Contextual Embedding with Cosine Similarity for Aspect-based Sentiment Analysis

Alvian Wijaya, SN: 7695895, aw519@uowmail.edu.au
Tran Phuc An Nguyen, SN: 7068414, tpan980@uowmail.edu.au
Nowmin Naj Manisha, SN: 8248862, nnm031@uowmail.edu.au
Weifeng Wu, SN: 8341631, ww468@uowmail.edu.au

School of Computing and Information Technology
University of Wollongong

CSCI933/CSCI443

May 31, 2024

Mark scored by report and code: /150

Group member	Alvian Wijaya	Tran Phuc An Nguyen	Nowmin Naj Manisha	Weifeng Wu
Contribution marks (out of 10)	10	10	10	10
Individual marks (out of 30)				

Abstract

In this report, we investigate aspect-based sentiment analysis (ABSA) from the IMDB movie review dataset. Our approach systematically identifies and extracts aspect terms from the review by utilizing part-of-speech tagging, BERT embedding, and cosine similarity. We also proposed a classification approach that implements the transformer architecture, we aim to accurately capture the sentiment of each aspect and determine the sentiment polarity of the overall review based on each aspect. Our approach is able to achieve an accuracy of 78%.

1 Introduction

The surge in user-generated online content has heightened the importance of sentiment analysis in understanding human emotions. Product review analysis, stock market prediction, and political analysis are some of the practical use cases of sentiment analysis. Recent studies focus on enhancing the granularity at the aspect level. Aspect-level sentiment analysis commonly known as Aspect-based Sentiment Analysis (ABSA), is a fine-grained form of sentiment analysis. With the rise of deep learning methodologies and an increase in computational power, deep learning offers a solution to achieve these goals to implement ABSA. Deep learning methods offer promise in achieving these goals by capturing both syntactic and semantic features of text without extensive features engineering (Do, Prasad, Maag, & Alsadoon, 2019).

Earlier approaches implement a rule-based approach that was introduced by the author (Poria, Cambria, Ku, Gui, & Gelbukh, 2014). The author utilizes common sense knowledge and sentence dependency trees to detect both implicit and explicit aspects. Drawing insights from recent advancements in common-sense reasoning and concept-level sentiment analysis, the approach leverages external knowledge to extract aspects and determine their associated polarity. Chong, Selvaretnam, and Soon (2014) utilizes Natural Language Processing (NLP) techniques to classify sentiment. Deep learning models such as Convolutional Neural Network (CNN) which is proposed by (Kim, 2014), and transformer-based architecture like BERT which is proposed by (Devlin, Chang, Lee, & Toutanova, 2019) have shown promising results in aspect-based sentiment analysis. These models can learn complex patterns and representations directly from raw text data, eliminating the need for extensive feature engineering.

2 Literature review

This literature review critically evaluates six peer-reviewed papers, examining the strengths and weaknesses of different ABSA methods. By analysing the methodologies, techniques, and experimental results of each approach, this review aims to provide insights into the current landscape of ABSA research and identify potential avenues for further development.

2.1 Attention-based Bidirectional CNN-RNN Deep Model

Basiri, Nemati, Abdar, Cambria, and Acharya (2021) proposed an Attention-based Bidirectional CNN-RNN Deep Model (ABCDM), by utilizing two separate Bidirectional Long Short-Term Memory (BiLSTM) and Gated Recurrent Unit (GRT) layers to capture past and future contexts by incorporating temporal information flow in both directions. An attention mechanism is then applied to assign varying weights to different words in the comment allowing the model to focus on informative words. Following this, convolutional operations are utilized to extract information on local features and reduce dimensionality. Two parallel convolution layers with different kernel sizes are employed independently for the BiLSTM and GRU branches. Maximum and average pooling layers are then applied independently to the outputs of the CNNs to down-sample feature maps. Finally, batch normalization is applied to accelerate network training and reduce over-fitting. A fully connected dense layer transforms the feature vector into a high-level sentiment representation, which is then fed into an output layer for binary classification. ABCDM is evaluated on long review and short Twitter datasets for sentiment analysis.

One of the advantages of this approach is the ability to handle long dependencies introduced by implementing BiLSTM. Furthermore, convolutional operations allow the extraction of local features. On the other hand, due the the increase in complexity of the approach, its computational complexity poses challenges.

2.2 Natural Language Processing Techniques

Chong et al. (2014) focuses on extracting sentiment by using Natural Language Processing (NLP) techniques. The experiment involved extracting tweets from a Twitter database and manually labelling them as positive negative or neutral. Before they conducted the analysis, the author implemented data preprocessing to structure the tweets into a machine-readable format.

The approach that was introduced by Chong et al. (2014) consists of three steps namely, subjectivity classification, semantic association, and polarity classification. Subjectivity classification is used to determine the subjectivity of tweets (categorised into either subjective or objective tweets). If the tweets are categorised as subjective, it goes to the next process of semantic association to identify the sentiment lexicons associated with the subjects. Lastly, the polarity classification classifies tweets based on the sentiment lexicons (e.g., adjective words) by implementing a rule-based approach.

While NLP techniques offer interpretability and flexibility, their reliance on linguistic rules limits their effectiveness in capturing complex contextual details.

2.3 Support Vector Machine

Maulana, Rahayuningsih, Irmayani, Saputra, and Jayanti (2020) explores sentiment analysis of movie reviews using a Support Vector Machine (SVM) and SVM with Information Gain feature selection. The study addresses the challenges of SVM's parameter and feature selection weaknesses, leading to lower classification accuracy. It employs two movie review datasets (i.e., Cornell and Stanford datasets), with a preprocessing stage including tokenization, stopwords removal and stemming. The information Gain technique serves as the feature selection method, with testing conducted using 10-fold cross-validation and evaluation metrics such as confusion matrix and AUC value. The proposed method involves applying Information Gain to improve SVM classifier accuracy. The classification process involves determining whether a sentence belongs to the positive or negative class based on probability calculations. If the probability for the positive class exceeds that of the negative class, the sentence is classified as positive; otherwise, it is classified as negative.

SVM, especially when combined with Information Gain, demonstrates improved accuracy. The main drawback of this approach is due to SVM's performances heavily rely on appropriate feature-engineering and hyper-parameter tuning.

2.4 Aspect-based Sentiment Analysis by using lexicon

Another approach that aims to increase the information extracted from textual data is introduced by Mowlaei, Saniee Abadeh, and Keshavarz (2020). The study proposes extensions of two lexicon generation methods. The first method is based on statistical techniques and the other on a genetic algorithm, both are adapted to better suit aspect-based datasets. One of the methods is the Aspect-Based Frequency Sentiment Analysis (ABFBSA) lexicon generation extends the Frequency Based Sentiment Analysis (FBSA) method.

While FBSA was designed for tweet-level opinion options, ABFBSA modifies it to suit aspect-based problems by considering the nearest aspect label to each word. ABFBSA also incorporates lemmatization, stopword removal, and negation handling to enhance accuracy. The algorithm calculates word scores based on positive and negative frequencies, with normalization to address class imbalance. The other method is Adaptive Lexicon Learning using Genetic Algorithm (ALGA). ALGA was initially designed for tweet-level lexicon learning, which proved unsuitable for aspect-based sentiment analysis (SA) due to the structure of reviews. Preprocessing steps include tokenization, stopword removal, part-of-speech tag filtering, lemmatization, and windowing around aspect terms. This preprocessing ensures that only relevant words are considered for lexicon generation. Aspect-Based Frequency-Based Sentiment Analysis (ABFBSA) and Aspect-Based ALGA (ABALGA) were utilized to merge datasets from benchmark datasets in order to generate lexicons.

While lexicon-based approaches offer simplicity, due to their limited understanding of contextual information from the sentence, they lack the capability to cover detailed sentiment expressions.

2.5 Bidirectional Encoder Representations from Transformers

Hoang, Bihorac, and Rouces (2019) explores the effectiveness of leveraging Bidirectional Encoder Representations from Transformers (BERT)'s contextual word representations and fine-tuning with additional generated text to tackle out-of-domain ABSA. BERT (Devlin et al., 2019) has demonstrated strong performance in NLP tasks.

For tasks like ABSA, BERT performance is further enhanced by a technique called Post-Training, which involves additional training on review text. This approach treats ABSA as a question-answering problem, employing a machine reading comprehension technique known as *review reading comprehension.* By framing ABSA as a sentence-pair classification task and constructing auxiliary sentences, BERT outperforms previous state-of-the-art models that relied on single-sentence classification. The aspect classification model utilizes BERT for sentence pair classification to determine aspect relevance. The sentiment polarity classifier predicts sentiment labels (positive, negative, neutral, or conflict) for given aspect-text pairs. The combined model integrates both aspect and sentiment classification, outputting sentiment if the aspect is related and an unrelated label otherwise. BERT excels in capturing intricate contextual information but requires substantial computational resources for training inference.

2.6 Modelling Context and Syntactical Features

Phan and Ogunbona (2020) proposed an end-to-end ABSA solution that leverages syntactical information, which has been undervalued in previous works. Their approach consists of two processes namely Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). *Contextualized syntax-based aspect extraction (CSAE)* was introduced by the author to tackle the AE process. Traditional models struggle with multi-word aspect boundaries and the syntactical correlation between aspect terms and context. To address these issues, CSAE enhanced the aspect extraction model by combining part-of-speech, dependency-based, and contextualized embedding (e.g., BERT, RoBERTa) and introduced the syntactic relative distance to allow to mitigate the impact of unrelated words in sentiment classification (Phan & Ogunbona, 2020).

Deep learning approaches like ABCDM and BERT outperform traditional methods in accuracy and generalization but require substantial computational resources. Lexicon-based methods offer simplicity and interpretability but may lack coverage and struggle with capturing detailed sentiment expression. SVM, when properly tuned and combined with the feature selection technique, can achieve competitive performance but requires careful parameter selection.

3 Methods

Since ABSA can be decomposed into two sub-tasks namely, aspect extraction (AE) which aims to extract the important aspect information from the sentence, and aspect sentiment classification (ASC) which aims to classify the sentiment polarity, our approach consists of two processes. Figure 1 shows the overall architecture of our approach for ABSA. For our AE, we implemented word type (part-of-speech tag) filtering, BERT embedding, and cosine similarity. Following that, our ASC approach leverages BERT embedding with a transformer encoder to classify the polarity of the sentiment from the given set of aspects.

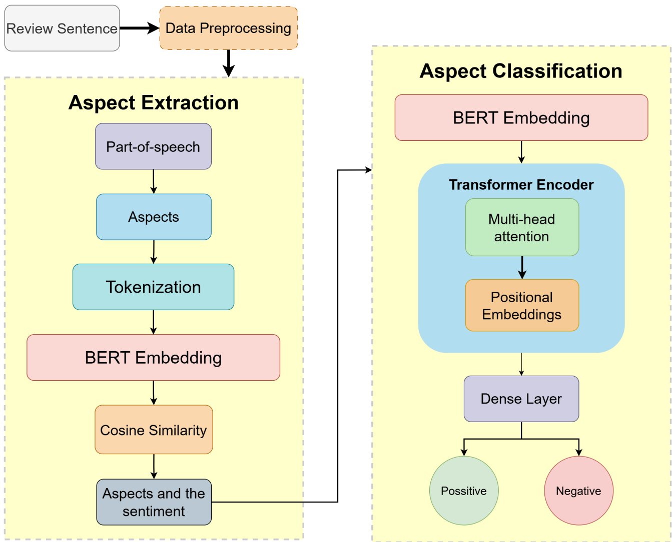

Figure 1: Overall architecture of the proposed method

3.1 Aspect Extraction

The first process of our ABSA approach is the Aspect Extraction (AE). In this process, we aim to gather the necessary aspects with their following terms and remove the unrelated terms from the aspects.

3.1.1 Part-of-Speech Tagging

Part-of-speech tagging (PoS) tagging is used to classify words based on the types. Since PoS can provide enough information from the sentence, we can extract the required aspects.

3.1.2 Tokenization

We implemented the BERT (Devlin et al., 2019) tokenization method. BERT tokenization implemented the *WordPiece* tokenization that was introduced by Wu et al. (2016). The purpose of the tokenization process is to break down words into tokens (can be a full word or sub-words) to help the model process rare words. This process involves converting sentences into BERT tokens which include BERT special tokens. The final result of this tokenization process is in the form of $[CLS] + sentence + [SEP]$. The $[CLS]$ token represents the start of the sentence and the $[SEP]$ token represents the reparation between sentences or represents the end of a sentence.

3.1.3 BERT Embedding

The generated tokens from the previous process will go through a word-embedding process which aims to map the token into vectors that can represent the semantic relationship between tokens. For our word-embedding process, we implemented pre-trained contextualised embedding (i.e., BERT embedding). As mentioned in the paper by Devlin et al. (2019), BERT embeddings implement the sum of token embeddings, segment embeddings, and position embeddings. We chose to implement BERT embedding due to its ability to capture the contextual information of each token. The generated contextual embedding from BERT is in a vector with a dimension of 1×768.

3.1.4 Cosine Similarity

Cosine similarity is used to measure the similarity between aspects extracted from the part-of-speech tagging and other words in the sentence. As mentioned in the paper by Oswald, Simon, and Bhattacharya (2022), cosine similarity is calculated as the formula below:

$$CosSimilarity = \cos\theta = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{n}^{i=1} A_i B_i}{\sqrt{\sum_{n}^{i=1} A_i^2}\sqrt{\sum_{n}^{i=1} B_i^2}} \quad (1)$$

where A is the aspects embedding vector and B is the embedding vector of each word from the whole sentence, both are 1×768 dimension vectors. The output value of the cosine similarity ranges from $[0, 1]$ where 1 represents the high similarity between the two embedding vectors.

3.1.5 Output

The output of our aspect extraction process is in the format of $A_1 + [SEP] + A_2 + [SEP] + ... + A_n$ where A_n represents the aspect terms. The generated output of our AE approach is in the form of a string consisting of the tokenized words that allow us to process the output further in the ASC process.

3.2 Aspect-based Sentiment Classification

Due to the output generated by our AE process being in a string form, we implemented a BERT tokenizer to split each string back into token form. Following that, we recalculate the embedding value of each review to get the new embedding vector.

3.2.1 Transformer Encoder Layer

The generated embeddings from above are passed through the transformer encoder (Vaswani et al., 2023). The Transformer's attention mechanism captures long-range dependencies and contextual relationships, which are crucial for understanding sentiment associated with specific aspects. We implemented the transformer encoder as mentioned in the paper by (Vaswani et al., 2023), therefore, we implemented the positional embeddings to our transformer architecture.

The Multi-Head Attention mechanism allows the model to focus on different parts of the input sequence concurrently. By using multiple self-attention heads, the model learns diverse representations of the input data, capturing various contextual relationships and improving the detection of sentiment associated with specific aspects.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$.

To incorporate the order of the tokens in the input sequence, positional embeddings are added to the transformer encoder layer, enabling the model to understand the sequence structure. As shown in the paper by (Vaswani et al., 2023), below is the function used to generate positional embedding.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right) \tag{3}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right) \tag{4}$$

Where pos is the position and i is the dimension.

3.2.2 Linear Layer

After we process the BERT embedding through the transformer encoder layer, the output of the transformer encoder (with the dimension following the BERT embedding {768}) is connected through the linear (or fully-connected) layer that determines the final classification for each aspect.

4 Experiments

In this report, we aim to experiment and evaluate the performance of our proposed model for aspect-based sentimental analysis. We aim to investigate the impact of the components within the model contribute to overall effectiveness.

4.1 IMDB movie review dataset

The dataset provided by Maas et al. (2011) consists of 50,000 movie reviews from IMDB, categorized as either positive or negative. Each movie has no more than 30 reviews included. A review is considered negative if it has a $rating \leq 4$, and positive if it has a $rating \geq 7$. The dataset does not contain any neutral reviews.

Additionally, an unsupervised dataset of 50,000 reviews without labels is available, but it will not be utilized in our experiments.

Figure 2 illustrates that the training and testing datasets are evenly balanced between positive and negative reviews on average, which guarantees a fair evaluation of the model's performance.

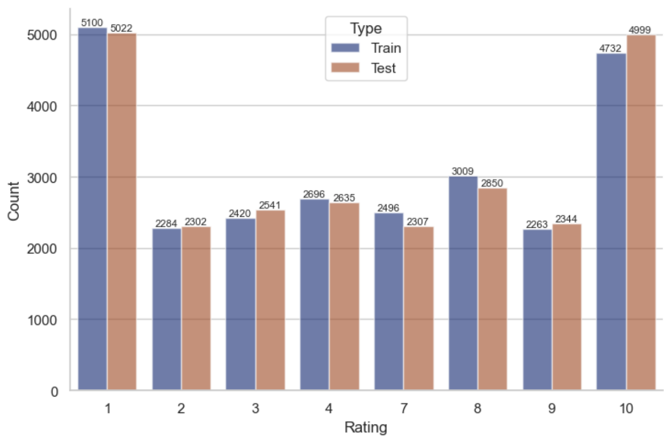

Figure 2: Rating distribution of the dataset

4.2 Experimental setup

4.2.1 Data Loading

Since the dataset provided by Maas et al. (2011) has an equal number of reviews between the positive and negative and also an equal dataset between the training and testing, we use a 50:50 train and test ratio for our experiment. We included 10,000 rows of reviews for our training dataset consisting of 5,000 positive reviews and 5,000 negative reviews. For our testing dataset, we included 10,000 reviews consisting of equal division between the positive and the negative reviews, resulting in a total of 20,000 reviews. We reduce the number of data to be loaded to reduce the time necessary to run our experiments.

4.2.2 Data Pre-processing

Upon reviewing the dataset, it was noted that the reviews contained HTML tags that are not necessary causing unnecessary tokens to be generated during the tokenization process and reducing the performance of our model. Therefore, we implemented a data pre-processing process by eliminating any HTML tags, punctuation, eliminating extra whitespace, and converting text to lowercase.

4.2.3 Aspect Extraction

In aspect extraction, we initially extracted aspects from each review by analyzing the part-of-speech tags of words. As most aspects are typically categorized as *nouns*, we considered *singular nouns, plural nouns, singular proper nouns, and plural proper nouns* in our experiments. We set the *max_content_length* to 256 for the BERT tokenizer.

4.2.4 Aspect Classification

In our sentiment classification process, we utilized a transformer encoder layer with multi-head attention and positional embeddings. This transformer encoder layer was then coupled with a dense layer to convert the high-dimensional representations into a format conducive to sentiment classification. The model was trained using the Adam optimizer, with a learning rate set to $2e-5$ and a batch size of 16, with 3 epochs of training. Furthermore, we set the number of attention-heads to 8, the transformer layer to 2, and the dropout to 0.3.

4.3 Experiment 1

In this experiment, we employed our complete approach for ABSA. We implemented a positional embedding layer and 2 transformer encoder layers.

Research Question 1: How does our approach perform?

4.4 Experiment 2

Following the first experiment, we investigate the impact of positional embedding on our model, since the BERT embedding has incorporated positional embeddings.

Research Question 2: How does an additional layer of positional embedding impact our ABSA model?

4.5 Experiment 3

For our last experiment, we replace the transformer encoder with BiLSTM following the BERT embeddings. This experiment aims to evaluate the importance of multi-head attention offered by the transformer architecture.

Research Question 3: How does removing the attention mechanism affect the performance of our ABSA model?

4.6 Results

We evaluate all three experiments based on the accuracy metric.

Loss is the average loss over all sentences in the dataset.

$$Loss = \frac{\sum_{N}^{s=1} L_s}{N} \tag{5}$$

where N is the number of batches (sentences) in the dataset, L_s is the loss of $s-th$ sentence.

Accuracy is the ratio of the total correct prediction to the total number of samples (present by M) in the dataset.

$$Accuracy = \frac{\sum_{N}^{s=1} \sum_{n_s}^{a=1} (y-pred_{sa} \equiv y_{sa})}{M} \tag{6}$$

where $y-pred_{sa}$ and y_{sa} is the predicted label and true label of the $a-th$ sample of the n_s sentence which is the $s-th$ sentence in the dataset contains N number of batches (sentences).

Table 1: Experiment results

Model	Train		Test	
	Loss	Acc	Loss	Acc
BERT+Pos Embed+Transformer	48.16%	76.66%	45.34%	78.61%
BERT+Transformer	42.38%	80.53%	39.19%	82.17%
BERT+BiLSTM	62.68%	68.19%	59.16%	71.06%

Table 1 summarises the results of our evaluations. The evaluation was conducted from 10,000 rows of reviews. The highest accuracy score was achieved by the BERT + transformer with an accuracy of 82.17% and a loss of 39.19%. On the other hand, our proposed approach (BERT + Pos embeddings + transformers) achieves an accuracy of 78.61%, with a loss of 45.34%. The lowest score was achieved by the BERT + BiLSTM with an accuracy of 71.06% and a loss of 59.16%.

Answer to RQ1: Our proposed model can provide a respectable accuracy.

In the following two experiments, we aim to study the effect of each component on the overall performance of our approach, we conducted the experiments by removing components in the ASC process.

In the second experiment, we implement our model without the positional embedding layer. The experimental results showed a notable decrease in the loss rate and a significant increase in accuracy during both the training and testing phases compared to the first experiment. In the training phase, the accuracy increased by 3.87%, from 76.66% to 80.53%, while the loss decreased by 5.78%, from 48.16% to 42.38%. In the testing phase, the accuracy increased by 3.56%, from 78.61% to 82.17%, while the loss decreased by 6.15%, from 45.34% to 39.19%.

The results of the second experiment indicate that:

Answer to RQ2: The absence of positional embeddings did not significantly reduce the model's performance; on the contrary, it enhanced the overall performance by 03.56% in accuracy. Since BERT already incorporates positional information within its embeddings, adding a layer of positional embeddings appears to be redundant and detracts performance. Thus, multi-head attention remains crucial for capturing complex dependencies in the data.

In the last experiment, we replace the transformer encoder layer with a BiLSTM network following the BERT embeddings. The results not only increase the loss rate but also heavily reduce the accuracy. In the training phase, the accuracy decreased by 12.34%, from 80.53% to 68.19%, while the loss increased by 2.03%, from 42.38% to 62.68%. In the testing phase, the accuracy decreased by 11.11%, from 82.17% to 71.06%, while the loss increased by 19.97%, from 39.19% to 59.16%.

The results of the third experiment indicate that:

Answer to RQ3: Removing the attention mechanism heavily reduced the performance of the model, decreased the accuracy by 11.11% and increased the loss rate by 19.97%

5 Discussion

In this report, the experiments showcase that the BERT+Transformer model, particularly without additional positional embeddings, offers the best performance for ABSA tasks on the IMDB dataset. We observe that BERT embeddings provide positional embeddings that capture the position of each word and do not require an additional positional embedding layer.

Moreover, the comparative analysis of the three experiments underscores the critical role of transformer encoders and multi-head attention mechanisms in enhancing model performance for ABSA tasks. The BiLSTM network is less efficient compared to the transformer encoder layer. While Bidirectional LSTM generally offer improvements over unidirectional ones, they still fall short of the performance achieved by transformer architectures. This result aligns with recent trends in NLP research, where transformer-based models have consistently outperformed RNN-based models in various tasks.

6 Conclusion

In this study, we investigated aspect-based sentiment analysis (ABSA) using the IMDB movie review dataset, focusing on identifying and extracting aspect terms through part-of-speech tagging, BERT embeddings, and cosine similarity. Our experiments demonstrated that the combination of BERT embeddings and transformer encoders yielded the highest accuracy, outperforming other models such as BERT with BiLSTM. Notably, the inclusion of additional positional embeddings was found to be redundant, as BERT embeddings already capture positional information effectively. Our proposed approach achieved an accuracy of 78.61%. These findings highlight the significance of transformer architecture and multi-head attention mechanisms in improving the performance of ABSA models. Future work could explore further optimization of model parameters and the integration of additional contextual information to enhance ABSA accuracy.

References

Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, *115*, 279-294. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167739X20309195` doi: https://doi.org/10.1016/j.future.2020.08.005

Chong, W. Y., Selvaretnam, B., & Soon, L.-K. (2014). Natural language processing for sentiment analysis: An exploratory analysis on tweets. In *2014 4th international conference on artificial intelligence with applications in engineering and technology* (p. 212-217). doi: 10.1109/ICAIET.2014.43

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding.*

Do, H. H., Prasad, P., Maag, A., & Alsadoon, A. (2019). Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Systems with Applications*, *118*, 272-299. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0957417418306456` doi: https://doi.org/10.1016/j.eswa.2018.10.003

Hoang, M., Bihorac, O. A., & Rouces, J. (2019, 09-10). Aspect-based sentiment analysis using BERT. In M. Hartmann & B. Plank (Eds.), *Proceedings of the 22nd nordic conference on computational linguistics* (pp. 187–196). Turku, Finland: Linköping University Electronic Press. Retrieved from https://aclanthology.org/W19-6120

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, *abs/1408.5882*. Retrieved from http://arxiv.org/abs/1408.5882

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. In D. Lin, Y. Matsumoto, & R. Mihalcea (Eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 142–150). Association for Computational Linguistics. Retrieved 2024-05-18, from https://aclanthology.org/P11-1015

Maulana, R., Rahayuningsih, P. A., Irmayani, W., Saputra, D., & Jayanti, W. E. (2020). Improved accuracy of sentiment analysis movie review using support vector machine based information gain. In *Journal of physics: Conference series* (Vol. 1641, p. 012060).

Mowlaei, M. E., Saniee Abadeh, M., & Keshavarz, H. (2020). Aspect-based sentiment analysis using adaptive aspect-based lexicons. *Expert Systems with Applications*, *148*, 113234. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417420300609 doi: https://doi.org/10.1016/j.eswa.2020.113234

Oswald, C., Simon, S. E., & Bhattacharya, A. (2022, sep). Spotspam: Intention analysis–driven sms spam detection using bert embeddings. *ACM Trans. Web*, *16*(3). Retrieved from https://doi.org/10.1145/3538491 doi: 10.1145/3538491

Phan, M. H., & Ogunbona, P. O. (2020). Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 3211–3220).

Poria, S., Cambria, E., Ku, L.-W., Gui, C., & Gelbukh, A. (2014). A rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media (socialnlp)* (pp. 28–37).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2023). *Attention is all you need.*

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... Dean, J. (2016, 10). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.* arXiv. Retrieved from http://arxiv.org/abs/1609.08144 (arXiv:1609.08144 [cs]) doi: 10.48550/arXiv.1609.08144