduct sentiment analysis with a focus on the sentiment prediction of ambiguous words based on the context and also devising weight factors based on the part-of-speech and then iteratively widen the sentiment words.

The authors discuss existing systems in the domain that mainly involve two methodologies, either a dictionary or statistics. The paper also discusses how traditional systems are flawed in classifying ambiguous words into the class with the highest frequency. The authors provide an approach to predict the sentiment of a sentence or a document based on the sentiment weights by representing every sentence as a vector of words and then representing every document as a vector of sentences. Thus, they show a model for predicting the sentiment of a sentence or document based on the orientation of the vectors. They also introduce an iterative procedure to extend the initial word sets. They provide an approach based on the Tongyici Cilin, a Chinese synonyms dictionary for ambiguous words.

Automatic discovery of sentiment words forms the basis of sentiment classification in documents. The paper employs the technique of utilizing existing synonyms thesauruses which are based on positive and negative paradigm sets. New words are iteratively classified based on the Tongyici Cilin.

The test data set has close to 20000 user reviews from www.dianping.com, and is preprocessed as sentences are split into a set of words tagged along with their part-of-speech and extract the adjectives, adverbs, nouns, verbs, and conjunctions as key features of a review.

The weights of the words are estimated by considering the impact of adversatives, degree adverbs, and privatives. The paper advocates an iterative expansion of the positive and negative sets based on the weight scores.

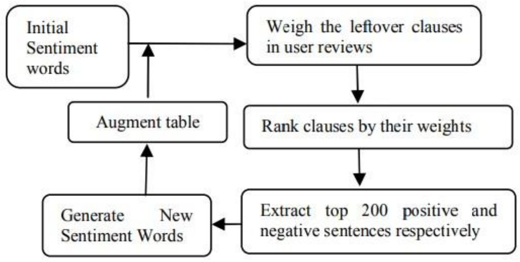



Figure 1. The flow path of iterative approach

The clauses in user reviews are first weighted and are ranked accordingly, where the top 200 sentences are extracted and then used to generate new sentiment words. This procedure is repeated to form an efficient method of classifying the sentiment. When the model is tested on 1219 positive and 1477 negative words, including 216 incorrect words and 255 ambiguous words, and run on 3 epochs, the model has an accuracy of 91.6% and 92.3% and the highest precision of 83.52%. The authors concede the performance is not optimum when only adjectives are used for sentiment analysis.

Overall, the paper provides a commendable and earnest approach in the context of automatic sentiment analysis for user reviews on the web and can be a valuable resource in the domain.

## 2010-14:

**Extracting and Grounding Contextualized Sentiment Lexicons - Albert Weichselbraun, Stefan Gindl, and Arno Scharl (2013):**

In the 2013 work "Extracting and Grounding Contextualized Sentiment Lexicons,"[9] Weichselbraun, Gindl, and Scharl examine the value of contextualized sentiment lexicons in NLP and the difficulties of developing such lexicons.

The paper's major topic is the method used by the authors to extract contextualized sentiment lexicons from online reviews and embeds them in a domain-specific ontology.

The paper begins with an overview of the current state of sentiment analysis and the limitations of existing sentiment lexicons. The authors argue that contextualized sentiment lexicons, which consider the specific context in which a word is used, are necessary for accurate sentiment analysis. They then describe their method for extracting contextualized sentiment lexicons, which combines automatic and manual techniques to identify sentiment-bearing words in online reviews and assign them to specific contexts.

The method finds ambiguous sentiment terms, gathers context terms for each, and then applies them to the sentiment analysis process to make it more accurate. The method uses tagged training corpora to identify ambiguous phrases. Such corpora may be assembled from pre-labeled web reviews or manually produced by reading papers and classifying them as favorable or negative. Such reviews don't require laborious, time-consuming preparation because their writers have already marked them. The system determines the distribution of each lexicon phrase using these prelabeled corpora. Unambiguous phrases are utilized "as is" since they don't benefit from contextualization, and two statistical factors are used to determine the term's ambiguity. If its observed sentiment values show a high standard deviation $(\sigma_{s(t_i)}) \geq v$ and if the deviations from its average sentiment value $(\mu_{s(t_i)}) + \sigma_{s(t_i)} \geq \omega$ and $(\mu_{s(t_i)}) - \sigma_{s(t_i)} \geq -\omega$.

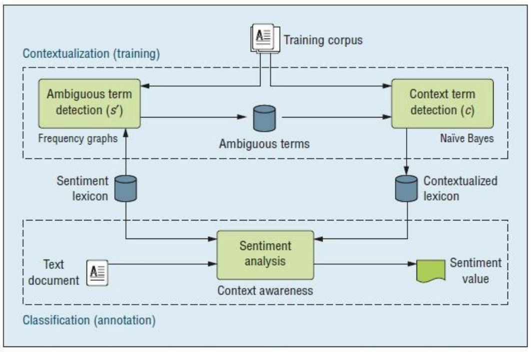


Figure 2. Creating contextualized sentiment lexicons based on tagged training corpora allows identifying and correctly processing ambiguous sentiment terms that shift their polarity depending on their context.

The system gathers context terms and saves them in a sentiment lexicon concerning each discovered ambiguous keyword. The co-occurrence of context phrases in positive and negative documents determines if an ambiguous term is positive or negative. The approach considers every term, regardless of whether it is part of speech or refers to a designated thing. Statistical refinement eliminates irrelevant terms by utilizing only context terms with the highest likelihood of occurring in a positive or negative context. The polarity of an ambiguous term is then estimated using the Naive Bayes method based on the probabilities of the gathered context terms.

$$c = \{c_1, \ldots, c_n\}$$

$$p_t(C_+|c) = \frac{p_t(C_+) \cdot \prod_{i=1}^{n} p_t(c_i|C_+)}{\prod_{i=1}^{n} p_t(c_i)}$$

Context-aware sentiment analysis integrates polarity values for unambiguous and ambiguous terms, identifies negation, and calculates the total sentiment value as the document's overall polarity.

$$s_{\text{total}} = \sum_{t_i \in \text{doc}} n(t_{i-1})\left[s(t_i) + s'(t_i|C)\right]$$

with

$$n(t_{i-1}) = \begin{cases} -1.0 & \text{if } t_{i-1} \text{ is a negation trigger} \\ +1.0 & \text{otherwise.} \end{cases}$$

The function takes the contextualized sentiment lexicon and returns the sentiment score of a term. Concept grounding creates a demarcation between the concepts used in a positive and negative context and a pivot for concepts from semantic data sources such as DBpedia, Freebase, and ConceptNet that aids in intertwining and optimizing the contextualized sentiment lexicon. To overcome the challenges posed by the graphical extrapolation of obscure repositories that give less textual information, the authors use an approach inspired by Roberto Navigli and Mirella Lapata[10] where the system creates a graph from the knowledge repository where nodes represent concepts and the edges represent the relation between them. Post the concept grounding, an algorithm exploits concept information from WordNet to identify relations between ambiguous terms and context terms. Including this external knowledge helps to determine the correct polarity for ambiguous sentiment terms.

To evaluate the procedure, the authors use product reviews from Amazon, TripAdvisor, and the review corpus of Bo Pang and Lilian Lee[11]. They perform both domain-specific and generic contextualization and assess the performance. They observe that the Naive Bayes method provides better results for domain-specific contextualization, and generic contextualization improves with training the lexicon on multiple corpora.

Overall, Weichselbraun, Gindl, and Scharl's paper provide a valuable contribution to natural language processing, highlighting the importance of contextualized sentiment lexicons and offering a method for extracting and grounding them in a domain-specific ontology. The authors' approach can potentially improve the accuracy and usefulness of sentiment analysis in various domain-specific applications.

**Convolutional Neural Networks for Sentence Classification - Yoon Kim, New York University (2014):**

Kim Yoon. 2014[12] proposes an approach to sentence classification using convolutional neural networks (CNNs). Kim's methodology includes a CNN model with a single convolution layer and a max-pooling layer. The input to the model is a matrix of word vectors, where each row corresponds to a word in the sentence. The convolution operation involves a set of filters that slide over the matrix and produce a feature map. The max-pooling operation selects the most salient features from the feature map. The output of the model is a softmax probability distribution over the classes.

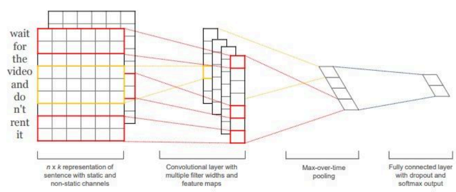



Figure 1: Model architecture with two channels for an example sentence.

$x_i \in R^k$ is the k-dimensional word vector corresponding to the $i^{th}$ word in the sentences. A sentence of length n is represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus \ . \ . \ . \ \oplus x_n$$

$x_{i:i+j}$ refer to a concatenation of words. A convolution operation involves a filter w which is applied to h words in a window to create a new feature.

$$c_i = f(w.x_{i:i+h-1} + b)$$

Here, b $\in R^k$ is a bias variable, and f is a non-linear function. This filter is applied to every window of words in the sentence to create a feature map.

$$c_1 = [c_1, c_2, \ . \ . \ . \ . \ . \ . \ , c_{n-h+1}]$$

The maximum value is chosen as the feature for this particular filter after a max over-time pooling operation has been applied to the feature map. The output of the model, which creates the penultimate layer by using various filters to get different features cascaded into a fully linked softmax layer, is the probability distribution over the labels. The method tests two-word vector channels, one of which is maintained static during training and the other of which is tweaked using backpropagation. Each filter is applied to both channels to determine ci, and the results are added. The model is regularized when a dropout is applied on the penultimate layer with a constraint on the $l_2$ norms of the weight vectors that prevent co-adaption of hidden units by randomly dropping out of a proportion of the hidden units during forward backpropagation.

The model is tested on several benchmarks like SST-1, SST-2, TREC, and others with hyperparameter tuning, and it outperforms models. There are also variations in the model like CNN-rand - where the words are randomly initialized and then modified during training, CNN-static - where the word vectors are kept static, CNN-non-static - where the word vectors are fine-tuned with every task, CNN-multichannel - utilizes two sets of word vectors, where every set is considered as a channel with the filter being applied to both channels. However, the gradients are back-propagated only through one of the channels.

The authors claim that the baseline model (CNN-rand), which uses only randomly initialized words, does not function well. However, a basic model with static vectors (CNN-static) works exceedingly well, providing results that are competitive with those of more complex deep learning models that use sophisticated pooling strategies (Kalchbrenner et al., 2014)[13] or necessitate the computation of parse trees in advance (Socher et al., 2013)[14]. These findings imply that the pre-trained vectors are efficient, 'universal' feature extractors that can be used on many datasets. Further advancements are made by fine-tuning the pre-trained vectors for each task (CNN-non-static).

Although the authors hoped the multichannel architecture would prevent overfitting, especially on smaller datasets, the results aren't encouraging. The regularization of the fine-tuning process is necessary to overcome the shortcomings. There is also an emphasis on static and non-static models, and the multichannel model is found to fine-tune the non-static channel to make it task specific. The usefulness of CNNs for tasks involving sentence classification is demonstrated in the paper. On four separate datasets, the CNN model outperforms conventional models, demonstrating CNNs' promise for use in natural language processing applications. The study also discusses the process of hyperparameter tuning for CNNs and demonstrates how the dataset affects the ideal settings. The introduction of a novel method for sentence classification advances the field of natural language processing as a whole.

**Aspect-Based Sentiment Analysis of Movie Reviews - Viraj Parkhe and Bhaskar Biswas (2014):**

Viraj Parkhe and Bhaskar Biswas (2014)[15] offer a unique approach to aspect-based sentiment analysis (ABSA) to determine the sentiment polarity of particular features mentioned in movie reviews. The authors' method involves the application of domain-specific rules and heuristics to extract attributes and their corresponding polarity. The authors argue that aspect-level sentiment analysis is a more in-depth review analysis as it is focused on different aspects of a review to predict the sentiment. They a method to find out the aspects that precisely define the review's sentiment, and they use the term “driving factors” to determine the weight scores of the different aspects of the movie. Hence, the overall score summarizes all the aspect scores weighted according to the driving factors. They differ from Jianxing Yu et al. 2011. [16]’s approach by using a randomized method instead of the Multivariate Gaussian Distribution to assign the weights driven by the factors.

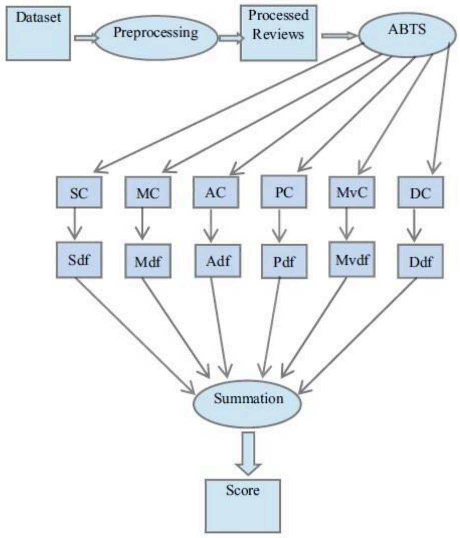



Figure 1. Diagram for the proposed method

ABTS= Aspect Based text Separator SC = Screenplay Classifier, MC = Music Classifier AC = Acting Classifier, PC = Plot Classifier, MvC = Movie Classifier, DC = Direction Classifier. Sdf= Screenplay driving factor multiplication, Mdf= Music driving factor multiplication, Adf = Acting driving factor multiplication, Pdf= Plot driving factor multiplication, Mvdf= Movie driving factor multiplication, Ddf= Direction driving factor multiplication.

The proposed method collects reviews from different sources and preprocesses them to make them appropriate for the method. The preprocessed reviews are passed through the aspect text separator that separates the review based on the screenplay, music, acting, plot, movie, and direction. They use an aspect-specific lexicon to aid in text separation. They use the Stanford Part-Of-Speech tagger [17] to identify the sentence's part of speech before matching the lexicon word inside the phrase that included that part of speech. The aspect-based separated sentences are given as input to a Naive Bayes Classifier. The aspect-based output is multiplied by the relevant driving factor according to the weighted average of the movie's driving factors that denote the importance of a specific aspect in the review. The driving factors follow the relationship $\Sigma\alpha_i = 1$ where $\alpha_i$ is the driving factor of $i^{th}$ a factor. The output obtained is the sum of the classifier outputs multiplied by their corresponding driving factors - $O(d) = \Sigma\alpha_i X_i$ , $X_i = [-1, 1]$ with $\alpha_i$ is the driving factor of the $i^{th}$ aspect and $X_i$ is the output of $i^{th}$ classifier and d is the document that is being reviewed. The threshold score lies in the interval [-1,1], with 0 being the threshold value determining whether the review is positive or negative.

The model is performed on IMDB and Stanford AI Lab Datasets with neutral reviews being omitted. They observe that due to the uneven size of reviews, the aspect separation leads to an unequal text distribution. With a train-test split of 70:30, the model is tested on 1000 iterations, and the driving factors with the highest accuracy form the best driving factors for the specific dataset. Post the experiments, they find that Movie, Acting, and Plot are the high driving factors that ascertain the sentiment of a review, resulting in an accuracy of 79.372%.
In their future work, the authors plan to develop a scoring method incorporating inter-word dependencies based on the genre using a context-specific lexicon. Overall the model can be used to determine the aspects of a movie that enhance the understanding of the sentimental analysis of user reviews.

## 2015-19:

**Sentiment analysis method based on context view - Chengyun Liu, Liancheng Xu, Zhihao Wang, Ningning Wang (2016):**

Chengyun Liu, Liancheng Xu, Zhihao Wang, and Ningning Wang. 2016[18] propose a sentiment analysis method based on the opinion of  data based on the context in their paper titled. The paper addresses the limitations of traditional emotional analysis methods on micro-blogs that ignore context information. To address this problem, the the authors propose a three-step process of calculating emotional weight for a given tweet, considering the context related to the tweet. The proposed approach involves establishing an emotional dictionary of micro-blogs and adding popular expressions to the dictionary. The micro-blog texts are then categorized into the text section, the emotions section, and the context related to a tweet. Finally, sentiment analysis is carried out in these three parts.

The paper presents an algorithm for calculating the emotional tendency value of a given tweet. The algorithm consists of four steps, with step three being divided into three sub-steps. The authors demonstrate their method's effectiveness in sentiment analysis by comparing its performance to traditional machine learning methods. The experimental results show that the proposed method significantly improves accuracy compared to traditional methods.
Sen($P$) = $\alpha$Sen($Pe$) + $\beta$Sen($Pc$) + $\gamma$Sen($Pt$) (1)
In formula (1), Sen($P$) represents the complex emotional weight of current tweet. Sen($Pe$) represents the original emotional weight of the current tweet. Sen($Pc$) is the emotional weight of the context related to the current tweet. Sen($Pt$) is the emotional weight of the original tweet. The parameters $\alpha$, $\beta$, $\gamma$ are

all positive and between 0 and 1. The sum of these parameters is equal to 1: $\alpha + \beta + \gamma = 1$.

$$Sen(P_{e1}) = \frac{1}{n} \sum_{i=1}^{n} Sen(P_{e1i}) (2)$$

In formula (2), Sen($Pe1$) is the emotional weight of microblog texts calculated by emotions. Sen($Pe1i$) represents the emotional weight of emotion $i$.

$$Sen(P_{e2}) = \frac{1}{n} \sum_{i=1}^{n} Sen(P_{e2i}) (3)$$

In formula (3), Sen($Pe2$) is the emotional weight of microblog texts calculated by feature words. Sen($Pe2i$) represents the emotional weight of the feature word $i$.

Step 1:

$$Sen(P_e) = Sen(P_{e1}) + Sen(P_{e2}) \ (4)$$

Step 2:

$$Sen(P_c) = \sum_{i=1}^{n} m(Sen(P^i_{c1}) + Sen(P^i_{c2})) \ (5)$$

$$m = 1 - \frac{i-1}{10} (6)$$

In formula (5), $i$ represents the position information of context related to current tweet. The distance between context position i and the current tweet is $i$ as well. Then, Sen($P^i_{c1}$) and Sen($P^i_{c2}$) are the emotional tendency values of emotions and feature words in position $i$, respectively.

$$Sen(P_t) = P_{fa}(Sen(P_{t1}) + Sen(P_{t2})) \ (7)$$

$Sen(P_{t1})$ and $Sen(P_{t2})$ will also adapt the calculation method in formula (2) and (3). $Sen(P_{t1})$ and $Sen(P_{t2})$ are emotional tendencies of emotions and feature words in original tweets respectively.

$$SV(S_i) = \alpha Sen(P_e) + \gamma Sen(P_t) \ (8)$$

In formula (8), $S$ is regarded current tweet and the context related to current tweet and then labeled the content in $S$ as ($S_1 S_2 S_n$) according to its forwarding order and at the same time the emotional tendency values of a current tweet is set as SV($S_i$).

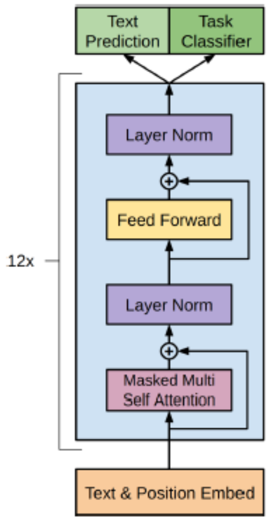
The experiments were conducted using the COAE2015 evaluation task with Naive Bayes Method and SVM Method as baselines. The method proposed in the paper has a total precision of 0.743 which is higher than the second-best model by 0.142. The proposed method is innovative and highlights the importance of considering contextual factors in sentiment analysis. Using emotional weight to capture the sentiment of individual words and phrases and the context of the tweet is a novel approach. The paper provides a well-organized and easy-to-follow algorithm for sentiment analysis, which is a significant contribution to the field. Overall, the proposed methodology can potentially be utilized for sentiment analysis in micro-blogs.

**Improving Language Understanding by Generative Pre-Training - Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever (2018):**

Radford et al. (2018)[19] propose a methodology for pre-training a language model using unsupervised learning and fine-tuning it on various language tasks. The paper introduces the Generative Pre-trained Transformer (GPT) model, which achieves state-of-the-art results on language modeling tasks. The authors claim that pre-training the GPT model on a large corpus of unlabelled text data improves its performance on language tasks.

To demonstrate the effectiveness of the pre-training methodology, the authors customize the pre-trained GPT model on multiple NLP Tasks. Compared to previous works, the GPT model presented in the paper achieves state-of-the-art results on language modeling tasks and demonstrated the effectiveness of pre-training and fine-tuning language models for various language tasks, including sentiment analysis. The paper also introduces the concept of transfer learning in natural language processing, where a pre-trained model can be fine-tuned on a specific task, resulting in improved performance.

The GPT model builds on previous works, such as the Transformer model (Vaswani et al., 2017), by pre-training a language model on a large corpus of unlabelled text data. The pre-training methodology used in the GPT model involves no supervision, allowing the model to capture contextual relationships between words as the tokens go through 12 layers of the Transformer model. The fine-tuning methodology involves adding a task-specific layer on top of the pre-trained model and fine-tuning the model on a labeled dataset for the specific task.




(1)

Figure (1) describes the transformer architecture and training objectives used in this work. It closely follows the Transformers' work and is trained as a 12-layer decoder-only transformer with masked self-attention heads (768-dimensional states and 12 attention heads). For the position-wise feed-forward networks, 3072-dimensional inner states are used. Adam optimization scheme with a max learning rate of 2.5e-4 is employed. The learning rate is increased linearly from zero over the first 2000 updates and annealed to 0 using a cosine schedule. The model is trained for 100 epochs on mini-batches of 64 randomly sampled, contiguous sequences of 512 tokens. Since layer norm is used extensively throughout the model, a simple weight initialization of N(0; 0:02) is sufficient - using Gaussian distribution with mean 0 and variance 0.02.

Unsupervised Pre-Training:

For an unsupervised corpus of tokens, they use a standard language modeling objective to maximize the likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

Then for the language model, they use a multi-layer Transformer decoder, but it is a variation of the existing model. They apply a multi-headed self-attention operation over the input tokens, which then goes through position-wise feedforward layers to give an output distribution over a set of target tokens.

$$h_0 = UW_e + W_p$$
$$h_l = \mathtt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$
$$P(u) = \mathtt{softmax}(h_n W_e^T)$$

Supervised Fine Tuning:
After training the model, they update the parameters to the supervised target task. They use a labeled dataset C, where every instance has a token sequence as an input. The sequence is passed through the pre-trained model to obtain the activation block for the final transformer, which is eventually fed into an added linear output layer with parameters to predict y.

$$P(y|x^1, \ldots, x^m) = \mathtt{softmax}(h_l^m W_y).$$

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$

They use language modeling as an auxiliary objective to the fine-tuning

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

In the context of sentiment analysis, the GPT model has become a popular choice for building context-based sentiment analysis models. Its ability to capture natural language's complex structure and semantics has made it highly effective in analyzing sentiment in text data. The fine-tuned GPT model presented in the paper achieves a new state-of-the-art accuracy score on 9 out of the 12 datasets.


In conclusion, "Improving Language Understanding with Unsupervised Learning" by Radford et al. (2018) presents a groundbreaking methodology for pre-training and fine-tuning language models that has since inspired numerous advancements in natural language processing including successive versions of GPT which is a watershed moment in the history of NLP. The paper's results demonstrated the effectiveness of the GPT model in improving language understanding and performance on various language tasks, including sentiment analysis. The GPT model's pre-training methodology builds on previous works and introduces the concept of transfer learning in natural language processing, which has become a standard approach in the field.

**BERT - Bidirectional Encoder Representations from Transformers - Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019):**

BERT stands for Bidirectional Encoder Representations from Transformers, is a pre-training language model that was proposed by Devlin et al. in 2018[20]. The paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" describes BERT's architecture and pre-training tasks and presents its performance on a range of benchmark tasks.

One of the key contributions of BERT is its ability to capture both contextual and global information and effectively handle long-term dependencies in language. This is achieved through a deep bidirectional transformer architecture that can pre-train a language model on large amounts of unannotated text. Unlike previous language models that were either unidirectional or shallow, BERT leverages a transformer architecture that allows for efficient parallel processing and captures the bidirectional relationships between the input tokens.

BERT's pre-training process consists of two tasks: masked language modeling (MLM) and next sentence prediction (NSP). MLM involves randomly masking some of the input tokens and predicting them based on the context of the remaining tokens. At the same time, NSP involves predicting whether two sentences in a pair are adjacent. The MLM task enables BERT to learn representations that capture the relationships between different sentence parts. In contrast, the NSP task allows the model to understand the relationships between different sentences.

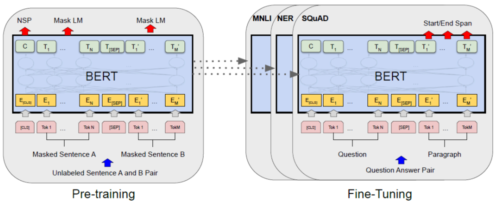



Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

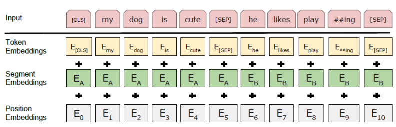



Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

The authors evaluate BERT's performance on various benchmark tasks, including the General Language Understanding Evaluation (GLUE) benchmark, the Stanford Question Answering Dataset (SQuAD), and the CoNLL-2003 named entity recognition task. They show that BERT outperforms previous state-of-the-art models on all these tasks, achieving new state-of-the-art results on several of them. BERT achieves a GLUE score of 80.5, 7.7 points higher than the previous state-of-the-art.

The authors also conduct a detailed analysis of BERT's performance and provide insights into the model's inner workings. They show that BERT's success can be attributed to its ability to effectively capture contextual and global information and handle long-term dependencies. They also demonstrate the importance of the layer-wise pre-training strategy, the attention mechanism's effectiveness, and the pre-training objectives' significance.

Moreover, the authors also conduct an ablation study to analyze the importance of various model components, including the transformer architecture, the pre-training objectives, and the layer-wise pre-training strategy. They observe that all these components were important for achieving the best performance, with the transformer architecture being the most critical.

Overall, the paper presents a significant advance in natural language processing, providing a new state-of-the-art model for a range of benchmark tasks. The technical details of BERT's architecture and pre-training process are well explained, and the results are impressive, demonstrating the power of deep bidirectional transformers in language modeling. The success of BERT has since inspired numerous follow-up works and has led to significant advancements in natural language processing.

## 2020-23:

**A New Sentiment Analysis Model for Mixed Language using Contextual Lexicon - Nurul Husna Mahadzir, Nor Hafizah Abdul Razak, Mohd Faizal Mohd Omar (2020):**

The study conducted by Nurul Husna Mahadzir et al. 2020[21] proposes a new sentiment analysis model for mixed language text using a contextual lexicon. The research aims to overcome the limitations of existing sentiment analysis models designed to analyze only monolingual texts. The model aims to identify the sentiment expressed in a mixed-language text by considering the context and the lexicon of the language used. It aims to tackle the mixed language ambiguity problem, emphasizing scenarios when one particular word is in multiple languages.

The paper deals with Lexicon-Based Sentiment Classification with a prime focus on identifying the polarity of words based on context. Another significant change is discarding the usual translation practice to a single language and generating a contextual sentiment lexicon specific to a mixed language environment for the classification task.

The authors ascertain a lexicon-based approach for the bilingual(Malay-English) implementation of sentiment analysis using the Twitter dataset.

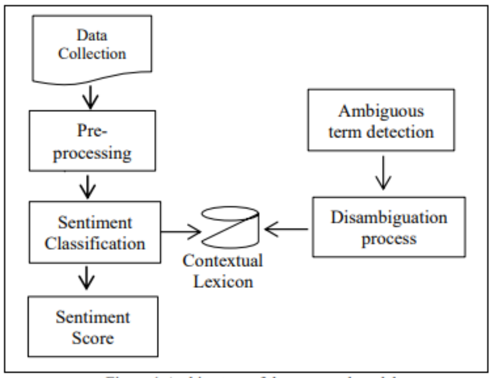


Figure 1 Architecture of the proposed model

The collected dataset consists of tweets in both Malay and English. The data is preprocessed by removing noisy aspects like URLs and stopwords. The contextual lexicon is constructed after detecting ambiguous terms, followed by a disambiguation process. To obtain a correct sense of an ambiguous term, a method is applied where the WordNet database for extracting word senses according to the context from WordNet glosses is used. Then, the process is evolved to find the similarity between sentiment and WordNet Glosses by mapping matrices of sentiment to obtain the correct sense and then revise the polarity score based on the correct sense obtained from previous steps to calculate the final score.

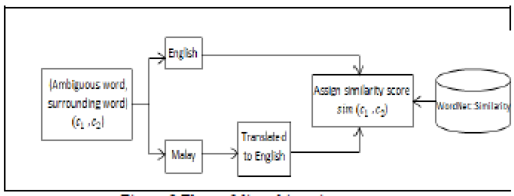


Figure 2 Flow of disambiguation process

The ambiguous word *c*2 with its surrounding words *c*2 are considered, and semantic relatedness between word pairs (*c*1,*c*2) is calculated using a formula to find the similarity score.

$$PR = \frac{PWS}{TWS} \quad (1)$$

$$NR = \frac{NWS}{TWS} \quad (2)$$

$$Sentiment\ score = PR - NR \quad (3)$$

where;
$PR$ = Positive Ratio, NR= Negative Ratio,
$PWS$ = Number of Positive words in each sentence,
$NWS$ = Number of Negative words in each sentence,
$TWS$ = Number of words in each sentence.

In conclusion, the paper highlights the importance of sentiment analysis in today's world and the challenges faced in analyzing the sentiment expressed in mixed-language texts. Its potential to overcome the limitations of existing models on the specific sphere of mixed-language models through the mapping process is a significant contribution to sentiment analysis and provides insights into developing more accurate and efficient models for analyzing sentiment in mixed-language texts.

**Context-Guided BERT for Targeted Aspect-Based Sentiment Analysis - Zhengxuan Wu, Desmond C. Ong (2020):**

Wu and Ong (2020)[22] put forward a new approach to improve the performance of aspect-based sentiment analysis (ABSA) by incorporating contextual information into the BERT model. The paper is designed to answer two research questions: (1) how to integrate contextual information into BERT to enhance targeted ABSA, and (2) whether their approach could outperform existing models on benchmark datasets.

To address these questions, the authors use a contextual-guided BERT model that encodes the input text with pre-trained BERT embeddings and passes it through several neural network layers to generate sentiment scores for each target aspect.

The authors pivot on defining ABSA and TABSA tasks to populate the model. Aspect-Based Sentiment Analysis (ABSA) is an NLP task that aims to identify and extract the sentiment of specific aspects or components of a product or service, and TABSA( Targeted Aspect Based Sentiment Analysis) is a revision of the ABSA model with a key focus on the targeted context. CG Bert, or Context Guided Bert, is an improvisation on the context-aware Transformer model conceived by Yang et al. (2019)[23] and used in the TABSA task. This model is especially used to find Softmax attention.

$$\mathbf{A}^{h}_{\text{Self-Attn}} = \text{softmax}\left(\frac{\mathbf{Q}^h \mathbf{K}^{h^T}}{\sqrt{d_h}}\right) \quad (1)$$

where $\mathbf{Q}^h \in \mathbb{R}^{n\times d}$ and $\mathbf{K}^h \in \mathbb{R}^{n\times d}$ are query and key matrices indexed by head $h$, and $\sqrt{d_h}$ is a scaling factor. We integrate context into BERT by modifying $\mathbf{Q}$ and $\mathbf{K}$ matrices of the original BERT model (Devlin et al. 2019):

$$\begin{bmatrix}\hat{\mathbf{Q}}^h \\ \hat{\mathbf{K}}^h\end{bmatrix} = \left(1 - \begin{bmatrix}\lambda^h_Q \\ \lambda^h_K\end{bmatrix}\right)\begin{bmatrix}\mathbf{Q}^h \\ \mathbf{K}^h\end{bmatrix} + \begin{bmatrix}\lambda^h_Q \\ \lambda^h_K\end{bmatrix}\left(\mathbf{C}^h \begin{bmatrix}\mathbf{U}_Q \\ \mathbf{U}_K\end{bmatrix}\right) \quad (2)$$

where $C^h \in R^{n\times d_c}$ is the context matrix for each head and $\{\lambda^h_{Q}, \lambda^h_{K}\} \in R^{n\times 1}$ is learned context weights, and $\{U_Q, U_K\} \in R^{d_c\times d_h}$ are weights of linear layers used to transform input context matrix $C^h$. The modified Q^ and K^ are then used to calculate context-aware attention weights using the dot-product of both matrices.

The second neural network conceived in the paper uses a Quasi Attention function for (T)ABSA. The value of the self-attention weights in a vanilla implementation restricts the boundaries between 0 and 1, allowing the hidden states to contribute only additively but not subtractively. Hence, they develop a quasi-attention calculation for both operations to the attended vector. The attention matrix is formulated as

$$\hat{\mathbf{A}}^h = \mathbf{A}^h_{\text{Self-Attn}} + \lambda^h_A \mathbf{A}^h_{Quasi\text{-Attn}} \quad (4)$$

where $\lambda^h_A$ is a scalar to represent the compositional factor to control the effect of context on attention calculation. $\mathbf{A}^h_{\text{Self-Attn}}$ is defined as in Eqn. 1. To derive the quasi-attention matrix, we first define two terms quasi-context query $\mathbf{C}^h_Q$ and quasi-context key $\mathbf{C}^h_K$:

$$\begin{bmatrix} \mathbf{C}^h_Q \\ \mathbf{C}^h_K \end{bmatrix} = \mathbf{C}^h \begin{bmatrix} \mathbf{Z}_Q \\ \mathbf{Z}_K \end{bmatrix} \quad (5)$$

where $\{\mathbf{Z}_Q, \mathbf{Z}_K\} \in \mathbb{R}^{d_c \times d_h}$ are weights of linear layers to transform the raw context matrix, and $\mathbf{C}^h$ is the same context matrix in Eqn. 2 (Defined in Sec. 3.6). Next, we define the quasi-attention matrix as:

$$\mathbf{A}^h_{Quasi\text{-Attn}} = \alpha \cdot \text{sigmoid}\left(\frac{f_\psi(\mathbf{C}^h_Q, \mathbf{C}^h_K)}{\sqrt{d_h}}\right) \quad (6)$$

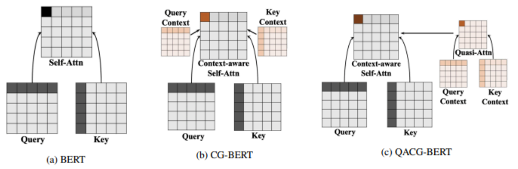



Figure 2: Illustration of the proposed models. (a) The vanilla self-attention network (e.g., BERT) calculates attention weights using the query and key matrices without considering context. (b) The CG-BERT model modifies query and key matrices using context, and then calculate attention weights as in (a). (c) The QACG-BERT model calculates attention weights by combining vanilla attention weights as in (a) with quasi-attention weights calculated using a separate pair of query and key matrices for context. Colors in the grids illustrate matrix operations.

The authors train their model on three benchmark datasets: SemEval 2014, SemEval 2015, and Twitter. They compare their model's performance with several existing models and report that it achieves state-of-the-art results on all datasets. Additionally, they performed an ablation study to examine the contribution of each component of their model. QACG-BERT outperforms CG-BERT, BERT, and other legacy models with an accuracy score of 93.8% on the SentiHood TABSA dataset. The model performs the best with the inputs appended with auxiliary sentences, with a 95.8% for binary sentiment analysis.

The authors' approach has made several significant contributions to the field of ABSA. First, they demonstrate the effectiveness of integrating contextual information into BERT to improve the performance of targeted ABSA. Second, they achieve state-of-the-art results on benchmark datasets, indicating the superiority of their model over existing approaches. Third, their ablation study provided insights into the effectiveness of their approach and sheds light on the contribution of each component of their model. In conclusion, Wu and Ong's paper presents a novel approach to ABSA using contextual-guided BERT. Their work has predominant implications for natural language processing and machine learning research.

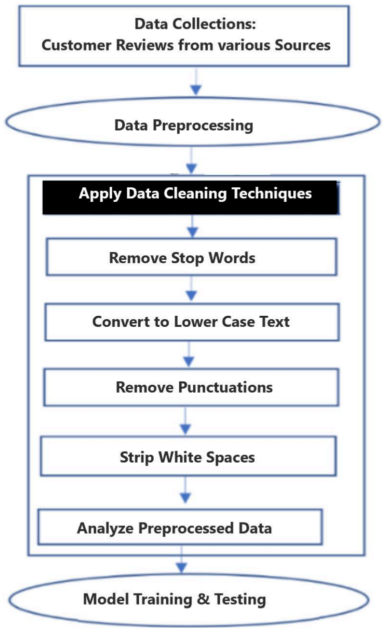
**Context-based sentiment analysis on customer reviews using machine learning linear models - Anandan Chinnalagu and Ashok Kumar Durairaj (2021):**

Chinnalagu, A. & Durairaj, A.K. (2021)[24] in Context-based sentiment analysis on customer reviews using machine learning linear models propose a machine learning linear model approach for sentiment analysis on customer reviews with primary attention on the context of the reviews to overcome the incoherence in polarity, model overfitting and performance issues, as well as high cost in data processing which are constant hassles in 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. The experiment conducted in the paper aims to solve these revealing issues by building a high-performance yet cost-effective model for predicting accurate sentiments from large datasets containing customer reviews.

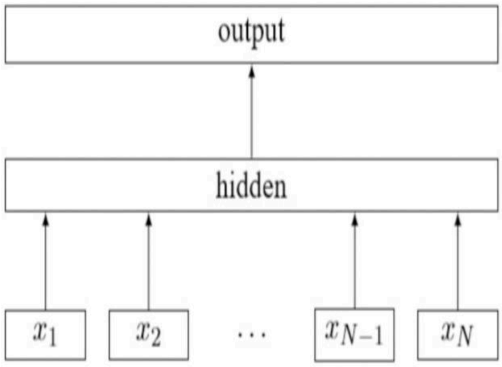
This paper uses the fastText library from Facebook's AI research (FAIR) Lab and the traditional Linear Support Vector Machine (LSVM) to classify text and word embeddings. Additionally, the authors compare their proposed fastText model with their custom multi-layer Sentiment Analysis (SA) Bi-directional Long Short-Term Memory (SA-BLSTM) model. The results showed that the new fastText model achieved a higher accuracy of 90.71% and a 20% improvement in performance compared to the LSVM and SA-BLSTM models.

- The study proposes a multi-layer SA model with Bi-LSTM, fastText, and LSVM to address issues with context-based sentiments. The authors customized data pre-processing to suit customer review and speech transcript datasets.
- To tackle vocabulary issues arising from datasets containing mixed language texts, the paper adds input and service layers to detect the baseline language, translate the text into English, and generate a transcript.
- To avoid out-of-vocabulary (OOV) issues, the paper adds Transcript and Translate service layers to the model's input layer to train and test domain-based mixed-language datasets.
- The paper provides new data pipeline techniques that help save costs while improving the performance of training models with large datasets.
- The hyperparameters of LSVM, fastText, and SA-BLSTM models are fine-tuned based on the dataset.

The main objective is to find a combination of the right methodologies that explains the usage of text classification algorithms to overcome sentiment analysis models' accuracy and performance problems posed by models built on Neural Networks.




The model extensively uses FastText, an open-source library developed by FAIR lab at Facebook for text representation and classification that is lightweight and can run on standard hardware with a multicore CPU. FAIR evaluates the fastText approach for tag prediction and sentiment analysis, showing that it is often on par with recently proposed Deep Learning methods regarding accuracy and performance and is faster for training and evaluation. FastText introduces a new word embedding approach that extends the continuous skip-gram and Continuous Bag of Words (CBOW) models, such as word2vec. Each word is represented as a bag of character n-grams, and fastText treats a vector representation as associated with each character n-gram. The original version of fastText is trained on Wikipedia and is available in 294 languages. The main difference between word2vec and fastText is that fastText sees words as the sum of their character n-grams, allowing it to calculate embeddings for out-of-vocabulary words. In contrast, word2vec treats words as the minimal entity and tries to learn their respective embedding vector.




A simple linear fastText architecture.

The process involves embedding and averaging the text features to create a hidden variable by a simple neural network with a single layer. Firstly, the bag-of-words representation of the text is input into a lookup layer where the embedding of each word is retrieved. Then, these embeddings are averaged to obtain a single average embedding for the entire text. At the hidden layer, the model has n words multiplied by the dimension of the embedding, where n words represent the vocabulary size. After the averaging, a single vector is obtained and passed through a linear classifier. The output of the input layer is transformed linearly by applying the softmax over a matrix with dimension 1 x N output, where N output represents the number of output classes. The negative log-likelihood function of the fastText model is represented by the equation -

$$-\frac{1}{N} = \sum_{n=1}^{N} y_n \log(f(BAx_n)).$$

Here, $x_n$ represents the n-gram feature of the word, A represents the lookup matrix of the word embedding, B represents the linear output of the model transformation, f represents the softmax function. The softmax function calculates the probabilities distribution of the event over n different events. The softmax takes a class of values and converts them to probabilities with sum 1. So, it effectively squashes a k-dimensional vector of arbitrary real values to a k-dimensional vector of real values within the range 0 to 1.

The experiments also employ the SA-BLSTM model, a bidirectional LSTM. On applying both the models on multiple datasets, the authors observed that FastText with epoch = 10, r = 0.01 SA-BLSTM and LSVM with an accuracy of 90.71% for trigrams outperformed the other two models with a staggering 20% difference in the case of SA BLSTM and 0.5% against LSVM. Additionally, the authors suggest extending the model to other languages and exploring the possibility of transfer learning. Finally, the study recommends exploring other datasets to test the proposed model's generalization capability. Hence, it is an effective model that enunciates the importance of complementing the usage of existing architecture with LSTMs to generate favorable results.

## Code Implementation:

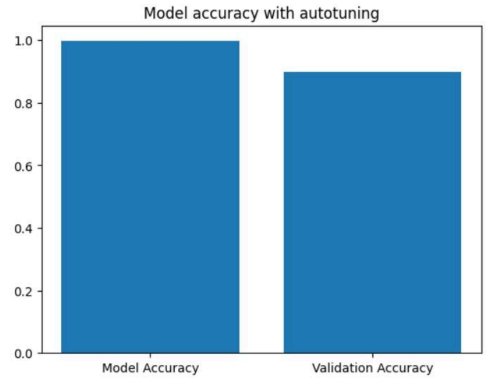
As part of the literature review, the methodologies presented in three of the twelve papers are implemented[25] to identify, contemplate and understand the integral characteristics enforced in NLP techniques in the domain of context-based sentiment analysis.

Convolutional Neural Networks for Sentence Classification[26]: The implementation achieves a test accuracy of 85.33%

```
In [18]: model.load_state_dict(torch.load('tut4-model.pt'))

         test_loss, test_acc = evaluate(model, test_iterator, criterion)

         print(f'Test Loss: {test_loss:.3f} | Test Acc: {test_acc*100:.2f}%')

Test Loss: 0.344 | Test Acc: 85.33%
```

Context-based sentiment analysis on customer reviews using machine learning linear models[27]: The implementation of FastText gives out peak test accuracy of 89.925%

```
plt.title("Model accuracy with autotuning")
plt.bar(labels, accuracy_data)
plt.show()

Result :  (32000, 0.9970625, 0.9970625)
Validation :  (8000, 0.89925, 0.89925)
```




BERT - Bidirectional Encoder Representations from Transformers: The implementation [28]:

```
# Set up the ModelCheckpoint callback to save the best model weights
checkpoint_callback = ModelCheckpoint(
    filepath=checkpoint_path,
    monitor='val_accuracy', # or 'val_loss' depending on which metric you want to monitor
    mode='max', # or 'min' depending on whether you want to maximize or minimize the monitored metric
    save_best_only=True,
    save_weights_only=True,
    verbose=1)

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=3e-5, epsilon=1e-08, clipnorm=1.0),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=[tf.keras.metrics.SparseCategoricalAccuracy('accuracy')])

# Train the model with the callback
history = model.fit(
    train_data,
    epochs=2,
    validation_data=validation_data,
    callbacks=[checkpoint_callback])

Epoch 1/2
   1250/Unknown - 1108s 836ms/step - loss: 0.2678 - accuracy: 0.8838
Epoch 1: val_accuracy improved from -inf to 0.87920, saving model to best_model_weights.h5
1250/1250 [==============================] - 1158s 876ms/step - loss: 0.2678 - accuracy: 0.8838 - val_loss: 0.3052 - val_accuracy: 0.8792
Epoch 2/2
1250/1250 [==============================] - ETA: 0s - loss: 0.0757 - accuracy: 0.9745
Epoch 2: val_accuracy improved from 0.87920 to 0.88160, saving model to best_model_weights.h5
1250/1250 [==============================] - 1076s 860ms/step - loss: 0.0757 - accuracy: 0.9745 - val_loss: 0.4865 - val_accuracy: 0.8816
```

The implementation achieves a test accuracy of 88.16%.

## Conclusion:

In conclusion, context-based sentiment analysis has gained much traction over the years and emerged as a remarkable study field in natural language processing. According to the reviewed literature, researchers have extensively incorporated contextual data into their models to increase the precision and effectiveness of sentiment analysis techniques. The performance of sentiment analysis in various applications, including social media, e-commerce, and customer feedback analysis, has considerably improved since the pre-2010 era owing to the development of deep learning models like CNN, BERT, and pre-training models. The examined literature also emphasizes context-based sentiment analysis's flaws and restrictions, such as the absence of labeled data, lack of domain adaption, and lack of support for several languages. Nevertheless, context-based sentiment analysis has a bright future with the expanding nature of the Information era that has transmogrified into the AI era with enormous leaps of growth in the domain of NLP. As a result, the ever-evolving need for context-based sentiment analysis becomes crucial and imperative.

## References:


1. Mostafa Al Masum Shaikh, Helmut Prendinger, and Mitsuru Ishizuka. 2007. An Analytical Approach to Assess Sentiment of Text.
2. Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales.
3. H. Liu, H. Lieberman, and T. Selker, “A Model of Textual Affect Sensing using Real-World Knowledge”, In Proceedings of the Seventh International Conference on Intelligent User Interfaces, (January 12-15, Miami, FL), ACM,2003, pp. 125-132.
4. Fellbaum, Christiane (2005). WordNet and wordnets. In: Brown, Keith, et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670.
5. Lipika Dey & Sk. Mirajul Haque(2009): Opinion Mining from Noisy Data
6. Customer Review Data Set: http://www.cs.uic.edu/~liub/FBS/sentimentanalysis.html
7. Data Set: http://www.indiacar.com
8. Sun Yueheng, Wang Linmei, Deng Zheng. 2009. Automatic Sentiment Analysis for Web User Reviews.
9. Albert Weichselbraun, Stefan Gindl, and Arno Scharl. 2013. Extracting and Grounding Contextualized Sentiment Lexicons.
10. R. Navigli and M. Lapata, “Graph Connectivity Measures for Unsupervised Word Sense Disambiguation,” Proc. 20th Int’l Joint Conf. Artificial Intelligence (IJCAI), Morgan Kaufmann, 2007, pp. 1683–1688
11. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques.
12. Kim Yoon. 2014. Convolutional Neural Networks for Sentence Classification
13. Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences
14. Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
15. Viraj Parkhe and Bhaskar Biswas. 2014. Aspect-Based Sentiment Analysis of Movie Reviews.
16. Jianxing Yu, Zheng-Jun Zha, MengWang, Tat-Seng Chua, “Aspect Ranking: Identifying Important Product Aspects from Online Consumer Reviews”. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1496–1505, Portland, Oregon, June 19-24, 2011.
17. Stanford Log linear Part-Of-Speech Tagger.
18. Liu, C., Xu, L., Wang, Z., & Wang, N. (2016). Sentiment analysis method based on context view. In 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 1439-1443). IEEE.

19. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding with unsupervised learning, OpenAI.
20. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding
21. Nurul Husna Mahadzir, Nor Hafizah Abdul Razak, Mohd Faizal Mohd Omar (2020). A New Sentiment Analysis Model for Mixed Language using Contextual Lexicon
22. Zhengxuan Wu, Desmond C. Ong. 2020. Context-Guided BERT for Targeted Aspect-Based Sentiment Analysis
23. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J. G.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. In Proceedings of the 57th Annual Meeting of the
24. Anandan Chinnalagu and Ashok Kumar Durairaj (2021). Context-based sentiment analysis on customer reviews using machine learning linear models
25. Literature Survey Review on "Context-Based Sentiment Analysis":(2023) https://github.com/saimadhur99/nlp-literature-review
26. Py Torch Sentiment Analysis. 2021 https://github.com/bentrevett/pytorch-sentiment-analysis
27. fastText Sentiment Analysis: https://www.kaggle.com/code/fauh45/fasttext-sentiment-analysis
28. Sentiment Analysis in 10 Minutes with BERT and TensorFlow: https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671
29. Twitter: https://www.kaggle.com/c/twitter-sentiment-analysis/data
30. IMDB:https://www.kaggle.com/columbine/imdb-dataset-sentiment-analysis-in-csv-format
31. Yelp:https://www.kaggle.com/omkarsabnis/sentiment-analysis-on-the-yelp-reviews-dataset/data
32. Amazon: https://www.kaggle.com/saurav9786/amazon-product-reviews
33. SemEval-2014 Task 4 Dataset: https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools
34. Sentihood Dataset: https://github.com/topics/sentihood-dataset