Journal of Data Science 24 (2), 319–337
April 2026
DOI: 10.6339/26-JDS1225
Statistical Data Science

Interpretable Word-Level Context-Based Sentiment Analysis

Chenyu Yang[1], Eric Larson[2], and Jing Cao[1,*]

[1]*Department of Statistics and Data Science, Southern Methodist University, Dallas, Texas, U.S.A*
[2]*Department of Computer Science, Southern Methodist University, Dallas, Texas, U.S.A*

Abstract

We propose a fine-grained attention-based multiple instance classification (FAMIC) model for interpretable word-level sentiment analysis (SA) using only document-level sentiment labels. By operating at the word level, FAMIC enhances interpretability while maintaining competitive performance in document-level classification. The model generates interpretable outputs such as contextual weighting, word neutrality, and negation cues, offering insights into how context shapes sentiment and how the model arrives at its predictions. FAMIC is built on a straightforward yet effective architecture that combines a multiple instance classification framework with self-attention and positionally encoded self-attention blocks. This design enables the model to capture both local and global contextual dependencies, supporting nuanced sentiment interpretation. We evaluate FAMIC on two sentiment classification datasets and provide an extensive analysis of its interpretability and performance.

Keywords *interpretable sentiment analysis; multiple instance classification; relative positional embedding; self-attention*

1 Introduction

Sentiment analysis (SA) is a key task in natural language processing (NLP) that aims to identify the underlying opinions or emotions expressed in text (Liu, 2012; Fang and Zhan, 2015). In this study, we introduce a transparent and lightweight SA architecture designed to address the black-box nature of transformer-based language models (Singh et al., 2023; Balderas et al., 2023). Our approach enhances interpretability and provides deeper insight into the sentiment decision-making process, which is often opaque in large-scale models. Specifically, the proposed model produces word-level sentiment outputs that are encouraged to learn interpretable linguistic properties such as negation and word neutrality, all within the context of the entire document. This design enables a more explainable sentiment analysis framework while maintaining competitive performance.

Before the recent popularity of large language models, SA was primarily conducted using statistical methods, which employ rigorous probability-based approaches to modeling text data (Medhat et al., 2014). For example, Tyagi and Sharma (2018) used logistic regression to detect hate speech in tweets. Naive Bayes models and support vector machines have also been used to classify movie reviews as positive or negative (Das and Chen, 2007; Pang et al., 2002). These relatively simple statistical models are often considered interpretable and straightforward, as they assume clear relationships between the input text and the outcome, allowing for transparent

*Corresponding author. Email: jcao@smu.edu.

Received July 31, 2025; Accepted February 26, 2026

explanations of model predictions. However, most statistical SA models lack an explicit mechanism for incorporating document context effectively. As a result, words or phrases that carry different sentiments or meanings depending on context often lead to unreliable performance.

Deep neural networks in recent years have gained popularity due to their flexible modeling capability. One of the key advantages is their ability to capture document context, where the context of a word or sentence is dependent on surrounding words, which helps to elucidate its meaning (Mikolov et al., 2013; Zhang and Wallace, 2017). For example, Kim (2014) first used CNNs for sentiment classification of movie reviews and showed that CNNs can effectively capture local contextual patterns. The long-short term memory (LSTM) model (Hochreiter and Schmidhuber, 1997) and its variant, the Bidirectional LSTM (BiLSTM) model (Zhang et al., 2015), have been successful in SA, especially when combined with CNNs (Minaee et al., 2019). One of the most successful transformer models is the Bidirectional Encoder Representations from Transformers (BERT) model, proposed by Devlin et al. (2019). With the self-attention mechanism, it can generate context-aware representations. Numerous works have developed transfer learning approaches from BERT that perform state-of-the-art in SA (Singh et al., 2021; Zhao and Yu, 2021; Wu and Ong, 2021).

Although language models have demonstrated their effectiveness in SA, they are often criticized as being "black box" due to their complex structures and difficult interpretability. Especially for methods which focus on sentence-level or document-level analysis, the reasoning behind the final classification/prediction is often unclear. This lack of interpretability can undermine trust in the results and prevent applications of the models in areas where interpretability is as important as predictive accuracy (Petch et al., 2022). Therefore, there is a growing need for interpretable high-performance SA models that can provide greater transparency in the inference processes.

Recent work on interpretable sentiment analysis has largely focused on post-hoc explanation frameworks applied to high-capacity neural models, including CNN-based Class Activation Mapping (CAM), self-attention visualization, attention rollout, and model-agnostic attribution methods such as LIME and SHAP (Zhou et al., 2015; Ribeiro et al., 2016; Lundberg and Lee, 2017; Abnar and Zuidema, 2020). While these approaches offer insight into which input tokens correlate with a model's prediction, they do not provide intrinsic interpretability, as the explanatory mechanism is not part of the model's learning objective (Rudin, 2019). Moreover, attribution scores produced by these methods are often relative, model-dependent, and unstable under small input perturbations, limiting their reliability for fine-grained sentiment reasoning (Kindermans et al., 2017; Jain and Wallace, 2019). As a result, these techniques tend to answer where a model attends, rather than how sentiment is composed or quantified.

A further limitation shared by many interpretable transformer-based approaches is their reliance on subword-level tokenization, which fragments semantically meaningful units and complicates word-level sentiment interpretation (Devlin et al., 2019; Wu et al., 2016). Subword attributions do not naturally aggregate into coherent sentiment units, making it difficult to compare sentiment strength across words or documents in a consistent and human-interpretable manner (Thogesan et al., 2025). These limitations motivate the development of intrinsically interpretable sentiment models that operate directly at the word level, producing explicit word-level sentiment scores that naturally compose into a document-level sentiment score. Such formulations enable transparent sentiment aggregation, facilitate direct comparison of sentiment intensity across words and texts, and avoid the ambiguity inherent in post-hoc attribution methods.

One approach to interpretable sentiment analysis (SA) is to adopt a word-level context-based sentiment analysis (WCSA) framework, which assigns sentiment scores to individual words

while explicitly modeling their contextual dependencies. By capturing how word-level sentiment emerges from surrounding context, the approach enables a more nuanced and transparent understanding of which words or phrases drive the overall sentiment of a text and to what extent. Despite this potential, most existing SA models operate primarily at the sentence or document level, largely due to the inherent difficulty of assigning sentiment to individual words whose polarity and intensity are often ambiguous and highly context-dependent.

In our previous work (Yang and Cao, 2025), we introduced a WCSA model built on a multiple instance classification (MIC) framework combined with a self-attention mechanism, resulting in a transparent yet effective architecture. While MIC offers interpretability in the inference process and self-attention captures contextual dependencies, the model lacks the ability to explicitly distinguish between local and global context. This limits its capacity to handle finer linguistic phenomena such as negation, sentiment intensity, and word neutrality, which can lead to incorrect results on sentiment analysis. For example, this method can not distinguish the sentiment in the following two sentences with opposite sentiment, where the only difference is the placement of the word "not":

Sentence I: The service of the restaurant is good, the overall experience is **not** bad.

Sentence II: The service of the restaurant is **not** good, the overall experience is bad.

To address these limitations, we propose a new interpretable SA framework called the Fine-grained Attention-based Multiple Instance Classification model (FAMIC). FAMIC explicitly models both global and local contextual dependencies, enabling more accurate and interpretable handling of complex linguistic features, while retaining the benefits of our earlier WCSA architecture. As illustration in Tables 3 and 4, FAMIC can correctly distinguish the different sentiments in Sentences I and II. We have provided a number of other examples demonstrating FAMIC's abilities in handling linguistic features that will aid correct interpretation of SA. Because FAMIC has addressed a non-trivial issue, it makes a meaningful contribution to the interpretable SA research field.

The remainder of the paper is structured as follows. Section 2 provides background information on the model components, i.e., the multiple instance classification framework and the self-attention mechanism with relative position representations. The proposed SA model and its algorithm are detailed in Section 3. Section 4 compares FAMIC to a number of commonly used SA methods on two sentiment classification datasets and provides an extensive analysis of its interpretability and performance. Finally, Section 5 offers a discussion of the findings and concludes the paper.

2 Model Components

2.1 Multiple Instance Classification

Multiple instance learning (MIL) is a form of weakly supervised learning where the classification (MIC) or prediction (MIP) task is performed on a set of labeled bags, each containing a collection of instances whose labels are often unobserved. Each individual instance is described by a set of covariates (or features). Instances in a bag contribute to the observed bag-level response (or label). MIL was first introduced by Dietterich et al. (1997) for drug activity prediction. The bag label is positive if at least one instance label is positive, and the bag label is negative if all instance labels are negative. The goal is to predict the label of a new bag. More details of MIL can be found in Carbonneau et al. (2018).

Ray and Page (2001) presented an approach based on primary instance, which assumes that the bag label is solely determined by the primary instances, while the non-primary instances carry little information on the bag label. Xiong et al. (2024) followed this assumption and introduced a Bayesian MIC approach for cancer detection using T-cell receptor sequences. It is composed of two nested probit regression models, where the inner model predicts the primary instances and the outer model predicts bag labels based on the features of the primary instances identified by the inner model.

Note that the task of predicting the sentiment in text documents can be formulated as an MIC problem. Each text can be considered as a bag consisting of individual words as instances, where the features of these instances are represented by the corresponding word embeddings. Note that word embedding is a commonly used technique to represent text as data, where it maps a word to a latent word representation vector space where words with similar contexts are in proximity (Mikolov et al., 2013). Thus, predicting the overall sentiment of text documents is equivalent to predicting the bag labels.

Specifically, we assume that words in text can be categorized either as sentiment words or as function words. Sentiment words are associated with a clear sentiment polarity, describing an emotion or experience that is either pleasant/desirable or unpleasant/undesirable. On the other hand, function words are words that are used to structure the sentence and convey meaning without sentiment implications, such as most prepositions, conjunctions, articles, pronouns, auxiliary verbs, etc. With the MIC modeling framework, FAMIC is able to recognize sentiment words, estimate sentiment score at the word level, determine the overall sentiment in a document by combining the sentiment scores of individual words, and provide interpretable results in SA.

2.2 Self-Attention with Relative Position Representations

The self-attention mechanism when incorporated in SA enables the model to assess the relationships between words in text. The connections and relationships between words are commonly referred to as dependencies (Nivre, 2005). Such dependencies can be categorized into two types: global dependency and local dependency. Global dependency refers to the long-range relationships between words across the entire text. Local dependency focuses on the immediate relationships between neighboring words within a small window of text, which includes dependencies such as the impact of negation words on sentiment. Capturing local dependencies can help a SA model to effectively utilize the subtle nuances and relationships among words that collectively contribute to the overall sentiment expressed in text.

The self-attention mechanism was originally designed to emphasize the relationships between words throughout the entire sentence, regardless of their specific positions. This characteristic makes it highly proficient in capturing global dependencies. However, its ignorance to the positions of words makes it incompetent in incorporating local dependencies. Without the knowledge of neighboring words, the self-attention mechanism, on its own, struggles to accurately discern sentiment shifts influenced by factors such as negation, word order, or proximity to other words. As shown earlier, self-attention, which is adopted by our previous work (Yang and Cao, 2025), can not distinguish the sentiment in the Sentences I and II with opposite sentiment, where the only difference is the placement of the word “not”.

It provide a clear illustration of why positional understanding is crucial for sentiment analysis: although both sentences contain the same tokens, the polarity reverses because the negation appears in different positions and thus modifies different sentiment-bearing words (“bad” versus “good”). Our previous model would assign exactly the same document-level sentiment score

(e.g., 5.4) to the two sentences above. This occurs because the model effectively treats the input as a bag of word without explicit positional information, so the same set of tokens yields the same sentiment score, regardless of ordering and local dependencies.

To incorporate positional information into self-attention, researchers have proposed different approaches (Shaw et al., 2018; Bilan and Roth, 2018; Chen et al., 2021). The method, introduced by Shaw et al. (2018), is known as self-attention with relative positional representations. The approach explicitly captures the relative positional information of words in a text by introducing relative positional embeddings, which encode both the position and direction between words. To incorporate the relative positional information in self-attention, two separate sets of relative positional embeddings are learned in the Key and Value vectors in the self-attention mechanism (Vaswani et al., 2017), respectively. Then the updated Key and Value vectors with positional awareness are calculated as the original self-attention Key and Value vectors plus their respective relative positional embeddings. The incorporation of relative positional information enables self-attention to consider the positional relationships between words and incorporate them in the computation of attention weights.

3 Approach

The FAMIC architecture is designed with word level terms that each have specific interpretable behaviors such as 1) identifying whether a word in a text is a sentiment word or a functional word (i.e., the neutrality of the word), 2) computing the context-independent sentiment for each word, which represents a word's intrinsic context-free sentiment, 3) computing the contextual global dependency and local dependency for each word respectively, which helps to elucidate how context affects sentiment in different perspective, 4) aggregating the word-level sentiment to produce text-level sentiment. FAMIC also incorporates a modified self-attention with relative positional representations, enabling the model to effectively handle nuanced linguistic complexities, such as negation.

We introduce FAMIC in the case of binary classification, with a note that it can be easily extended to multiclass classification or prediction with continuous outcomes. Figure 1 presents the FAMIC architecture, where y_i $(i = 1, 2, \ldots, n)$ represents the observed sentiment label of the ith document, x_{ij} is a length-d word embedding vector of the jth $(j = 1, 2, \ldots, m_i)$ word in the ith document. Note that only x_{ij} and y_i are observed values and they are represented with a square box in Figure 1.

The structure of the FAMIC model is designed to encourage the following learned behavior: v_{ij} is a real-valued scalar representing the context-independent sentiment score of word j in document i, with a positive value indicating positive sentiment and a negative value negative sentiment. Next r^s_{ij}, r^g_{ij}, and r^l_{ij} are d-dimensional vectors, where r^s_{ij} learns if word j in document i is a sentiment or functional word, r^g_{ij} learns global contextual dependency, and r^l_{ij} learns local contextual dependency. Then δ^s_{ij}, δ^g_{ij}, and δ^l_{ij} are real-valued scalars derived from r^s_{ij}, r^g_{ij}, and r^l_{ij}, respectively. δ^s_{ij} is the proxy indicator that takes the value of 1 or 0, where 1 indicates that the word is a sentiment word and 0 a functional word. δ^g_{ij} and δ^l_{ij} are used as global and local sentiment shifters, respectively. The value of δ^g_{ij} ranges from 0 to 10, quantifying the influence of global contextual dependency for the sentiment of the word. δ^l_{ij} represents the degree of local contextual dependency on the sentiment of the word. It takes a value between -1 and 1, where a negative value indicates the negation of the context-independent sentiment of the word in the document. Thus, FAMIC explicitly models both global and local contextual dependencies in a parallel way.

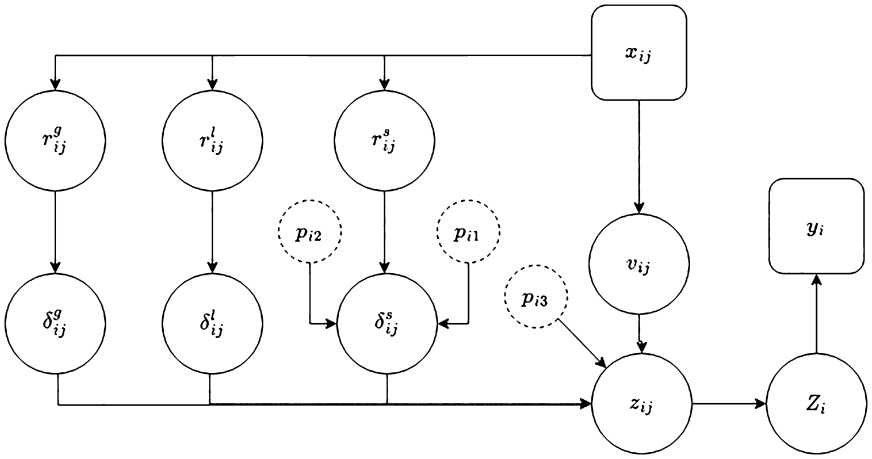

Figure 1: FAMIC architecture.

Finally, p_{i1}, p_{i2}, and p_{i3} are the penalty terms, z_{ij} is the word's context-dependent sentiment, and Z_i represents the overall sentiment score of document i. The remainder of this section provides a detailed description of the architecture and training procedure of the FAMIC model.

We first use a feedforward operation on x_{ij} to calculate v_{ij}, where x_{ij} is a pre-trained word embedding. Thus v_{ij} is context-independent, as it is not influenced by other words in the document. Next we update r^g_{ij} and r^s_{ij}. Since r^g_{ij} contains the word representations on global dependency and r^s_{ij} on the identification of sentiment words, they are under the influence of document context but not affected by the order of words. Consequently, they are updated by the self-attention mechanism.

Recall that r^l_{ij} represents local contextual dependency which is based on the relationships between neighboring words in a document. The self-attention with relative positional representations based on Shaw et al. (2018) follows:

$$e_{ijk} = (x^Q_{ij})(x^K_{ik} + a^K_{j\to k})^T, \qquad \alpha_{ijk} = \frac{\exp(e_{ijk})}{\sum_{h=1}^{m_i}\exp(e_{ijh})}, \qquad r^l_{ij} = \sum_{k=1}^{m_i}\alpha_{ijk}(x^V_{ik} + a^V_{j\to k}), \tag{1}$$

where the superscript Q, K, V represent the Query, Key, and Value vectors, respectively; $a^K_{j\to k}$ and $a^V_{j\to k}$ are the relative positional embeddings for the Key and Value vectors, respectively; α_{ijk} is the attention weight incorporating relative positional representations. We propose a novel variation of the algorithm, where e_{ijk} and α_{ijk} remain the same, but r^l_{ij} becomes:

$$r^l_{ij} = \sum_{k=1}^{m_i}\alpha_{ijk}(a^V_{j\to k}). \tag{2}$$

In Equation (1) r^l_{ij} is a weighted representation of the Value vector modified by the positional information. The Value vector is commonly regarded as encodings of the semantic meaning of words in the input sequence. They capture the fundamental semantic information that plays a crucial role in comprehending the entire sequence, thus proving valuable for various downstream tasks like sentiment analysis (Tabinda Kokab et al., 2022), machine translation (Ghader and Monz, 2017), and question-answering (Seonwoo et al., 2020). Following Equation (2), r^l_{ij} now is a weighted representation of the positional embedding. This adjustment allows r^l_{ij} to focus

on the positional information rather than encoding the semantic meaning carried by the Value vectors. By removing the Value vectors, the model becomes more sensitive to the relative positions of words, enabling it to better capture the local contextual dependencies based on their positions in the sequence. Note that FAMIC allows r^l_{ij} to only use the positional embedding. It is because FAMIC has decomposed the semantic/sentiment information of words in multiple latent dimensions: context-independent score and context-dependent score, where the transition from the former to the latter is further explained by the joint function of global shifter and local shifter, so that r^l_{ij} does not need a comprehensive word representation but a specific word representation focusing on local positional dependency. In addition, by removing the Value vectors in Equation (1), the proposed algorithm yields a more efficient model that requires roughly 30% fewer parameters compared to (Shaw et al., 2018).

In Equation (1) r^l_{ij} is a weighted representation of the Value vector modified by the positional information. The Value vector is commonly regarded as encodings of the semantic meaning of words in the input sequence and is therefore useful for downstream tasks such as sentiment analysis (Tabinda Kokab et al., 2022), machine translation (Ghader and Monz, 2017), and question-answering (Seonwoo et al., 2020). However, in FAMIC, semantic and sentiment content is already explicitly modeled by the parallel components v_{ij}, r^s_{ij}, and r^g_{ij}: v_{ij} provides a context-independent sentiment score derived solely from x_{ij}, r^s_{ij} produces δ^s_{ij} to identify sentiment versus functional words, and r^g_{ij} produces the global shifter δ^g_{ij} to quantify document-level contextual influence. In contrast, the purpose of the local component is to produce δ^l_{ij}, which captures *local* contextual dependency such as whether the context-independent sentiment v_{ij} should be inverted (e.g., negation) or modulated based on nearby words. Therefore, the role of r^l_{ij} in FAMIC is not to re-encode semantics (which are already captured by x_{ij}, v_{ij}, r^s_{ij}, and r^g_{ij}), but to encode *positional relationships* that govern how local context should shift v_{ij} when forming the context-dependent sentiment z_{ij}.

Motivated by this decomposition, we propose the variation in Equation (2) to compute r^l_{ij} solely from the relative positional embeddings $a^V_{j\to k}$ while keeping e_{ijk} and α_{ijk} unchanged. This design encourages the local sentiment shifter to focus on where local context matters (through relative distance and direction) rather than duplicating semantic encoding via x^V_{ik}. In particular, when δ^l_{ij} is intended to take values in $[-1, 1]$ and represent local polarity reversal, anchoring r^l_{ij} to positional information provides a direct inductive bias toward modeling local phenomena such as negation scope and proximity effects. Concretely, in a standard relative self-attention layer, the local shifter would learn three projection matrices for Query, Key, and Value, whereas our variation removes the Value projection entirely while leaving the computation of e_{ijk} and α_{ijk} unchanged. As a result, the local sentiment shifter retains its ability to form position-aware attention weights but avoids the additional parameters and computation associated with transforming x_{ik} into x^V_{ik}, leading to an approximate one-third reduction in projection parameters (conservatively reported as roughly 30% in this work) compared to (Shaw et al., 2018).

Next, δ^s_{ij}, δ^g_{ij}, and δ^l_{ij} are updated as follows, where σ is the sigmoid activation:

$$\delta^s_{ij} = \sigma({r^s_{ij}}^T b^s), \qquad \delta^g_{ij} = 10 \times \sigma({r^g_{ij}}^T b^g), \qquad \delta^l_{ij} = \tanh({r^l_{ij}}^T b^l). \tag{3}$$

Note that the result from the continuous activation function of δ^s_{ij} will produce an 0/1 indicator value with the constraints added by the penalty terms (explained later). A scaling factor of 10 is applied to the sigmoid function for the global sentiment shifter δ^g_{ij} to address the potential vanishing gradient problem. The hyperbolic tangent function is used to update the local sentiment shifter δ^l_{ij} which has an output range between -1 and 1.

The context-dependent sentiment score z_{ij} is obtained by multiplying together the context-independent sentiment v_{ij}, the sentiment word indicator δ^s_{ij}, the global sentiment shifter δ^g_{ij}, and the local sentiment shifter δ^l_{ij}:

$$z_{ij} = v_{ij} \times \delta^s_{ij} \times \delta^g_{ij} \times \delta^l_{ij}. \tag{4}$$

The product of these factors captures their collective impact on the final sentiment score for the word. If a word is identified as a function word (i.e., $\delta^s_{ij} = 0$), then $z_{ij} = 0$, which means that it does not contribute to the document-level sentiment score Z_i. For a sentiment word (i.e., $r^s_{ij} = 1$), the context-dependent score z_{ij} stems from the context-independent score v_{ij} modified by its global contextual dependency and local contextual dependency. The document-level sentiment score Z_i is then determined by averaging the sentiment scores in the document:

$$Z_i = \frac{\sum_{j=1}^{m_i} z_{ij}}{\sum_{j=1}^{m_i} \delta^s_{ij}}, \qquad \hat{y}_i = \mathbf{1}_{[0.5,1]}\left(\sigma(Z_i)\right), \tag{5}$$

where $\hat{y}_i$ denotes the predicted sentiment label. $\hat{y}_i = 1$ if $\sigma(Z_i) \geqslant 0.5$ and $\hat{y}_i = 0$ otherwise.

The penalty terms are constructed as follows,

$$p_{i1} = c_1 \sum_{j=1}^{m_i} \sqrt{\delta^s_{ij}(1 - \delta^s_{ij})}, \qquad p_{i2} = c_2 \sum_{j=1}^{m_i} \delta^s_{ij}, \qquad p_{i3} = c_3 \sqrt{\sum_{j=1}^{m_i} (v_{ij} \times \delta^g_{ij} \times \delta^l_{ij})^2}. \tag{6}$$

Note that an 0/1 indicator function is not differentiable, so we can not use an exact 0/1 indicator for sentiment word identification when using backpropagation. Instead we employ a proxy indicator, δ^s_{ij}, which is a differentiable sigmoid function, and we add the penalty term p_{i1} to ensure that δ^s_{ij}, after rounding to a certain decimal place, has a dichotomous outcome to adequately approximate an 0/1 indicator function. The function in penalty p_{1i} has a dome shaped curve, encouraging δ^s_{ij} to take values close to 0 or 1.

The second penalty term p_{2i} (an L1-norm) promotes sparsity in the identification of sentiment words. Our preliminary examination of the application datasets shows that the sentiment words typically account for less than 30% of all the words in a document. This observation initiates the introduction of sparsity in sentiment word identification. The third penalty term p_{3i} is to ensure the stability in the estimation of z_{ij} in Equation (4). Specifically, it imposes an L2 penalty to prevent the model from arbitrarily inflating the magnitude of $v_{ij} \times \delta^g_{ij} \times \delta^l_{ij}$ in situations where δ^s_{ij} may take close-to-zero values in the early training stage. c_1, c_2, and c_3 are tuning parameters in the penalty terms.

The parameters in FAMIC are trained using gradient descent to minimize the binary cross-entropy loss and the three penalty terms:

$$l(X_i, y_i) = -\frac{1}{n} \sum_{i=1}^{n} \left(\left[y_i \log(\sigma(Z_i)) + (1 - y_i) \log(1 - \sigma(Z_i)) \right] + p_{i1} + p_{i2} + p_{i3} \right), \tag{7}$$

where the cross-entropy loss encourages the model to produce accurate document-level sentiment prediction. The training scheme for FAMIC is presented in Algorithm 1. Additional training details are provided in Appendix A.

Algorithm 1: FAMIC training procedure.

Data: x_{ij}, y_i, where $i = 1, \ldots, n$, and $j = 1, \ldots, m_i$
Initialization: θ_v, θ_l, θ_g, θ_s
First Pass (with $z = v$):
while *not converged* **do**
  Draw random mini batch from data
  for all(x_{ij}, y_i) *in batch* **do**
    Evaluate the objective function, $l_{ce}(x_{ij}, y_i)$
    $\theta_v \leftarrow \text{ADAM}(\nabla_{\theta_v}, l_{ce}(x_{ij}, y_i), \theta_v)$
  end
end
Second Pass (with $z = v \times \delta^s \times \delta^g \times \delta^l$);
set $q = 3$; $c_1 = 1e-4$; $c_2 = 1e-3$; $c_3 = 1e-4$;
while *not converged* **do**
  Draw random mini batch from data
  for all(x_{ij}, y_i) *in batch* **do**
    Calculate δ^g_{ij}, δ^l_{ij}, δ^s_{ij}
    Evaluate the objective function, $l_{ce}(x_{ij}, y_i)$ and penalties, p_{i1}, p_{i2}, p_{i3}
    $\theta_l \leftarrow \text{ADAM}(\nabla_{\theta_l}, l_{ce}(x_{ij}, y_i) + p_{i3}, \theta_l)$
    $\theta_g \leftarrow \text{ADAM}(\nabla_{\theta_g}, l_{ce}(x_{ij}, y_i) + p_{i3}, \theta_g)$
    $\theta_s \leftarrow \text{ADAM}(\nabla_{\theta_s}, l_{ce}(x_{ij}, y_i) + p_{i1} + p_{i2}, \theta_s)$
  end
end

4 Results

We have evaluated FAMIC on two datasets: a wine review dataset (Katumullage et al., 2022) and a Twitter Sentiment140 dataset (Go et al., 2009). The wine review dataset consists of 141,409 reviews collected from the website of the renowned wine magazine *Wine Spectator* dated from 2005 to 2016. Each year, the magazine's editors chose more than 15,000 wines for blind tasting, where they provided tasting notes, numeric ratings, and recommendations. The tasting scores are on a 100-point scale. The majority of wines have a rating in the range of 80–100. For demonstration purposes, we labeled the sentiment of a wine as positive if its rating is at least 90, and negative otherwise.

The Sentiment140 dataset consists of 1.6 million tweets with brief messages each limited to a maximum of 140 characters. The tweets collected were posted between April 6 and June 25 in 2009. Manually labeling such a large dataset would be impractical due to its size. To overcome this challenge, Go et al. (2009) adopted a technique introduced in (Read, 2005), where emoticons were utilized as sentiment labels. Out of the 1.6 million tweets, 800,000 were associated with a negative sentiment and the other 800,000 with a positive sentiment.

4.1 Document Level Performance Evaluation

For the wine dataset, we choose to use the 300-dimensional word embeddings (Glove-300-Wiki) trained on Wikipedia as the embeddings of x_{ij}. Glove-300-Wiki is considered to be a reliable word embedding choice for texts using formal language because Wikipedia mainly consists of

Table 1: FAMIC sentiment classification performance on wine and Twitter datasets.

Wine					
Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	# of Parameters
Naïve Bayes	85.53	85.78	80.31	82.01	<60k
Logistic Regression	87.50	84.02	78.77	81.31	<60k
CNN	88.02	83.38	83.38	83.38	<870k
BiLSTM	88.69	81.91	85.08	83.47	<100k
FAMIC	88.99	84.74	82.83	83.77	<1M
BERT	89.12	82.90	86.58	84.70	110M
Twitter					
Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	# of Parameters
Naïve Bayes	77.45	77.46	77.45	77.45	<60k
Logistic Regression	78.40	77.53	79.80	78.65	<60k
CNN	79.33	79.48	79.00	79.24	<870k
BiLSTM	80.32	81.28	78.82	80.03	<100k
FAMIC	83.75	83.08	84.75	83.91	<1M
BERT	86.72	88.35	83.64	86.39	110M

documents written in formal language using proper words. The language in the wine review dataset is also standard, which makes it appropriate to use the Glove-300-Wiki embeddings. Words that are not present in the Glove-300-Wiki vocabulary were removed, resulting in an elimination of 3.3% of the words. For the Sentiment140 dataset, we employ word2vec (Mikolov et al., 2013) to generate word embeddings. The tweets in Sentiment140 were often written in informal language, so employing word2vec to generate wording embeddings allows FAMIC to potentially use better word representation in Twitter Sentiment140.

Both datasets are partitioned into training, validation, and test sets using an 18:1:1 ratio. The goal of this evaluation is to compare the performance of FAMIC against several commonly used sentiment analysis (SA) methods in document-level classification. As shown in Table 1, FAMIC achieves the second-highest accuracy on the wine review dataset (0.8899), closely trailing BERT (0.8912), while outperforming BiLSTM (0.8869) and CNN (0.8802). Despite being designed for interpretable WCSA, FAMIC does not compromise its effectiveness in document-level sentiment classification. Interestingly, logistic regression also performs well on the wine review dataset, despite its simplicity. This may suggest that a limited vocabulary of strong sentiment-bearing words is sufficient to capture global sentiment in this domain, motivating further exploration of FAMIC on a more linguistically nuanced dataset.

On the Sentiment140 dataset, FAMIC demonstrates strong performance, outperforming most baseline models but falling short of BERT. We hypothesize that the informal and colloquial nature of language in Sentiment140 may require additional tuning or deeper transformer layers to better capture sentiment nuances. Nevertheless, FAMIC achieves competitive results with significantly fewer parameters than BERT, highlighting its efficiency and interpretability.

Table 2: FAMIC ablation study on Sentiment140 Twitter dataset.

Models	Accuracy (%)	v_{ij}	δ^g_{ij}	δ^l_{ij}	δ^s_{ij}
	79.54	✓	×	×	×
	79.63	✓	×	×	✓
	81.32	✓	✓	×	×
Ablations	81.67	✓	✓	×	✓
	82.34	✓	×	✓	×
	82.54	✓	✓	✓	×
	83.32	✓	×	✓	✓
Full Model	83.73	✓	✓	✓	✓

4.2 Model Ablation Evaluation

To avoid redundancy, the effectiveness of FAMIC in conducing WCSA in the remaining section will be demonstrated exclusively using the Sentiment140 dataset, which features more colloquial, everyday language. To better understand the impact of various model components, we conduct an ablation study focusing on the v_{ij} parameter (representing context-independent sentiment) and the δ parameters—specifically, the global sentiment shifter (δ^g_{ij}), the local sentiment shifter (δ^l_{ij}), and the sentiment word indicator (δ^s_{ij}), as shown in Table 2.

The results indicate that removing the local sentiment shifter leads to the most significant drop in performance, underscoring its importance in recognizing local sentiment patterns. This finding is further supported by the fact that all top-performing FAMIC ablation variants include the local sentiment shifter. As hypothesized, this component plays a critical role in handling negation, making it essential to the model's interpretability. The sentiment word indicator emerges as the next most influential component, while the global sentiment shifter appears to be the least critical. Notably, even without the amplification provided by the global shifter, FAMIC maintains comparable performance.

4.3 Interpretability: WCSA Performance of FAMIC

In this section, we examine the performance of FAMIC in conducting WCSA, focusing on explaining how it is capable of providing interpretable SA results. Specifically, we give detailed description on FAMIC's analysis of Sentence I and II mentioned in Section 2.2 and a number of other examples in the Sentiment140 dataset. These examples are used to illustrate FAMIC's proficiency in providing informative sentiment estimation and its effectiveness in handling delicate linguistic complexities such as negation and sentiment intensifier by incorporating word positional information.

We start with Sentence I (see Table 3), which consists of two separate clauses separated by a comma. The sentiment in the first clause is positive. In the second clause, the subject "the overall experience" is modified "not bad". Although "bad" itself carries a negative sentiment, it is negated by "not", resulting in a positive sentiment too.

Table 3 provides a summary of the components used in the calculation of the context-dependent sentiment of individual words and the document-level sentiment label for Sentence I. The v_{ij} column contains the context-independent sentiment score, which remains constant regardless of the context. For instance, the context-independent sentiment of "good" is 21.2 in

Table 3: FAMIC's analysis result of sentence I (identifying negation).

Raw text	The service of the restaurant is good, the overall experience is not bad.												
Input text	the service	of	the	restaurant	is	good	the	overall	experience	is	not	bad	
v_{ij}	7.10	−1.0	6.9	7.1	19.3	−2.1	21.2	7.1	30.1	11.4	−2.1	−17.4	−28.1
δ^s_{ij}	0	0	0	0	0	0	1	0	0	0	0	1	1
δ^g_{ij}	-	-	-	-	-	-	1.47	-	-	-	-	6.6	2.6
δ^l_{ij}	-	-	-	-	-	-	0.45	-	-	-	-	−0.9	−0.8
z_{ij}	0	0	0	0	0	0	14.2	0	0	0	0	103.8	62.5
Z_i	60.17	Sentiment Label: Positive											

all the sentences it appears. FAMIC identifies "good", "not", and "bad" as sentiment words (i.e., $\delta^s_{ij} = 1$). FAMIC effectively handles negation by recognizing "bad" in the sentence is part of "not bad." The negative context-independent sentiment of "bad", −28.1, after being negated by "not", has a negative local shifter, −0.8, resulting in a positive context-dependent sentiment of 62.5. The document-level sentiment score is 60.17, indicating an overall positive sentiment conveyed in Sentence I.

Sentence II, which has the same collection of words as Sentence I, also consists of two clauses (see Table 4). The only difference is the location of the word "not". Both clauses express a negative sentiment. FAMIC has identified three sentiment words in Sentence II: "not", "good", and "bad". In the first clause, the positive context-independent sentiment score of "good" (21.2, the same value as in Sentence I) is reversed by "not" in the phrase "not good" leading to a negative context-dependent sentiment score of −30.7. In the second clause, the sentiment of "bad" is not reversed because there are no negation words nearby, resulting in a context-dependent sentiment score of −11.7. The document-level sentiment score is −44.37, indicating an overall negative sentiment conveyed in Sentence II.

It is also interesting to point out the different treatment of "not" in these two sentences by FAMIC. Note that 'not" has the same content-independent sentiment score of −17.4 (a negative sentiment) in both cases. This makes sense because "not" is typically used to express negation, denial, refusal, or prohibition. In Sentence I, "not" is placed next to "bad". Though both words carry a context-independent negative sentiment, together "not bad" conveys a positive sentiment.

Table 4: FAMIC's analysis result of sentence II (identifying negation).

Raw text	The service of the restaurant is not good, the overall experience is bad.												
Input text	the service	of	the	restaurant	is	not	good	the	overall	experience	is	bad	
v_{ij}	7.10	−1.0	6.9	7.1	19.3	−2.1	−17.4	21.2	7.1	30.1	11.4	−2.1	−28.1
δ^s_{ij}	0	0	0	0	0	0	1	1	0	0	0	0	1
δ^g_{ij}	-	-	-	-	-	-	6.59	1.46	-	-	-	-	2.66
δ^l_{ij}	-	-	-	-	-	-	0.79	−0.99	-	-	-	-	0.16
z_{ij}	0	0	0	0	0	0	−91.7	−30.7	0	0	0	0	−11.7
Z_i	−44.37	Sentiment Label: Negative											

Table 5: FAMIC's analysis result of a succeeding negation example.

Raw text	Celebrating Phil being one year cancer free!						
Input text	celebrating	phil	being	one	year	cancer	free
v_{ij}	32.9	17.3	−11.5	2.3	−4.6	−38.6	19.9
δ^s_{ij}	1	1	0	0	0	1	1
δ^g_{ij}	1.98	1.63	-	-	-	1.4	2.03
δ^r_{ij}	0.73	0.79	-	-	-	−0.28	0.45
z_{ij}	47.80	22.5	0	0	0	14.9	18.5
Z_i	25.93	Sentiment Label: Positive					

Thus, the sentiment polarity of "not" is flipped to a positive sentiment with 103.8 as its context-dependent sentiment score. In Sentence II, the sentiment of "not" is not flipped as it is positioned adjacent to "good". It is even strengthened to have a more negative context-dependent sentiment score of −91.7, capturing the clear negative sentiment expressed in the phrase "not good". This comparison further demonstrates FAMIC's capability in providing nuanced understanding and accurate identification of sentiment in sentences by incorporating word position information.

Natural language has a rich representation of negative expressions, where negation can be classified into different groups in different ways (Xiang et al. (2014)), for example, preceding negation vs. succeeding negation by the location of the negation word with respect to the negated concept, or explicit negation vs. implicit negation by whether negation is in the asserted meaning or in the non-asserted content. The negation structure in both Sentence I and Sentence II is considered to be preceding negation, because the negation word "not" precedes the word whose sentiment is negated by it. It also falls into the category of explicit negation because of the use of an explicit negation word "not". The following example tweet demonstrates the model's ability to capture succeeding negation with the word "free" (Table 5).

FAMIC recognizes "celebrating", "phil", "cancer", and "free" as sentiment words, among which only "cancer" conveys a negative context-independent sentiment, while the other three words carry a positive context-independent sentiment. The succeeding negation word "free" negates the sentiment conveyed by "cancer". Consequently, the context-independent sentiment score of "cancer" at −38.6 is shifted to a positive context-dependent sentiment score of 14.9, accurately interpreting the positive sentiment conveyed by the phrase "cancer-free". In contrast, the context-independent sentiment scores of "celebrating", "phil", and "free" remain unchanged, contributing to the overall positive sentiment conveyed in the tweet.

Next we present an example of implicit negation (Table 6). The first portion of the sentence, "Chicago was awesome", expresses a positive sentiment, whereas the second portion, "my dreams were shattered", conveys a strong negative sentiment. FAMIC identifies "awesome", "although", "dreams", and "shattered" as sentiment words in the sentence. The phrase "my dreams were shattered" contains implicit negation, as it conveys a negative sentiment without using explicit negation words like "not" or "never". The implicit negation is implied through the word "shattered". Because of it, the word "dreams" which carries a positive context-independent sentiment (11.3), is coupled with a negative local sentiment shifter, resulting in a negative context-dependent sentiment (−20.9). The strong negative sentiment in the second part of the sentence leads to an overall negative document-level sentiment.

Table 6: FAMIC's analysis result of an implicit negation example.

Raw text	Chicago was awesome although my dreams were shattered.							
Input text	chicago	was	awesome	although	my	dreams	were	shattered
v_{ij}	1.7	−3.1	35.1	2.1	−7.4	11.3	−5.6	−25.5
δ^s_{ij}	0	0	1	1	0	1	0	1
δ^g_{ij}	-	-	1.30	0.96	-	4.25	-	1.31
δ^l_{ij}	-	-	0.04	0.87	-	−0.43	-	0.50
z_{ij}	0.0	0.0	1.7	1.7	0	−20.9	0.0	−16.8
Z_i	−8.57	Sentiment Label: Negative						

Table 7: FAMIC analysis result of an intensifier word.

Raw text	Bad, very bad!		
Input text	bad	very	bad
v_{ij}	−28.1	3.8	−28.1
δ^s_{ij}	1	0	1
δ^g_{ij}	7.61	-	7.61
δ^l_{ij}	0.67	-	0.92
z_{ij}	−142.6	0.0	−196.7
Z_i	−169.7	Sentiment Label: Negative	

In addition to negation handling, FAMIC can also deal with other types of language complexity, such as use of intensifiers, which are adverbs or adverbial phrases that strengthen the meaning of other expressions and show emphasis. Table 7 shows such an example. It has two short phrases, "bad" and "very bad." Apparently, "very bad" delivers a stronger negative sentiment than just "bad". The global sentiment shifter, without bearing positional awareness, can not recognize that "very" only intensifies the second "bad", and it takes the same value of 7.61 for both occurrences of "bad" in the sentence. However, the local sentiment shifter, equipped with positional awareness, accurately recognizes that "very" is a function word that only intensifies the sentiment of the second "bad" but not the first "bad". As a result, the local sentiment shifter for the second "bad" is larger that for the first "bad", resulting in a stronger negative sentiment for the second "bad" correctly.

5 Conclusion

In this study, we introduced FAMIC, an interpretable sentiment analysis model that operates at the word level while requiring only document-level sentiment labels. FAMIC achieves competitive performance relative to BERT-based baselines while providing a transparent architecture with substantially fewer parameters, making it efficient and accessible.

FAMIC's interpretability is driven by two key components: a local sentiment shifter that models positional effects such as negation scope and sentiment intensity, and a sentiment word indicator that suppresses neutral or functional words and concentrates explanations on sentiment-

bearing tokens. Through qualitative examples, we showed that FAMIC produces fine-grained word-level, context-based sentiment explanations without relying on word-level annotations, sentiment lexicons, or seed word lists, making it scalable to settings where only document-level supervision is available.

We also note several limitations and directions for extension. First, while word-level scores are directionally meaningful, their absolute magnitudes may not be immediately intuitive; calibration to a standardized scale could improve interpretability and comparability across documents and domains. Second, the current architecture is intentionally shallow to preserve transparency and controllability, which may limit its ability to capture more complex compositional interactions; a natural extension is to incorporate deeper self-attention layers to model richer context while retaining the explicit word-level decomposition and interpretable shifter structure. Third, our experiments are single-domain, and future work will examine cross-domain training or pretraining to study transfer effects on both accuracy and explanation stability.

We do not compare FAMIC directly with large language models such as Claude, GPT, or LLaMA and therefore make no claims about relative performance. Nonetheless, controlled comparisons between compact intrinsically interpretable models and general-purpose large language models remain an important direction for understanding trade-offs among model size, computational cost, domain specificity, and interpretability.

Supplementary Material

The full codebase, pretrained model weights, and step-by-step tutorials (Jupyter notebook and Google Colab notebook) for reproducing all results are available at the public GitHub repository: https://github.com/YCY198888/FAMIC.

A Experimental and Training Details

This appendix describes the empirical training procedure used for FAMIC and provides practical guidance on the penalty coefficients c_1, c_2, and c_3 in Equation (6). These coefficients are central to the intended interpretability behavior, since they control (i) whether the proxy sentiment indicator δ^s becomes quasi-binary, (ii) how sparse the identified sentiment words are, and (iii) the stability of the overall shifted sentiment magnitude. In our experiments, model behavior is most sensitive to the timing and magnitude of c_1 and c_2, particularly early in training. This sensitivity is expected because the word-level contribution is multiplicative and δ^s is encouraged to become sparse and close to binary; if c_1 or c_2 is set too large at initialization, δ^s can be pushed toward zero for many tokens before the model has learned which words carry sentiment signal. In this situation, many tokens receive little effective gradient signal through the multiplicative form, and they may not recover as sentiment words later. To ensure stable learning, we adopt a staged training strategy with a gradual penalty schedule.

A.1 Two-Stage Training with Gradual Penalty Scheduling

We use a two-stage procedure. In Stage 1, we train only the context-independent sentiment module and the sentiment word identification module while excluding the sentiment shifters, and we set c_1 and c_2 to very small values. This warm start provides a stable baseline and allows the model to identify candidate sentiment words before strong sparsity and binarity constraints

are enforced; empirically, this stage requires only a small amount of training, often less than one epoch. In some runs, we additionally set c_2 to a small negative value for a brief warm-up period (typically only a few batches). The purpose is to slightly bias δ^s toward non-zero activations at initialization, ensuring that each word receives sufficient gradient signal and has a "fighting chance" to be identified as sentiment-bearing before sparsity is enforced. After this brief warm-up, we return c_2 to a small positive value and proceed with the standard schedule.

In Stage 2, we train the full model including both global and local sentiment shifters, still starting from small penalties, and we gradually increase c_1 and c_2 during training. Increasing c_1 encourages δ^s to approach a quasi-binary indicator as intended, while increasing c_2 promotes sparsity by pushing many tokens toward $\delta^s \approx 0$, effectively treating them as functional words that do not contribute to the document-level sentiment. We keep c_3 fixed at a small value throughout training, since its primary role is stabilizing the magnitude of the shifted sentiment rather than controlling sparsity or binarity.

A.2 Practical Guidance for Choosing c_2 and Avoiding Degenerate Behavior

The coefficient c_2 effectively controls how sparse the final selection of sentiment words will be. While larger c_2 values can yield cleaner and more selective explanations, setting c_2 too high—especially early in training—can lead to degenerate solutions. One common failure mode is a "dead activation" regime in which many tokens are pushed to $\delta^s \approx 0$ prematurely, reducing gradient flow and preventing recovery of meaningful sentiment words. Another failure mode can occur when c_2 is excessively large relative to the classification objective: the model may reduce the effective penalty by assigning non-zero δ^s to frequent functional words (e.g., "the", "a") that appear broadly across documents. In this case, such high-frequency tokens can act as a surrogate pooling mechanism, allowing the model to propagate document-level information while avoiding the intended sparsity constraint on true sentiment-bearing words. Practically, this motivates (i) starting with small penalties, (ii) ramping c_2 only after the model has learned reasonable sentiment word candidates, and (iii) monitoring whether high-frequency function words are being incorrectly identified as sentiment words, which indicates that c_2 is too strong or introduced too early.

A.3 Implementation Notes and Reproducibility

Due to the sensitivity of the training dynamics and the possibility of entering degenerate regimes (e.g., dead activations from overly large early c_2, or the pooling behavior described above), training often requires active monitoring of intermediate quantities such as the distribution of δ^s, the fraction of words identified as sentiment-bearing, and the identity of the highest-scoring tokens. In our experience, if these diagnostics indicate that training has entered an undesirable regime, restarting with a milder penalty schedule is more effective than continuing optimization. For this reason, it is difficult to provide a single fully automatic training script that is robust across datasets and random initializations. To support reproducibility and transparency, we provide a public GitHub repository (Yang, 2025) that includes the full codebase and pretrained model weights, allowing readers to validate the reported performance directly. We also provide a step-by-step tutorial in a Jupyter notebook and a Google Colab notebook to facilitate easy execution and inspection of results. Specifically, the repository (i) reports classification performance in terms of accuracy, F1, precision, and recall; (ii) demonstrates that tokenization is performed at the word level; (iii) documents the core implementation components of FAMIC, including the

explicit decomposition of sentiment factors, the mask-based sentiment word identification module, the global and local sentiment shifters, and the final orchestration that produces word- and document-level outputs; and (iv) includes our novel implementation of multi-head self-attention with relative positional embeddings.

References

Abnar S, Zuidema W (2020). Quantifying attention flow in transformers. arXiv preprint: https://arxiv.org/abs/2005.00928

Balderas L, Lastra M, Benítez JM (2023). Can persistent homology whiten transformer-based black-box models? A case study on bert compression. arXiv preprint: https://arxiv.org/abs/2312.10702

Bilan I, Roth B (2018). Position-aware self-attention with relative positional encodings for slot filling. arXiv preprint: https://arxiv.org/abs/1807.03052

Carbonneau MA, Cheplygina V, Granger E, Gagnon G (2018). Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77: 329–353. https://doi.org/10.1016/j.patcog.2017.10.009

Chen K, Wang R, Utiyama M, Sumita E (2021). Context-aware positional representation for self-attention networks. *Neurocomputing*, 451: 46–56. https://doi.org/10.1016/j.neucom.2021.04.055

Das SR, Chen MY (2007). Yahoo! For Amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9): 1375–1388. https://doi.org/10.1287/mnsc.1070.0704

Devlin J, Chang MW, Lee K, Toutanova K (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Dietterich TG, Lathrop RH, Lozano-Pérez T (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1): 31–71. https://doi.org/10.1016/S0004-3702(96)00034-3

Fang X, Zhan J (2015). Sentiment analysis using product review data. *Journal of Big Data*, 2, Article number: 5. https://doi.org/10.1186/s40537-015-0015-2

Ghader H, Monz C (2017). What does attention in neural machine translation pay attention to? arXiv preprint: https://arxiv.org/abs/1710.03348

Go A, Bhayani R, Huang L (2009). Twitter sentiment classification using distant supervision. CS224N project report. *Stanford*, 1, Article number: 12.

Hochreiter S, Schmidhuber J (1997). Long short-term memory. *Neural Computation*, 9(8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jain S, Wallace BC (2019). Attention is not explanation. arXiv preprint: https://arxiv.org/abs/2005.00928

Katumullage D, Yang C, Barth J, Cao J (2022). Using neural network models for wine review classification. *Journal of Wine Economics*, 17(1): 27–41. https://doi.org/10.1017/jwe.2022.2

Kim Y (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kindermans PJ, Hooker S, Adebayo J, Alber M, Schütt KT, ..., Kim B (2017). The (un)reliability of saliency methods. arXiv preprint: https://arxiv.org/abs/1711.00867

Liu B (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1): 1–167. https://doi.org/10.1007/978-3-031-02145-9

Lundberg SM, Lee S (2017). A unified approach to interpreting model predictions. CoRR. arXiv preprint: https://arxiv.org/abs/1705.07874

Medhat W, Hassan A, Korashy H (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4): 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Mikolov T, Chen K, Corrado GS, Dean J (2013). Efficient estimation of word representations in vector space. arXiv preprint: https://arxiv.org/abs/1301.3781

Minaee S, Azimi E, Abdolrashidi A (2019). Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. arXiv preprint: https://arxiv.org/abs/1904.04206

Nivre J (2005). Dependency grammar and dependency parsing. *MSI Report*, 5133(1959): 1–32.

Pang B, Lee L, Vaithyanathan S (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 79–86.

Petch J, Di S, Nelson W (2022). Opening the black box: The promise and limitations of explainable machine learning in cardiology. *Canadian Journal of Cardiology*, 38(2): 204–213. https://doi.org/10.1016/j.cjca.2021.09.004

Ray S, Page D (2001). Multiple instance regression. In: *ICML* (CE Brodley, AP Danyluk, eds.), 425–432. Morgan Kaufmann.

Read J (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. *Proceedings of the ACL Student Research Workshop*, 43–48.

Ribeiro MT, Singh S, Guestrin C (2016). "Why should I trust you?": Explaining the predictions of any classifier. arXiv preprint: https://arxiv.org/abs/1602.04938

Rudin C (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. arXiv preprint: https://arxiv.org/abs/1811.10154

Seonwoo Y, Kim JH, Ha JW, Oh A (2020). Context-aware answer extraction in question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2418–2428.

Shaw P, Uszkoreit J, Vaswani A (2018). Self-attention with relative position representations. arXiv preprint: https://arxiv.org/abs/1803.02155

Singh C, Hsu AR, Antonello R, Jain S, Huth AG, ..., Gao J (2023). Explaining black box text modules in natural language with language models. arXiv preprint: https://arxiv.org/abs/2305.09863

Singh M, Jakhar AK, Pandey S (2021). Sentiment analysis on the impact of coronavirus in social life using the bert model. *Social Network Analysis and Mining*, 11(1): 33. https://doi.org/10.1007/s13278-021-00737-z

Tabinda Kokab S, Asghar S, Naz S (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, 14:100157. https://doi.org/10.1016/j.array.2022.100157

Thogesan T, Nugaliyadde A, Wong KW (2025). Integration of explainable ai techniques with large language models for enhanced interpretability for sentiment analysis. arXiv preprint: https://arxiv.org/abs/2503.11948

Tyagi A, Sharma N (2018). Sentiment analysis using logistic regression and effective word score heuristic. *International Journal of Engineering and Technology (UAE)*, 7: 20–23.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, ..., Polosukhin I (2017). Attention is all

you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Wu Y, Schuster M, Chen Z, Le QV, Norouzi M,..., Dean J (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint: https://arxiv.org/abs/1609.08144

Wu Z, Ong DC (2021). Context-guided bert for targeted aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35:16, 14094–14102.

Xiang M, Grove J, Giannakidou A (2014). Semantic and pragmatic processes in the comprehension of negation: An event related potential study of negative polarity sensitivity. *Journal of Neurolinguistics*, 38: 71–88.

Xiong D, Park S, Lim J, Wang T, Wang X (2024). Bayesian multiple instance classification based on hierarchical probit regression. *The Annals of Applied Statistics*, 18(1): 80–99. https://doi.org/10.1214/23-AOAS1780

Yang C (2025). Famic: Code and pretrained models. https://github.com/YCY198888/FAMIC. Accessed: 2025-12-27

Yang C, Cao J (2025). Interpretable sentiment analysis using the attention-based multiple instance classification model: An application to wine reviews. *Harvard Data Science Review*, 7(2). https://doi.org/10.1162/99608f92.caab9466

Zhang S, Zheng D, Hu X, Yang M (2015). Bidirectional long short-term memory networks for relation classification. *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, 73–78.

Zhang Y, Wallace BC (2017). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 253–263.

Zhao A, Yu Y (2021). Knowledge-enabled bert for aspect-based sentiment analysis. *Knowledge-Based Systems*, 227:107220. https://doi.org/10.1016/j.knosys.2021.107220

Zhou B, Khosla A, Lapedriza À, Oliva A, Torralba A (2015). Learning deep features for discriminative localization. CoRR. arXiv preprint: https://arxiv.org/abs/1512.04150