applied sciences

MDPI

Article

Interpretability in Sentiment Analysis: A Self-Supervised Approach to Sentiment Cue Extraction

Yawei Sun [1,2], Saike He [3,*], Xu Han [4] and Yan Luo [5]

[1] Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China; sunyawei@bupt.edu.cn
[2] School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China
[3] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[4] Institute of Scientific and Technical Information of China, Beijing 100038, China
[5] Institute of Medical Information, Chinese Academy of Medical Sciences & Peking Union Medical College, Beijing 100020, China
* Correspondence: saike.he@ia.ac.cn

Abstract: In this paper, we present a novel self-supervised framework for Sentiment Cue Extraction (SCE) aimed at enhancing the interpretability of text sentiment analysis models. Our approach leverages self-supervised learning to identify and highlight key textual elements that significantly influence sentiment classification decisions. Central to our framework is the development of an innovative Mask Sequence Interpretation Score (MSIS), a bespoke metric designed to assess the relevance and coherence of identified sentiment cues within binary text classification tasks. By employing Monte Carlo Sampling techniques optimized for computational efficiency, our framework demonstrates exceptional effectiveness in processing large-scale text data across diverse datasets, including English and Chinese, thus proving its versatility and scalability. The effectiveness of our approach is validated through extensive experiments on several benchmark datasets, including SST-2, IMDb, Yelp, and ChnSentiCorp. The results indicate a substantial improvement in the interpretability of the sentiment analysis models without compromising their predictive accuracy. Furthermore, our method stands out for its global interpretability, offering an efficient solution for analyzing new data compared to traditional techniques focused on local explanations.

Keywords: sentiment cue extraction; self-supervised learning; interpretable machine learning

Citation: Sun, Y.; He, S.; Han X.; Luo, Y. Interpretability in Sentiment Analysis: A Self-Supervised Approach to Sentiment Cue Extraction. *Appl. Sci.* **2024**, *14*, 2737. https://doi.org/10.3390/app14072737

Academic Editor: Douglas O'Shaughnessy

Received: 26 February 2024
Revised: 20 March 2024
Accepted: 20 March 2024
Published: 25 March 2024

1. Introduction

In the rapidly evolving landscape of the information age, the prolific growth of textual data on various online platforms has propelled Natural Language Processing (NLP) into a position of increased importance. Within this domain, sentiment analysis [1], also referred to as opinion mining, stands out as a critical area. This process involves the automatic detection and interpretation of sentiments, emotions, and subjective information within textual data [2]. The application of sentiment analysis spans a wide spectrum, from the analysis of customer feedback in product reviews to the evaluation of public sentiment on social media platforms [3].

In the ever-evolving digital landscape, the exponential growth of textual data across various online platforms has elevated NLP to a critical technological frontier. Among the myriad applications of NLP, sentiment analysis plays a pivotal role. This field, focusing on the automatic detection and interpretation of sentiments, emotions, and subjective information within textual content, finds widespread application from analyzing customer feedback in product reviews to monitoring public sentiment on social media platforms.

Despite the remarkable advances and successes in sentiment analysis, a significant hurdle persists: the challenge of interpretability, which encompasses the difficulty of understanding and explaining how sentiment analysis models make their decisions, particularly in terms of identifying specific factors or textual elements that influence these decisions [4]. Traditional sentiment analysis models are often criticized for their "black-box" nature, which obscures the transparency of their decision-making processes [5]. This opacity generates concerns about accountability and dependability, especially in scenarios where precision and reliability are paramount.

To mitigate these concerns, our research introduces a novel self-supervised framework focused on sentiment cue extraction. This approach involves the identification and extraction of crucial linguistic elements—such as specific words, phrases, or syntactic patterns, referred to as "sentiment cues" in this paper—that significantly influence sentiment determination. Our approach is instrumental in demystifying the decision-making process of sentiment analysis models, thus contributing to a deeper understanding and trust in these systems.

For example, in finance, discerning the exact cues that drive sentiment predictions can be a game changer for market analysis [6–8]. Similarly, in healthcare, the analysis of sentiment cues in patient feedback, particularly from online sources, is essential to improve the quality of healthcare services. By evaluating positive and negative sentiments expressed in patient reviews, healthcare providers can identify strengths and areas for improvement in their services, such as facility cleanliness, staff behavior, and general patient care [9].

Our study introduces a groundbreaking framework based on self-supervised learning that incorporates sequence labeling techniques to significantly improve the interpretability of sentiment analysis models. Traditional approaches in sentiment cue extraction often involve labor-intensive and time-consuming data annotation processes. Existing interpretability methods for text classification models, while offering partial solutions, primarily depend on local interpretative methods. These local methods typically require individual training for each data instance, presenting significant challenges in efficiently handling new data.

In contrast, our innovative approach uses the abundance of existing annotated sentiment classification data through self-supervised learning. This enables our framework to interpret sentiment classification models in scenarios where explicit annotation is lacking, effectively facilitating sentiment cue extraction. Importantly, this methodology transcends the boundaries of local interpretability techniques, offering a global interpretability approach. Such a global perspective allows for a more holistic and comprehensive understanding of the model's decision-making process across various instances, rather than being confined to localized, instance-specific explanations.

To the best of our knowledge, ours is the first work to combine Monte Carlo methods with self-supervised learning to address the global interpretability issue in binary text classification [10–12]. The key contributions of our research are as follows.

- We propose a Self-Supervised Sentiment Cue Extraction (SS-SCE) method. This approach, inspired by the concept of interpretability in text classification models, accomplishes the extraction of sentiment cues from texts under conditions of scarce labeled data through a global interpretability analysis of the text classification models.
- We have developed a pseudo-label generation scheme for sentiment cue extraction models. This scheme selects appropriate mask sequences as pseudo labels for the sentiment cue extraction model based on the prediction results of a trained text classification model. Furthermore, we enhance the efficiency of pseudo-label generation by employing a Monte Carlo Sampling strategy.
- We have introduced the Mask Sequence Interpretation Score (MSIS) metric, designed to evaluate generated mask sequences based on the prediction results of a text classification model, thereby providing a basis for the generation of pseudo labels. Empirical evidence demonstrates the effectiveness of our MSIS metric.

The remainder of this paper is organized as follows: Section 2 discusses related work, providing background on sentiment analysis, self-supervised methods for information extraction, interpretability in machine learning, and the use of Monte Carlo methods. Section 3 details our methodology, explaining the sentiment cue extraction process, the use of Monte Carlo sampling, label sequence selection, and the sentiment cue extraction algorithm. In Section 4, we present our experimental setup, the datasets used, and a thorough evaluation of the performance of the SS-SCE framework. This includes an in-depth analysis of our results and a comparative study with state-of-the-art interpretability methods. Finally, Section 5 concludes the paper, summarizing our key findings, discussing the implications and potential applications of our work, and suggesting avenues for future research.

2. Related Works

2.1. Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a crucial subfield of NLP that focuses on discerning and categorizing opinions expressed in text [13,14]. Its primary goal is to determine the writer's position toward specific topics or the general polarity of the sentiment of the text. This analysis typically involves categorizing text polarity at various levels: document, sentence, or feature/aspect level, determining whether the expressed opinion is positive, negative, or neutral [3].

With the advent of deep learning, sentiment analysis has undergone significant advances. Models such as Bidirectional Encoder Representations from Transformers (BERT) and its variants have been extensively employed for nuanced sentiment analysis, enhancing context and semantic understanding [15,16]. Moreover, transformer-based models like GPT-3 have pushed the boundaries further in generating human-like text, which is advantageous for more intricate sentiment analysis scenarios [17].

Sentiment analysis finds extensive applications across various domains, from customer service and market research to social media monitoring and political campaigns. It is essential for businesses and organizations to gauge public opinion, conduct market research, monitor the reputation of the brand and the product, and understand customer experiences [1].

In today's era of advanced NLP technology, sentiment analysis has emerged as a highly focused research area within the field, benefiting from a plethora of readily available high-quality datasets, such as IMDb [18] and SST-2 [19]. This availability has injected significant vitality into research in this direction. However, the "black box" nature of many deep learning models used in sentiment analysis poses another major limitation. These models, while powerful, often lack transparency in their decision-making processes, making it difficult for users to understand and trust their predictions.

Furthermore, sentiment analysis faces challenges in detecting nuances such as sarcasm, irony, and context-dependent meanings. Future research may involve more sophisticated models that understand complex human emotions and incorporate multimodal data (text, images, and videos) to better understand sentiments [20].

The field of sentiment analysis in NLP continues to be dynamic, with ongoing efforts to enhance the accuracy and versatility of sentiment detection algorithms. As computational models evolve, their ability to discern sentiments from text is expected to become increasingly refined and sophisticated.

2.2. Self-Supervised Methods for Information Extraction

Self-supervised learning in NLP has emerged as a fundamental approach to information extraction, harnessing the potential of unlabeled data to train predictive models. This paradigm involves creating learning tasks in which models predict certain parts of the input using other parts [21–23].

By utilizing large volumes of unlabeled data, self-supervised learning allows models to learn rich representations. These representations are beneficial for diverse downstream

NLP tasks, especially valuable in contexts where labeled data are scarce or expensive to acquire [16,21].

Among the popular methodologies in self-supervised learning, Masked Language Modeling (MLM) stands out. MLM is a key technique in self-supervised learning, notably used by BERT. It involves hiding some words in a sentence and training the model to predict these hidden words using the surrounding context. This process aids in understanding the context and relationships between words [16].

Permutation-based language modeling, as introduced by XLNet, is another significant methodology. It extends the concept of MLM to predict a token based on all permutations of tokens in a sentence. This approach offers a more comprehensive context understanding [22].

Additionally, models like BART [23] and Text-to-Text Transfer Transformer (T5) [24] utilize a corrupted text generation task for pre-training. In this approach, models learn to reconstruct the original text from a corrupted version, thereby enhancing their understanding of language structure and coherence [23].

In the evolving landscape of self-supervised learning models, the Generative Pre-trained Transformer (GPT) series by OpenAI marks a pivotal juncture [17,25,26]. Unlike BERT, renowned for its bidirectional approach to language comprehension, GPT models excel at text generation by predicting the subsequent word in a sequence. Consequently, while BERT shines in nuanced language understanding tasks, GPT excels in producing coherent and contextually apt text.

Continuing this trajectory, ChatGPT (https://chat.openai.com (accessed on 19 March 2024)), a notable addition to the GPT lineage, heralds further breakthroughs. Specifically, ChatGPT exemplifies the prowess of large-scale language models across an array of uses, from crafting human-like narratives to conducting nuanced sentiment analyses. Its adaptability for fine-tuning targeted tasks significantly expands its utility and effectiveness in addressing diverse NLP challenges. Parallel to ChatGPT's emergence, a myriad of other large language models like Gemini (https://gemini.google.com/ (accessed on 19 March 2024)) and ERNIE Bot (https://yiyan.baidu.com/ (accessed on 19 March 2024)) have surfaced, enriching the field with their distinct contributions.

However, these advancements are not without challenges. ChatGPT's closed-source nature hinders research transparency and restricts community-driven enhancements. Moreover, the substantial computational resources required to operate or fine-tune such models often necessitate reliance on cloud-based APIs provided by the developers. This reliance raises concerns regarding cost-effectiveness, latency issues, and data privacy implications [27,28].

Self-supervised learning has achieved remarkable success in tasks such as named entity recognition, relation extraction, and event extraction. By pretraining on extensive text corpora, these models capture nuanced language patterns, significantly increasing their task performance [29,30].

2.3. Interpretability of Deep Learning Models

The interpretability of deep learning models in NLP is a vital research area, concentrating on deciphering and explaining how these models make decisions. This aspect is particularly critical in applications where trust and transparency are paramount [4,31].

Interpretability in deep learning models is essential to validate and improve model performance, ensure fairness, and provide information on model behavior, especially in areas such as healthcare, finance, and legal applications [31].

Several techniques have been developed to enhance the interpretability of deep learning models. These include attention mechanisms, which underscore parts of the input data most relevant to the model decision [32], and Local Interpretable Model-Agnostic Explanations (LIME), which approximate the model locally using interpretable models [5]. In addition, researchers also use topic models such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) to achieve interpretability [33]. For example, Xiong and Li [34] combined LDA with deep learning models to not only grade student essays but also identify the characteristics of excellent essays in terms of language expression.

Despite these advances, understanding deep learning models, particularly transformers, remains challenging. Their black-box nature often hinders the understanding of their predictive reasoning [4].

Future research in model interpretability is likely to focus on developing more robust generalizable techniques that offer clear explanations of model decisions, including integrating interpretability directly into model architecture and training [35].

As deep learning models continue to advance and find application in critical domains, the significance of interpretability will only escalate. Ensuring that these models are transparent and that their decisions are understandable is key to their successful and ethical application.

2.4. Monte Carlo Methods

Monte Carlo methods represent a class of computational algorithms that employ repeated random sampling to yield numerical outcomes. In the realms of NLP and machine learning, these methods are applied across a spectrum of tasks, including optimization, numerical integration, and probabilistic inference [36,37].

The foundational principle of Monte Carlo methods is the utilization of randomness to address problems that, while theoretically deterministic, are complex in nature. These methods are particularly effective in computing quantities that are challenging for deterministic algorithms, largely because of their high-dimensional characteristics.

In the field of NLP, Monte Carlo methods have found extensive applications in language modeling, particularly in tasks that encompass uncertainty and probabilistic models. A notable example of their application is in Bayesian learning methodologies, where they are instrumental in estimating the posterior distributions of model parameters [38].

Recent progress in Monte Carlo methods has geared towards enhancing both efficiency and accuracy, especially within the context of deep learning. Techniques such as Markov Chain Monte Carlo (MCMC) have been adapted for compatibility with complex model structures, including deep neural networks [37].

A primary challenge in the implementation of Monte Carlo methods within NLP pertains to the computational demands, which are accentuated when large datasets and intricate model architectures. Consequently, future research is anticipated to focus on the development of more efficient sampling techniques and the integration of Monte Carlo methods with other machine learning approaches [39].

3. Task Definition

The primary aim of this study is to enhance the interpretability of sentiment classification models applied to texts. Specifically, our focus is on identifying the key factors—words or phrases within a text—that sentiment classification models rely on to determine the sentiment polarity of that text. These influential words or phrases are collectively referred to as "Sentiment Cues". Therefore, we term the task we explore in this paper as Sentiment Cue Extraction (SCE). This endeavor seeks to uncover and articulate the rationale behind sentiment polarity judgments made by these models, making the decision-making process more transparent and understandable to both users and researchers.

To clarify the task of SCE more distinctly, let us illustrate with the following two examples:

- `Instance 1`: Very **friendly** customer service.
- `Instance 2`: If I could give a **zero star**, I would!

Here, the word "friendly" in the first instance allows us to identify its sentiment as positive; similarly, the phrase "zero star" in the second instance indicates a negative sentiment. Hence, "friendly" and "zero star" serve as what we define as sentiment cues.

4. Methodology

4.1. Overview of Our Method

To address the SCE task, this paper conceptualizes SCE as a sequence labeling task. This perspective allows for a systematic approach to identifying sentiment cues across varying textual instances.

Given an instance $X = \{x_1, x_2, \cdots, x_n\}$, our objective is to assign a corresponding label set $Y = \{y_1, y_2, \cdots, y_n\}$, where $y_i = 1$ signifies that the element constitutes a significant sentiment cue. For example, regarding Instance 1, the corresponding X and Y are as illustrated in Equations (1) and (2), respectively.

$$X = \{\text{“}Very\text{”}, \text{“}friendly\text{”}, \text{“}customer\text{”}, \text{“}service\text{”}, \text{“}.\text{”}\}, \tag{1}$$

$$Y = \{0, 1, 0, 0, 0\}. \tag{2}$$

However, a principal challenge within this work is the absence of annotated data for the SCE task, meaning that Y is unknown within the dataset. To address this, we introduce a Self-Supervised Sentiment Cue Extraction (SS-SCE) method that employs self-supervised learning to tackle the SCE task. In the SS-SCE framework, we utilize a sentiment classification model, which has been widely labeled, to generate pseudo labels for the SCE task. These pseudo labels, derived from samples X in the sentiment classification dataset, serve as inputs and outputs for constructing the SCE training dataset, thereby enabling the training of an SCE sequence labeling model. The fundamental steps of this approach are depicted in Figure 1.

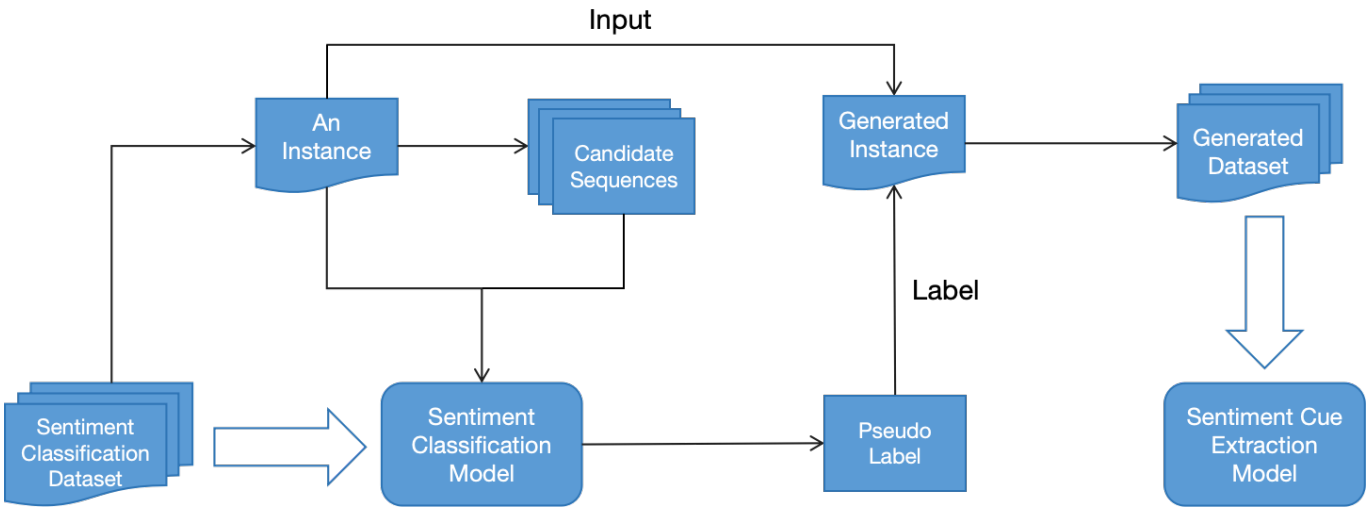

Figure 1. This figure illustrates the workflow of our self-supervised sentiment cue extraction method. “Input” and “Label” represent the roles of “An Instance” and “Pseudo Label” within the “Generated Instance”, respectively. The bold arrows indicate the process of training the corresponding models using the dataset.

Figure 1 illustrates the basic workflow of our method. Initially, we train a sentiment classification model based on a sentiment classification dataset. Building on this, we generate candidate sequences of pseudo labels (referred to as the Candidate Sequences in the figure) for an instance within the dataset and then use the sentiment classification model to select one sequence from these candidates as the pseudo label. Thus, by taking an instance as input and using the obtained pseudo label as the label, we can form a sequence labeling instance (referred to as the Generated Instance in the figure). By generating such generated instances for other instances in the sentiment classification, we can compile a

dataset (referred to as the Generated Dataset in the figure) that is suitable for training the SCE model. Based on this dataset, the SCE model can then be trained.

In the following sections, we will introduce our method in detail.

4.2. Generating Dataset for Sentiment Cue Extraction

4.2.1. Generation of Candidate Sequences

As described earlier, for an input X, it is necessary to generate several candidate sequences of pseudo labels, denoted by $Y^c = \{y_1^c, y_2^c, \cdots, y_n^c\}$. Theoretically, for an input X of length n, there are 2^n possible configurations for Y^c. This implies that for $n = 20$, Y^c could have over 1 million possible combinations—a daunting figure. This calculation pertains to just a single data instance, whereas training the SCE model requires thousands of such pseudo-labeled data instances. Enumerating all possible combinations is impractical, both in terms of time and computational resources. Therefore, we employ the Monte Carlo Sampling [40] method to randomly generate a specified number of candidate sequences, significantly reducing the time complexity associated with generating these candidate sequences. The Monte Carlo Sampling algorithm we use is outlined in Algorithm 1. This approach allows us to efficiently produce a manageable subset of potential label sequences for further analysis and selection, ensuring the feasibility of the SCE model training process.

Algorithm 1 Monte Carlo Sampling for generating one candidate sequence

Require: instance X, sampling ratio p
1: $Y^c \leftarrow \emptyset$
2: **for** $i \in [1, 2, \ldots, n]$ **do**
3: generate g uniformly at random in the range [0, 1]
4: **if** $g < p$ **then**
5: $y_i^c \leftarrow 1$
6: **else**
7: $y_i^c \leftarrow 0$
8: **end if**
9: Add y_i^c to Y^c
10: **end for**
11: **return** Y^c

In Algorithm 1, we commence by specifying a sampling ratio p. For each element x_i in X, we randomly generate a decimal number g uniformly within the range of 0 to 1. If $g < p$, then y_i^c is set to 1; otherwise, it is set to 0. This mechanism ensures that each y_i^c has a probability p of being assigned the value 1. Consequently, it can be inferred that the proportion of elements labeled 1 in the generated sequence Y^c is expected, on average, to be p.

This method does more than simply allow for the manipulation of positive label density within candidate sequences; it also facilitates the emulation of varied labeling densities in scenarios devoid of pre-annotated data by modulating the p-value. Ideally, p should mirror the proportion of tokens in the text X that significantly influence the sentiment classification model's decision-making process, equivalent to the proportion of elements valued at 1 in Y. However, this proportion is unknown. Therefore, to generate candidate sequences as comprehensively as possible, we employ multiple values for p during the sampling process, conducting sampling under these varied p-values.

4.2.2. Sentiment Classification Model

As demonstrated in Figure 1, selecting an optimal pseudo label from the array of candidate sequences involves scoring each candidate. Within the SS-SCE framework, this scoring process is facilitated by a sentiment classification model. Herein, we provide an overview of the sentiment classification model used in the SS-SCE context.

In our research, the sentiment classification model is built on BERT as the encoding mechanism, mainly due to its ability to effectively capture contextual nuances within the

text. BERT, a transformer-based model, stands out for its dynamic encoding capabilities, compared to static word vector methods, such as GloVe [41], which may not fully grasp the context-dependent aspects of language.

Moreover, compared to GPT [25,26], another transformer-based architecture, BERT is more aligned with our needs. While GPT excels in text generation tasks due to its unidirectional nature, BERT's bidirectional training strategy makes it particularly suitable for understanding the nuanced expressions of sentiment in texts. This bidirectionality allows BERT to gather context from both sides of a token, offering a richer representation of the input text and enhancing the model's ability to discern the underlying sentiment.

Additionally, sentiment classification models within the academic community often leverage BERT-based architectures, facilitating straightforward comparisons with other models in the field.

When encoding the input X using BERT for our sentiment classification model, it is necessary to prepend a [CLS] token at the beginning and append a [SEP] token at the end of X. Thus, the actual sequence inputted into BERT becomes $X = \{[\text{CLS}], x_1, x_2, \cdots, x_n, [\text{SEP}]\}$. For text classification tasks, the encoding of the [CLS] token is typically utilized to represent the encoding of the entire sentence.

To facilitate comparisons with other models, we have constructed a remarkably straightforward binary text classification model f_{sc} based on the base version of BERT. In this model, f_{sc}, the enhanced input X is encoded using BERT, resulting in a 768-dimensional vector representation, h_X^{768}. This vector, specifically derived from the encoding of the [CLS] token, is then transformed into a two-dimensional vector, h_X^2, via a fully connected layer. Subsequently, a softmax function converts h_X^2 into a pair of probabilities that indicate the likelihood of X belonging to categories 0 (negative sentiment) and 1 (positive sentiment), respectively. The sentiment classification model f_{sc} can thus be expressed as:

$$f_{\text{sc}}(X) = \text{softmax}(\text{FC}^{768\times 2}(\text{BERT}(X)_{[\text{CLS}]})) \tag{3}$$

where $\text{FC}^{768\times 2}$ denotes the fully connected layer mapping the 768-dimensional BERT encoding to a 2-dimensional output, and $\text{BERT}(X)_{[\text{CLS}]}$ refers to the representation of the [CLS] token produced by BERT, which serves as the aggregate representation of the enhanced input text for classification purposes.

Accordingly, for an input, the model yields the following probability pair:

$$(p_{c0}, p_{c1}) = f_{\text{cls}}(X), \tag{4}$$

where p_{c0} and p_{c1} correspond to the probabilities of X being classified under negative and positive sentiments, respectively. Consequently, the sentiment prediction for X by f_{cls} is determined as:

$$C = \arg\max(p_{c0}, p_{c1}), \tag{5}$$

This procedure also facilitates the computation of the Probability Discrepancy between the categories:

$$\Delta P = |p_{c0} - p_{c1}|. \tag{6}$$

In this context, ΔP, referred to as "Probability Discrepancy", is utilized to assess the intensity of the sentiment inclination prediction made by f_{sc} for X. A larger ΔP value indicates a more pronounced sentiment inclination in X, reflecting the model's confidence in its sentiment classification.

4.2.3. Mask Sequence Interpretation Score

In the SCE task, for a given input X with labels Y, there exists an inverse sequence $\bar{Y} = \{1 - y_1, 1 - y_2, \cdots, 1 - y_n\}$. As defined by the task, if the token x_i in X is identified as an SC within X, then the corresponding label y_i is assigned a value of 1; if not, y_i is set to 0. This principle suggests that masking all tokens x_i in X for which $y_i = 1$, resulting in a masked input $X^{\bar{Y}}$, would hinder the sentiment classification model's ability to accurately

determine the sentiment inclination of $X^{\bar{Y}}$. Conversely, retaining only the tokens in X, where $y_i = 1$, and masking those with $y_i = 0$, to create a new input X^Y, should enable the sentiment classification model to predict its sentiment inclination effectively.

In typical scenarios, to obtain X^Y, it is necessary to replace tokens x_i in X corresponding to $y_i = 0$ in Y with a meaningless symbol like [MASK]. However, BERT provides a more straightforward solution for us. By using Y as the attention mask directly input into BERT, it automatically disregards tokens x_i in X corresponding to $y_i = 0$ in Y.

Therefore, for X, when using Y as the mask sequence, we obtain:

$$(p_{c0}^Y, p_{c1}^Y) = f_{\text{sc}}(X, Y), \tag{7}$$

yielding the sentiment category prediction:

$$C^Y = \arg\max(p_{c0}^Y, p_{c1}^Y), \tag{8}$$

and calculating the Probability Discrepancy as:

$$\Delta P^Y = |p_{c0}^Y - p_{c1}^Y|. \tag{9}$$

Similarly, when using the inverse sequence $\bar{Y}$ as the mask sequence, we can determine:

$$(p_{c0}^{\bar{Y}}, p_{c1}^{\bar{Y}}) = f_{\text{sc}}(X, \bar{Y}), \tag{10}$$

with the corresponding sentiment category determined by:

$$C^{\bar{Y}} = \arg\max(p_{c0}^{\bar{Y}}, p_{c1}^{\bar{Y}}), \tag{11}$$

and the Probability Discrepancy for $X^{\bar{Y}}$ calculated as:

$$\Delta P^{\bar{Y}} = |p_{c0}^{\bar{Y}} - p_{c1}^{\bar{Y}}|. \tag{12}$$

When selecting a candidate sequence Y as the pseudo label, the ideal scenario aims to maximize ΔP^Y while ensuring that $C^Y = C$, and simultaneously minimize $\Delta P^{\bar{Y}}$. However, this approach might lead to an extreme case where all elements of Y are set to 1 and all elements of $\bar{Y}$ are set to 0. In such a scenario, X^Y would be identical to X, and $X^{\bar{Y}}$ would contain no informative content, which, while adhering to the principle, is not desirable for effective sentiment cue extraction. To circumvent this issue, it is preferable to have as few elements labeled as 1 in Y as possible. To achieve this balance, we introduce the Ratio of Cue Tokens (RCT), calculated as follows:

$$RCT = \frac{\sum(Y)}{n} \tag{13}$$

where $\sum(Y)$ represents the number of elements valued at 1 in Y, and n denotes the total number of elements in Y.

Moreover, within X, there may be tokens that inversely affect the prediction of X's sentiment inclination. Such tokens might cause f_{sc} to predict the sentiment category of $X^{\bar{Y}}$ as being entirely opposite to that of X. In these situations, it is desirable for $\Delta P^{\bar{Y}}$ to be as large as possible to reflect a clear differentiation in sentiment inclination.

Taking into consideration the principles mentioned above, we propose an evaluation metric named the Mask Sequence Interpretation Score (MSIS) as follows:

$$MSIS = \begin{cases} \dfrac{\Delta P^{Y}}{\Delta P^{\bar{Y}} \cdot RCT^2}, & C = C^{Y} \cap \Delta P^{\bar{Y}} < \Delta P^{Y} \\ \dfrac{\Delta P^{Y} \cdot \Delta P^{\bar{Y}}}{RCT^2}, & C = C^{Y} \neq C^{\bar{Y}} \cap \Delta P^{\bar{Y}} \geq \Delta P^{Y} \\ 0, & Others \end{cases} \tag{14}$$

Algorithm 2 outlines the procedure for evaluating the MSIS for a given candidate sequence Y^c associated with an instance X.

Algorithm 2 Evaluating the candidate sequence Y^c

Require: instance X, the sentiment category of the instance C, length of the instance n, well trained sentiment classification model f_{sc}, candidate sequence $Y^c = \{y_1^c, y_2^c, \cdots, y_n^c\}$

$\bar{Y}^c \leftarrow \emptyset$
for $i \in [1, 2, \ldots, n]$ **do**
    Add $1 - y_i^c$ to $\bar{Y}^c$
end for
$(p_{c0}^{Y^c}, p_{c1}^{Y^c}) = f_{\text{sc}}(X, Y^c)$, $(p_{c0}^{\bar{Y}^c}, p_{c1}^{\bar{Y}^c}) = f_{\text{sc}}(X, \bar{Y}^c)$
$C^{Y^c} = \arg\max(p_{c0}^{Y^c}, p_{c1}^{Y^c})$, $C^{\bar{Y}^c} = \arg\max(p_{c0}^{\bar{Y}^c}, p_{c1}^{\bar{Y}^c})$
$\Delta P^{Y^c} = |p_{c0}^{Y^c} - p_{c1}^{Y^c}|$, $\Delta P^{\bar{Y}^c} = |p_{c0}^{\bar{Y}^c} - p_{c1}^{\bar{Y}^c}|$
$RCT_{Y^c} \leftarrow RCT(Y^c, n)$ ▷ Refer to Equation (13)
$MSIS_{Y^c} \leftarrow MSIS(C, C^{Y}, C^{\bar{Y}^c}, \Delta P^{Y^c}, \Delta P^{\bar{Y}^c})$ ▷ Refer to Equation (14)
return $MSIS_{Y^c}$

The process begins by creating an inverse sequence $\bar{Y}^c$, which serves as a complement to Y^c by inverting the binary values. This step ensures that we can compare the effects of including versus excluding specific tokens identified as sentiment cues on the predictions of the sentiment classification model.

Next, the algorithm employs f_{sc} to calculate the probabilities of X belonging to each sentiment category, both with and without the sentiment cues as indicated by Y^c and $\bar{Y}^c$, respectively. These probabilities allow the computation of the Probability Discrepancy (ΔP) for both sequences, offering insight into the decisiveness of the sentiment classification under different conditions.

The RCT for Y^c is then calculated, providing a measure of the proportion of tokens in X identified as sentiment cues by Y^c. This ratio is crucial for ensuring that the selection of sentiment cues is both significant and minimal, avoiding over-representation of cues.

Finally, the MSIS for Y^c is determined based on the sentiment category predictions and Probability Discrepancies for both Y^c and its inverse.

4.2.4. Process of Selecting Pseudo Label for X

Algorithm 3 details the comprehensive process for generating a pseudo label for an instance X. This process involves evaluating multiple candidate sequences generated under various sampling ratios, each with the aim of identifying the sequence that best represents the sentiment cues within X. The algorithm utilizes a well-trained sentiment classification model f_{sc} to calculate the MSIS for each candidate sequence, ultimately selecting the sequence with the highest MSIS as the pseudo label for X.

It should be noted that in order to ensure the RCT of the candidate sequences obtained through sampling is as uniform as possible, covering different instances, we will uniformly select several decimals between 0 and 1 to serve as sampling ratios.

By employing this algorithm for all instances in the dataset, a collection of pseudo labels is generated, forming a dataset that can be used to train the SCE model.

Algorithm 3 Process of selecting pseudo label for X

Require: instance X, the sentiment category of the instance C, length of the instance n, well trained sentiment classification model f_{sc}, set of sampling ratios $\{p_1, p_2, \ldots, p_k\}$, sampling number m

$Y \leftarrow \emptyset$ ▷ Initialize the pseudo label ad a empty set
$MSIS_Y \leftarrow 0$ ▷ Initialize $MSIS$ of Y with 0
for $p \in \{p_1, p_2, \ldots, p_k\}$ **do**
  for $i \in [1, 2, \ldots, m]$ **do**
    $Y^c \leftarrow$ Generate a candidate sequence with X and p ▷ Refer to Algorithm 1
    $MSIS_{Y^c} \leftarrow$ Evaluate Y^c ▷ Refer to Algorithm 2
    if $MSIS_{Y^c} > MSIS_Y$ **then**
      $Y \leftarrow Y^c$ ▷ Update Pseudo Label Y if a higher score is achieved
      $MSIS_Y \leftarrow MSIS_{Y^c}$ ▷ Update the score accordingly
    end if
  end for
end for
return Y

4.3. Sentiment Cue Extraction Model

Our SS-SCE approach conceptualizes the SCE task as a sequence labeling problem. This requires performing a binary classification for each token x_i within the input X. Consequently, the architecture of the SCE model is highly analogous to that of the sentiment classification model, with a key distinction: while the sentiment classification model focuses on classifying the [CLS] token to infer the overall sentiment of the input, the SCE model extends this classification to all tokens within X. Therefore, the SCE model can be formalized as follows:

$$f_{\text{sce}}(X) = \text{softmax}(\text{FC}^{768\times 2}(\text{BERT}(X))) \tag{15}$$

where $\text{BERT}(X)$ produces a sequence of 768-dimensional vector representations for each token in X. The fully connected layer, denoted as $\text{FC}^{768\times 2}$, maps each 768-dimensional vector to a 2-dimensional output, corresponding to the binary classification for sentiment cue detection. The softmax function is applied to these 2-dimensional vectors, yielding a probability distribution over two classes (cue vs. non-cue) for each token in X.

After generating pseudo labels for each instance X in the train set, these labels will be utilized as the ground truth for training the SCE model, f_{sce}. It is important to note that each X is augmented with [CLS] and [SEP] tokens at the beginning and end, respectively. While these tokens are essential for BERT's processing, they should not be overlooked by f_{sce}. Consequently, their corresponding labels in the pseudo label sequence are fixed to 1. However, we do not consider these specific tokens ([CLS] and [SEP]) as sentiment cues.

5. Experiments

5.1. Dataset

To rigorously evaluate the methodology proposed in this paper, we perform experiments using the IMDb [18] and SST-2 [19] datasets, both of which are sentiment classification datasets composed of English movie reviews. It is essential to note that BERT, the underlying model, is limited to processing sequences of a maximum of 512 tokens. Given that the IMDb dataset contains numerous instances exceeding this token limit, we selectively use instances with a length not surpassing 512 tokens for our experimental data.

Furthermore, we meticulously curate a subset of review data from the Yelp (https://www.yelp.com/dataset (accessed on 19 March 2024)) website. From the original Yelp dataset, we extract the top 14,000 reviews with the highest ratings and the bottom 14,000 reviews with the lowest ratings. After random swab, this dataset is divided into 20,000 reviews for training, 4000 for validation, and 4000 for testing.

To further extend the scope of our evaluation and validate the versatility of our methodology across different languages, we have incorporated the ChnSentiCorp dataset (https://aistudio.baidu.com/datasetdetail/10320 (accessed on 19 March 2024)). This dataset consists of Chinese-language hotel reviews, providing an opportunity to assess our model's performance in a non-English context.

The statistical characteristics of these four datasets are succinctly summarized in Table 1.

Table 1. This table shows the sizes of the training, validation, and test sets for four different datasets.

Dataset	Training Set Size	Validation Set Size	Testing Set Size
SST-2	60,000	7349	872
Yelp	20,000	4000	4000
IMDb	17,008	4310	21,500
ChnSentiCorp	9146	1200	1200

5.2. Experimental Setup

We initially train sentiment classification models for each of the three datasets. Then, for each instance X in the training and evaluation sets of each dataset, we generate candidate sequences using Algorithm 1. Since it is not possible to predict the proportion of tokens in X that are sentiment cues, denoted as p, we test different values of p from the set {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7}.

For both the sentiment classification model and the sentiment cue extraction model, we employed the bert-base-uncased (https://huggingface.co/bert-base-uncased (accessed on 19 March 2024)) architecture as our encoder and employed softmax [42] as the decoder. The learning rate is set to 0.00001, and we use the Adam optimizer with the applied cross-entropy loss function.

In the training phase of the SCE model, we primarily use cross-entropy loss as the main evaluation metric. We systematically selected the model parameters that achieved the minimum loss in the validation set as the final parameters of the model.

All computations are performed on a Tesla V100-SXM2-16GB GPU manufactured by NVIDIA Corporation, headquartered in Santa Clara, CA, USA. Due to variations in the maximum length of samples in the three datasets and limitations in GPU memory, the number of candidate sequences generated per run differed. Specifically, we generated 100 mask sequences for SST-2 and Yelp in a single run, while for IMDb and ChnSentiCorp, we could only generate 10 mask sequences per run.

In training the classification and SCE models, we adjust the batch size based on the dataset to optimize resource utilization and training efficiency. For SST-2 and Yelp, the batch size is set to 32, accommodating a larger number of instances per training step due to their relatively shorter text lengths. In contrast, for IMDb and ChnSentiCorp, which consist of longer text instances, the batch size is set to 8.

5.3. Evaluation Metrics

To assess the effectiveness of our SS-SCE approach, evaluations are conducted from both quantitative and qualitative perspectives.

5.4. Results and Analysis

5.4.1. Computational Efficiency of Monte Carlo Sampling

To assess the computational demands of our method, we performed Monte Carlo Sampling in the training and validation sets of the SST-2, Yelp, IMDb, and ChnSentiCorp datasets. We generated a fixed number of 10,000 candidate sequences for each instance.

To elucidate the computational efficiency of our Monte Carlo Sampling process, detailed statistics are presented in Table 2. This table shows the Average Time Per Sampling (ATPS) in milliseconds (ms) and the Average Time for the Optimal Mask Sequence (ATOMS) in seconds (s) for each dataset.

Table 2. This table presents the average length of each instance in four datasets, the average time consumed per sampling, and the average time to obtain the optimal mask sequence.

Metric	SST-2	Yelp	IMDb	ChnSentiCorp
Average Length	32.02	79.57	265.02	90.49
ATPS (ms)	0.70	3.71	9.84	6.03
ATOMS (s)	1.49	19.84	49.68	30.27

As evident from Table 2, the time required for generating a single sample increases with the length of the text, as does the average time to complete the sampling process for obtaining the optimal mask sequence. This outcome indicates that our approach is relatively less efficient for longer texts. As the length of the text increases, more time is required to complete the sampling process.

5.4.2. Main Performance Evaluation

Given the absence of annotated data, it is challenging to directly apply traditional sequence labeling evaluation metrics to assess SS-SCE. According to the definition of the SCE task, the sentiment orientation of X^Y, obtained by masking X with the pseudo-label Y, should align with that of X. Therefore, we can indirectly evaluate SS-SCE by comparing the performance metrics of instances in the test set when using X as input versus using X^Y as input in the sentiment classification model. Specifically, we calculate the accuracy, precision, recall, and F1 scores for the test set when using X and X^Y as inputs, respectively, and measure the performance loss caused by using X^Y as input.

Additionally, to statistically assess the impact of our SS-SCE method on the performance of sentiment classification, we conduct a t-test comparing the predictions made by the sentiment classification model for both the original input X and the input with extracted sentiment cues X^Y. The null hypothesis (H0) posits that the SS-SCE method does not significantly reduce the performance metrics of sentiment classification compared to the original input X. The alternative hypothesis (H1), on the other hand, suggests a significant reduction in these performance metrics, which would indicate an effect of the SS-SCE method. We set the confidence level for this test at 0.01, meaning a p-value less than 0.01 is required to reject the null hypothesis. Rejecting H0 would imply that the SS-SCE method significantly impacts the performance of the model, whereas failing to reject H0 would suggest that the SS-SCE method can extract sentiment cues without substantially compromising classification accuracy.

However, relying solely on this is not sufficient, as there could be special cases where all values of Y are 1, leading to $X^Y = X$. To avoid this scenario, we also evaluate using RCT, which is the proportion of sentiment cues extracted by SS-SCE relative to the original input.

To demonstrate the effectiveness and detailed impact of SS-SCE on sentiment classification accuracy, including any performance loss, Table 3 offers a comprehensive comparison. This table contrasts the performance metrics—accuracy, precision, recall, and F1 scores—for the original input (X) and the input with extracted sentiment cues (X^Y), across various datasets. It quantifies the performance loss incurred using X^Y as input and includes RCT to indicate the proportion of sentiment cues identified. Additionally, the table details the results of the t-test, providing statistical insight into the significance of the differences observed between the performances of X and X^Y.

For the SST-2 dataset, compared to the original input X, the prediction results using X^Y as input show a decrease across all major metrics, but the decrease is within 0.1, and the p-value from the t-test is greater than 0.01. This indicates that our SS-SCE method effectively extracts the majority of sentiment cues from the SST-2 dataset, albeit with some minor losses. The Ratio of Cue Tokens (RCT) is 0.1682, which means that tokens identified as sentiment cues by SS-SCE constitute 16.82% of the total in the SST-2 test set. This performance suggests that SS-SCE can extract sentiment cues without significantly compromising the accuracy of sentiment classification.

Table 3. This table displays accuracy, precision, recall, F1 scores, and performance loss for original (X) versus cue-extracted (X^Y) inputs across SST-2, Yelp, IMDb, and ChnSentiCorp datasets. RCT values and *t*-test results are also included to assess the extraction's effectiveness.

	SST-2			**Yelp**			**IMDb**			**ChnSentiCorp**		
Metric	X	X^Y	**Loss**	X	X^Y	**Loss**	X	X^Y	**Loss**	X	X^Y	**Loss**
Accuracy	0.9300	0.8719	0.0585	0.9885	0.9748	0.0138	0.9328	0.8798	0.0531	0.9369	0.8367	0.1002
Precision	0.9379	0.9072	0.0307	0.9876	0.9723	0.0153	0.9305	0.8333	0.0971	0.9387	0.7940	0.1448
Recall	0.9182	0.8224	0.0958	0.9895	0.9776	0.0120	0.9359	0.9501	−0.014	0.9372	0.9174	0.0198
F1	0.9280	0.8627	0.0652	0.9886	0.9749	0.0136	0.9332	0.8879	0.0453	0.9380	0.8512	0.0867
RCT	-	0.1682	-	-	0.3795	-	-	0.2858	-	-	0.3148	-
$p_{t\text{-test}}$		>0.01			>0.01			<0.01			<0.01	

For the Yelp dataset, the decline in metrics for X^Y is notably subtle, with all reductions less than 0.02. Furthermore, the *t*-test results reveal no significant differences in metrics between X^Y and X within this dataset. However, the relatively higher RCT indicates that SS-SCE may employ a more lenient criterion when extracting sentiment cues on the Yelp dataset.

Regarding the IMDb dataset, the results with X^Y as input show the highest decrease in accuracy and precision among the three datasets, while the impact on recall is the opposite, even surpassing the performance using X as input. This phenomenon could be attributed to longer texts containing more distracting information, which our SS-SCE method is adept at effectively filtering out. The relatively lower RCT value among the three datasets corroborates this observation. Furthermore, the higher recall rate for X^Y suggests that SS-SCE effectively extracts sentiment cues from X, improving the model's ability to identify relevant sentiment information. The *p*-value of the *t*-test being less than 0.01 indicates a significant difference in the sentiment classification results between X and X^Y. Coupled with the increase in recall, we consider this impact positive.

On the ChnSentiCorp dataset, the RCT is 0.3148, indicating that 31.48% of tokens in X were extracted as sentiment cues. In this context, the loss in recall is minimal, only 0.0198, suggesting that SS-SCE likely captures the majority of sentiment cues. However, compared to the IMDb dataset, the performance metrics on ChnSentiCorp are noticeably poorer. This indicates that our SS-SCE method may have certain limitations when processing Chinese data. This could be due to BERT's character-level processing of Chinese, whereas Chinese semantics are typically conveyed at the word level. Therefore, during the sampling process, words might be segmented into characters that fail to express complete semantics, thereby affecting the model's performance.

In summary, the experimental results prove that our SS-SCE method achieves good results on English datasets, especially on datasets with longer text lengths, where the extraction of sentiment cues is more effective. However, there are clear deficiencies in the Chinese dataset. In future research, we will consider addressing the issues encountered in the Chinese dataset.

5.4.3. Model Generalization Tests

To ascertain the adaptability and generalizability of our proposed method, we conduct cross-testing on three English datasets. Specifically, this involves using the model trained on each dataset to test the other two datasets. Additionally, we combine the datasets generated by the SS-SCE method from all three datasets to train a single sentiment cue extraction model, which is then tested on all three datasets.

Additionally, we merge the datasets sampled from the three English datasets to train collectively and conduct tests on each dataset individually. For the amalgamated dataset, we use the term "combined" to denote it.

In the cross-testing, we continue to use the same evaluation metrics as those presented in Table 3. It is noted that we use subscripts to denote the training dataset of the sentiment

extraction model. For example, $X^{Y}_{SST\text{-}2}$ represents the X^{Y} generated by the sentiment cue extraction model trained on the SST-2 dataset.

As shown in Table 4, when models trained on Yelp and IMDb datasets are tested on SST-2, they show a notable performance decline, particularly in accuracy and recall. The most pronounced drop is observed in the model trained on IMDb, with a 16.97% decrease in accuracy. This can be attributed to the disparity in text length and complexity between IMDb and SST-2 datasets. Although the precision of the IMDb-trained model remained relatively stable, indicating a consistent ability to identify true positives, the substantial decrease in recall, especially for this model, suggests challenges in capturing the full range of sentiment cues in shorter SST-2 texts.

Moreover, the recall of $X^{Y}_{combined}$ shows an improvement, indicating that the incorporation of Yelp and IMDb enhances the ability to extract sentiment cues. However, this integration also introduces additional information, which adversely affects the accuracy and precision of the model.

Table 4. This table shows the test results on the SST-2 dataset for models trained on SST-2, Yelp, IMDb, and the combined dataset.

Metric	X	$X^{Y}_{SST\text{-}2}$	$loss_{SST\text{-}2}$	X^{Y}_{Yelp}	$loss_{Yelp}$	X^{Y}_{IMDb}	$loss_{IMDb}$	$X^{Y}_{combined}$	$loss_{combined}$
Accuracy	0.9300	0.8716	0.0585	0.8234	0.1067	0.7603	0.1697	0.8039	0.1261
Precision	0.9379	0.9072	0.0307	0.8549	0.0830	0.9195	0.0184	0.7495	0.1884
Recall	0.9182	0.8224	0.0958	0.7710	0.1472	0.5607	0.3575	0.9234	-0.005
F1	0.9280	0.8627	0.0652	0.8108	0.1172	0.6967	0.2313	0.8274	0.1006
RCT	-	0.1682	-	0.1256	-	0.1012	-	0.1489	-

In Table 5, the adaptability of the models to the Yelp dataset is more promising. The decrease in accuracy and the F1 score is less severe compared to their performance in the SST-2 dataset. This implies that the models are better equipped to handle the moderate text lengths and complexity of Yelp reviews. However, the performance of the model trained on the IMDb dataset is significantly poorer, especially in terms of recall. Similarly, the model trained on the combined dataset also experiences some degree of performance degradation, which may be attributed to the influence of the IMDb dataset.

Table 5. This table shows the test results on the Yelp dataset for models trained on SST-2, Yelp, IMDb, and the combined dataset.

Metric	X	$X^{Y}_{SST\text{-}2}$	$loss_{SST\text{-}2}$	X^{Y}_{Yelp}	$loss_{Yelp}$	X^{Y}_{IMDb}	$loss_{IMDb}$	$X^{Y}_{combined}$	$loss_{combined}$
Accuracy	0.9885	0.9650	0.0235	0.9748	0.0138	0.9260	0.0625	0.9655	0.0230
Precision	0.9876	0.9722	0.0154	0.9723	0.0153	0.9853	0.0023	0.9626	0.0250
Recall	0.9895	0.9577	0.0319	0.9776	0.0120	0.8655	0.1240	0.9484	0.0411
F1	0.9886	0.9649	0.0237	0.9749	0.0136	0.9215	0.0670	0.9554	0.0332
RCT	-	0.1397	-	0.3795	-	0.2698	-	0.2773	-

Table 6 indicates that models trained on shorter text datasets, such as SST-2 and Yelp, also perform effectively on the IMDb dataset, positively influencing accuracy. However, there is a negative impact on recall. This suggests that while the models retain their ability to correctly identify true positives in the context of longer texts, their capacity to capture the full range of sentiment cues across the broader dataset is somewhat diminished.

These results indicate that while models trained on shorter texts, such as SST-2, exhibit relatively better generalization capabilities across datasets, models trained on datasets with longer texts, such as IMDb, show limited adaptability to shorter texts. Additionally, when conducting cross-dataset experiments, training on a combination of multiple datasets, although generally not outperforming training on their own respective datasets, tends to yield better results than training on any single, different dataset. This implies that when

extending SS-SCE to new data, considering training across multiple similar datasets could enhance model performance. This strategy may leverage the diverse characteristics of each dataset to build a more robust and adaptable model.

Table 6. This table shows the test results on the IMDb dataset for models trained on SST-2, Yelp, IMDb, and the combined dataset.

Metric	X	$X^{Y}_{SST\text{-}2}$	$loss_{SST\text{-}2}$	X^{Y}_{Yelp}	$loss_{Yelp}$	X^{Y}_{IMDb}	$loss_{IMDb}$	$X^{Y}_{combined}$	$loss_{combined}$
Accuracy	0.9328	0.8925	0.0403	0.8709	0.0620	0.8798	0.0531	0.8934	0.0394
Precision	0.9305	0.9205	0.0099	0.8564	0.0761	0.8333	0.0971	0.9327	−0.0020
Recall	0.9359	0.8598	0.0762	0.8918	0.0441	0.9501	−0.014	0.8474	0.0921
F1	0.9332	0.8891	0.0441	0.8737	0.0594	0.8879	0.0453	0.8880	0.0452
RCT	-	0.1153	-	0.1186	-	0.2858	-	0.2478	-

5.5. Case Study: Comparing SS-SCE with Established Interpretability Methods

To evaluate the unique contributions and effectiveness of SS-SCE, we perform a comparative analysis with established interpretability methods in text classification, including LIME [5], LIG [43], OCC [44], SVS [45], and LDS [46].

For this comparison, we use the Thermostat tool (https://github.com/DFKI-NLP/thermostat (accessed on 19 March 2024)) [47], which integrates state-of-the-art interpretability methods, offering a unified platform for analysis. This tool allowed us to apply these methods in a standardized way, ensuring a fair and consistent comparison between different interpretability approaches.

Our analysis aimed not to compare SS-SCE directly with these methods, but to showcase how SS-SCE's focused approach on sentiment cues provides a different, potentially more nuanced perspective in understanding model decisions, especially in the context of sentiment analysis.

Using Thermostat, we applied interpretability models trained on various datasets, such as IMDb with pre-trained language models such as BERT and ALBERT [48]. For a fair comparison, we chose the interpretability model trained with BERT on the IMDb dataset. To facilitate a comparison with SOTA methods, we manually annotated two selected instances, a positive and a negative, from the IMDb test set. We then calculated the precision, recall, and F1 score for each method's sentiment cue extraction on these annotated instances. The results of this comparative analysis are presented in Tables 7 and 8.

Table 7 shows that the SS-SCE models, particularly SS-SCE$_{SST\text{-}2}$, demonstrate superior performance in extracting sentiment cues from the positive text instance when compared with SOTA interpretability methods, SS-SCE$_{SST\text{-}2}$ achieved the highest precision of 0.7778, recall of 0.8235, and F1 score of 0.8000, indicating a robust capability in accurately identifying and recalling relevant sentiment cues.

The SS-SCE models trained on Yelp and IMDb datasets showed varying degrees of effectiveness, with SS-SCE$_{Yelp}$ displaying moderate performance and SS-SCE$_{IMDb}$, showing decent accuracy but lower effectiveness compared to SS-SCE$_{SST\text{-}2}$. This variation suggests the influence of training data characteristics on the model's performance.

In contrast, the standard interpretability methods, while useful in their own right, exhibited lower performance metrics in comparison. LIME, LIG, OCC, SVS, and LDS demonstrated lower precision, recall, and F1 scores, indicating a potential limitation in their ability to capture the nuanced sentiment cues as effectively as the SS-SCE approach.

Table 8 presents the performance of different interpretability methods in extracting sentiment cues from the negative text instance. The results indicate that the SS-SCE models, particularly SS-SCE$_{IMDb}$ and SS-SCE$_{SST-2}$, perform effectively in this context, albeit with some variations in precision and recall.

Table 7. This table shows the performance comparison of our Self-Supervised Sentiment Cue Extraction (SS-SCE) model trained on three datasets with SOTA interpretability methods on a positive instance.

Method	Result	Precision	Recall	F1
human	This is a **great horror movie**. **Great plot**. And a person with a fear of midgets will definately **love** the evil midget! This is a **must see** for **any horror fan**. Finally a lower budget movie with **decent effects** and a **great cast**! **Highly recommended**.	-	-	-
SS-SCE$_{SST\text{-}2}$	This is a **great** horror **movie**. **Great plot**. And a person with a fear of midgets will definately **love** the evil midget! **This** is a **must see** for **any** horror **fan**. Finally a lower budget **movie with decent** effects **and** a **great cast**! **Highly recommended**.	**0.7778**	**0.8235**	**0.8000**
SS-SCE$_{Yelp}$	**This** is a **great** horror movie. **Great plot**. **And** a person with a fear of midgets will definately love the evil midget! **This** is **a must see for any** horror fan. **Finally a** lower budget movie **with decent** effects **and a great cast**! **Highly recommended**.	0.5556	0.6250	0.5882
SS-SCE$_{IMDb}$	**This is a great** horror movie**. Great** plot. And a person with a fear of midgets will definately **love** the evil midget! **This is a must** see for any horror fan. Finally a lower budget movie with **decent** effects and a **great** cast! **Highly recommended**.	0.6154	0.4706	0.5333
LIME	This **is** a great horror **movie**. **Great** plot. And a **person** with a fear of midgets will definately love the **evil midget**! This is a **must see** for any horror **fan**. **Finally** a **lower budget movie** with decent effects and a great cast! **Highly** recommended.	0.4286	0.3529	0.3871
LIG	This **is a great** horror **movie**. **Great** plot. **And** a person **with** a fear of **midgets** will definately **love** the evil midget! This **is** a **must** see for any horror **fan**. **Finally** a lower budget movie **with** decent effects and **a great** cast! **Highly recommended**.	0.5000	0.5294	0.5143
OCC	**This** is a great horror movie. Great **plot**. And a person with **a fear of midgets** will definately **love the evil midget**! This is a must see **for any horror** fan. Finally **a lower budget movie with decent effects** and **a great cast! Highly** recommended.	0.3600	0.5294	0.4286
SVS	**This is a great** horror **movie**. **Great plot**. **And** a **person** with a **fear** of midgets will definately **love the** evil midget**!** This is **a must** see for any horror **fan**. Finally a lower budget **movie** with decent effects **and** a **great** cast**! Highly recommended**.	0.4545	0.5882	0.5128
LDS	This is a great **horror movie**. Great plot. And a person with a fear of **midgets will definately love the evil** midget**!** This is a must see for any **horror** fan**.** Finally **a lower budget movie** with decent **effects** and a great **cast**! Highly **recommended**.	0.4118	0.4118	0.4118

The bold tokens represent the extracted sentiment cues.

SS-SCE$_{IMDb}$ achieved the highest precision (0.8571), reflecting its strong ability to accurately identify relevant negative sentiment cues. However, its recall (0.3750) is relatively lower, suggesting that, while it is precise, it may miss some relevant cues. Conversely, SS-SCE$_{SST-2}$, with a recall of 0.5000, demonstrates a balanced performance with a precision of 0.5714 and an F1 score of 0.5333. This balance indicates its ability to capture a broader range of relevant cues while maintaining accuracy.

SS-SCE$_{Yelp}$, despite having the highest precision (0.8333), shows a lower recall (0.3125), indicating a tendency to be very selective in cue extraction, which may lead to missing some pertinent sentiment indicators.

In comparison, traditional interpretability methods show lower performance in both precision and recall. LIME and LDS, in particular, demonstrate limited effectiveness in accurately identifying negative sentiment cues. The lower performance of these methods may be attributed to their design, which might not be as fine-tuned for sentiment cue extraction as the SS-SCE approach.

Overall, the comparative analysis of sentiment cue extraction presented in Tables 7 and 8 demonstrates the robustness and versatility of the SS-SCE models across both positive and negative text instances. The SS-SCE models, especially SS-SCE$_{SST-2}$, consistently exhibit a balanced performance in terms of precision and recall, highlighting their ability to accurately and comprehensively extract sentiment cues. This is particularly evident in SS-SCE$_{SST-2}$, which shows strong performance in both positive and negative contexts. While SS-SCE$_{IMDb}$ and SS-SCE$_{Yelp}$ demonstrate higher precision in specific in-

stances, they sometimes compromise on recall, indicating a more selective extraction of cues. In comparison to the SOTA interpretability methods, the SS-SCE approach stands out for its enhanced capability to identify both explicit and subtle sentiment indicators.

Table 8. This table shows the performance comparison of our Self-Supervised Sentiment Cue Extraction (SS-SCE) model trained on three datasets with SOTA interpretability methods on a negative instance.

Method	Result	Precision	Recall	F1
human	**Unfortunately**, this movie is **absolutely terrible**. It's **not even laughably bad**, **just plain bad**. The actors do their best with what is **the cheesiest script ever**. **How scary** can a movie be when the climax actually involves a roomful of millions of styrofoam peanuts?	-	-	-
SS-SCE$_{SST\text{-}2}$	**Unfortunately**, this movie is absolutely **terrible**. It's **not** even laughably **bad**, just plain **bad**. The actors **do** their **best** with what is the **cheesiest** script ever. **How scary** can a movie be when the climax actually **involves** a **roomful** of millions of **styrofoam peanuts**?	0.5714	**0.5000**	**0.5333**
SS-SCE$_{Yelp}$	**Unfortunately**, this movie is absolutely **terrible**. It's not even laughably **bad**, just plain **bad**. The actors do their best with what is the cheesiest script ever. How **scary** can a movie be when the climax actually involves a roomful of millions of styrofoam **peanuts**?	0.8333	0.3125	0.4545
SS-SCE$_{IMDb}$	**Unfortunately**, this movie is **absolutely terrible**. It's not even laughably **bad**, just plain **bad**. The actors do their **best** with what is the **cheesiest** script ever. How scary can a movie be when the climax actually involves a roomful of millions of styrofoam peanuts?	**0.8571**	0.3750	0.5217
LIME	**Unfortunately, this** movie is **absolutely terrible**. **It**'s not **even** laughably bad, just plain bad. The actors do their best with what is the **cheesiest** script ever. How scary can a **movie be** when **the climax** actually **involves a roomful** of millions of **styrofoam** peanuts?	0.3125	0.3125	0.3125
LIG	**Unfortunately, this** movie **is absolutely terrible**. It's **not** even laughably **bad,** just plain **bad.** The actors do their best with what is the cheesiest script ever. How scary can a **movie** be when the climax actually involves a roomful of millions of styrofoam peanuts?	0.5455	0.3750	0.4444
OCC	Unfortunately, **this movie is absolutely terrible.** It's not even laughably bad, just plain bad. The actors do their best with what is the cheesiest script ever. How scary can a movie be when the climax actually involves a roomful of millions of styrofoam **peanuts?**	0.2857	0.1250	0.1739
SVS	**Unfortunately**, this movie is absolutely **terrible**. It's not **even** laughably **bad**, **just** plain **bad**. The actors do their best with what is the cheesiest script ever. How scary can a movie be when the climax actually involves a roomful of millions of styrofoam peanuts**?**	**0.8571**	0.3750	0.5217
LDS	**Unfortunately,** this **movie** is absolutely **terrible**. It's not even laughably bad, just plain bad. The actors **do** their best with what is the **cheesiest** script ever**. How** scary can a **movie** be when the climax actually involves a roomful of millions of styrofoam peanuts?	0.5000	0.2500	0.3333

The bold tokens represent the extracted sentiment cues.

Simultaneously, it is important to note that our approach represents a global interpretability method, which significantly outperforms traditional techniques in terms of efficiency when applied to new data. This global perspective enables a comprehensive understanding of the model's decision-making process across various datasets and scenarios, rather than focusing on individual instances.

5.6. Ablation Study on MSIS

To validate the effectiveness and contribution of each component within the MSIS, we conduct an ablation study. This study systematically examines how the removal or alter-

ation of each MSIS component affects the overall performance of our SS-SCE framework. The components of MSIS are as follows.

Probability Discrepancy (PD): This component, denoted as ΔP^{Y}, assesses the clarity of sentiment cues within the candidate sequence. It ensures that elements marked with 1 in the candidate sequence effectively contribute to the sentiment classification model's decision-making process.

Inverse Probability Discrepancy (IPD): Represented as $\Delta P^{\bar{Y}}$, it evaluates the absence of sentiment cues within the inverse attention mask $\bar{Y}$.This ensures elements marked with 0 in Y do not contribute significantly to sentiment interpretation, emphasizing the specificity of extracted cues.

Ratio of Cue Tokens (RCT): This component aims to minimize the inclusion of irrelevant tokens in the candidate sequence, promoting a concise extraction of sentiment cues. It is calculated as the proportion of 1s in Y^{c}, with a higher RCT indicating a more focused extraction of sentiment cues.

The results of our ablation study are summarized in Tables 9–12 . Each row represents a variant of the MSIS, indicating the presence (+) or absence (-) of each component. Performance metrics include the accuracy, precision, recall, and F1 score of the sentiment cue extraction under each variant.

Table 9. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of SST-2 dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.8716	0.9072	0.8224	0.8627	0.1682
-	+	+	0.7879	0.7595	0.8536	0.8038	0.1405
+	-	+	0.7397	0.6764	0.9369	0.7856	0.0540
+	+	-	0.8807	0.8586	0.9167	0.8867	0.5214

Table 10. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of Yelp dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.9748	0.9723	0.9776	0.9749	0.3795
-	+	+	0.9635	0.9446	0.9844	0.9641	0.3928
+	-	+	0.8395	0.7614	0.9869	0.8596	0.0181
+	+	-	0.9838	0.9854	0.9819	0.9837	0.8910

Table 11. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of IMDb dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.8798	0.8333	0.9501	0.8879	0.2858
-	+	+	0.8654	0.9529	0.7728	0.8535	0.1542
+	-	+	0.6207	0.9586	0.2637	0.4136	0.0038
+	+	-	0.9261	0.9176	0.9385	0.9279	0.8547

Table 12. This table shows the ablation study results for Mask Sequence Interpretation Score (MSIS) components of ChnSentiCorp dataset.

PD	IPD	RCT	Accuracy	Precision	Recall	F1	RCT
+	+	+	0.8367	0.7940	0.9174	0.8512	0.3148
-	+	+	0.8746	0.8444	0.9240	0.8824	0.5042
+	-	+	0.6641	0.6392	0.7818	0.7033	0.0198
+	+	-	0.9234	0.9227	0.9273	0.9250	0.8676

As shown in Tables 9–12 , the ablation study systematically evaluates the contribution of each component within the MSIS on the SST-2, Yelp, IMDb, and ChnSentiCorp datasets. This study offers a nuanced understanding of how each element influences the framework's ability to extract and utilize sentiment cues.

Removing the PD component results in performance degradation across most metrics, particularly evident in the reduction of precision and the F1 score. This suggests that PD is crucial for identifying clear sentiment cues within the text, ensuring that the elements marked as sentiment cues in the candidate sequence contribute effectively to the decision-making process of the sentiment classification model. However, on the Chinese dataset, the performance after removing the PD component is slightly better than the overall performance with the complete MSIS. This may be attributed to the fact that the Chinese language processes characters as the smallest units, rather than words. It is important to note that while the removal of PD results in a decrease in RCT by 0.1894, the F1 score only drops by 0.0312, illustrating the effectiveness of our method.

The absence of the IPD leads to a significant decrease in recall and a noticeable drop in the RCT, indicating a diminished ability to exclude non-sentiment-related tokens from being marked as sentiment cues. This highlights the IPD's role in refining the specificity of extracted cues by ensuring that elements marked with 0 in Y do not significantly contribute to sentiment interpretation.

Removing RCT results in an improvement in sentiment classification performance but at the cost of a substantial increase in RCT. This implies that while the RCT component restricts the inclusion of irrelevant tokens in the candidate sequence, its absence leads to a wider selection of tokens as sentiment cues, including potentially irrelevant ones.

In summary, each component of the MSIS plays a vital role in the sentiment cue extraction process. PD ensures the clarity and relevance of cues, IPD enhances the specificity of cue extraction, and RCT promotes conciseness and focus. The ablation study demonstrates the delicate balance between these components, underscoring their collective contribution to the effectiveness of the SS-SCE framework.

6. Conclusions

In conclusion, our research introduces a novel self-supervised framework for sentiment cue extraction that significantly improves the interpretability of sentiment analysis models. Through meticulous identification and extraction of key linguistic elements that influence sentiment determination, our approach demystifies the decision-making process of sentiment analysis models, thereby fostering greater trust and understanding in these systems.

Our innovative use of Monte Carlo Sampling for efficient cue identification and the development of the Mask Sequence Interpretation Score (MSIS) metric to evaluate the extraction of sentiment cues represent substantial advances in the field of sentiment analysis. Importantly, our methodology extends beyond traditional local interpretability techniques, providing a global interpretability approach that enhances understanding across various instances and datasets. The application of our method in diverse datasets, such as SST-2, Yelp, IMDb, and ChnSentiCorp, demonstrates its effectiveness in extracting pertinent sentiment cues.

However, our study is not without its limitations. The computational demands of our approach, especially in handling longer texts, highlight the need for further optimization to enhance efficiency without sacrificing accuracy. Additionally, while our method shows promising results in extracting sentiment cues, the performance variability across different text lengths and complexities suggests room for improvement in the generalizability and adaptability of the model. Furthermore, when processing Chinese data, our method faces additional challenges. This is partly due to BERT's character-level processing of Chinese, whereas Chinese semantics are more accurately represented at the word level. Consequently, during sampling, words may be segmented into characters that fail to convey full semantics, affecting the model's performance. This aspect highlights the

importance of tailoring our approach to better accommodate the linguistic characteristics of Chinese, suggesting a direction for future research to improve the method's applicability and effectiveness in handling Chinese texts.

Looking forward, we see several avenues for future research. Enhancing the computational efficiency of our Monte Carlo Sampling process and exploring alternative sampling techniques could address current limitations in processing longer texts. Further refinement of the MSIS metric to better balance accuracy and interpretability could also produce improvements in sentiment cue extraction. Moreover, extending our framework to incorporate multimodal data (text, images, and videos) could offer a more holistic approach to sentiment analysis, reflecting the multifaceted nature of sentiment expression across various media. Then, addressing the specific challenges of processing Chinese data, such as adapting our approach to better capture the word-level semantics often lost in character-level processing, also constitutes a critical area for future exploration. This would not only improve the model's performance on Chinese texts but also enhance its applicability and effectiveness across linguistically diverse datasets.

Ultimately, our work contributes to the ongoing efforts to bridge the gap between advanced sentiment analysis techniques and their interpretability, aiming to create more transparent, reliable, and user-friendly NLP models. By emphasizing global interpretability, our approach offers a scalable and comprehensive solution for understanding complex sentiment analysis models. By continuing to refine and expand upon the foundations laid by this study, we anticipate contributing to the development of sentiment analysis models that are not only highly accurate but also thoroughly interpretable, ensuring their ethical and effective application in sensitive domains.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S., S.H. and X.H.; software, Y.S.; validation, Y.S.; formal analysis, Y.S., X.H. and Y.L.; investigation, Y.S.; resources, Y.S.; data curation, Y.S. and X.H.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S., S.H., X.H. and Y.L.; visualization, Y.S.; supervision, S.H.; project administration, Y.S.; funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Natural Science Foundation of China under Grant No. 71974187.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The research data can be found at https://drive.google.com/drive/folders/1tHogTUtHC5sqLCS2bCNYpYTJayQ-1WnS (accessed on 19 March 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, B. *Sentiment Analysis and Opinion Mining*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
2. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [CrossRef]
3. Wankhade, M.; Rao, A.C.S.; Kulkarni, C. A survey on sentiment analysis methods, applications, and challenges. *Artif. Intell. Rev.* **2022**, *55*, 5731–5780. [CrossRef]
4. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018; IEEE: New York, NY, USA, 2018; pp. 80–89.
5. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
6. Chiong, R.; Fan, Z.; Hu, Z.; Dhakal, S. A novel ensemble learning approach for stock market prediction based on sentiment analysis and the sliding window method. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 2613–2623. [CrossRef]
7. McCarthy, S.; Alaghband, G. Enhancing Financial Market Analysis and Prediction with Emotion Corpora and News Co-Occurrence Network. *J. Risk Financ. Manag.* **2023**, *16*, 226. [CrossRef]

8. Bharti, S.K.; Tratiya, P.; Gupta, R.K. Stock Market Price Prediction through News Sentiment Analysis & Ensemble Learning. In Proceedings of the 2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Odisha, India, 15–17 December 2022; IEEE: New York, NY, USA, 2022; pp. 1–5.
9. Greaves, F.; Ramirez-Cano, D.; Millett, C.; Darzi, A.; Donaldson, L. Use of sentiment analysis for capturing patient experience from free-text comments posted online. *J. Med. Int. Res.* **2013**, *15*, e2721. [CrossRef]
10. Nauta, M.; Trienes, J.; Pathak, S.; Nguyen, E.; Peters, M.; Schmitt, Y.; Schlötterer, J.; van Keulen, M.; Seifert, C. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Comput. Surv.* **2023**, *55*, 1–42. [CrossRef]
11. Madsen, A.; Reddy, S.; Chandar, S. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–42. [CrossRef]
12. Saeed, W.; Omlin, C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowl.-Based Syst.* **2023**, *263*, 110273. [CrossRef]
13. Yue, L.; Chen, W.; Li, X.; Zuo, W.; Yin, M. A survey of sentiment analysis in social media. *Knowl. Inf. Syst.* **2019**, *60*, 617–663. [CrossRef]
14. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
15. Liu, Y. Fine-tune BERT for extractive summarization. *arXiv* **2019**, arXiv:1903.10318.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
18. Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
19. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
20. Kaur, R.; Kautish, S. Multimodal sentiment analysis: A survey and comparison. In *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*; IGI Global: Hershey, PA, USA, 2022; pp. 1846–1870.
21. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
22. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
23. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
24. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
25. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (accessed on 19 March 2024).
26. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *Openai Blog* **2019**, *1*, 9.
27. Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; Hu, X. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data* **2023**, *epub ahead of print*. [CrossRef]
28. Tian, S.; Jin, Q.; Yeganova, L.; Lai, P.T.; Zhu, Q.; Chen, X.; Yang, Y.; Chen, Q.; Kim, W.; Comeau, D.C.; et al. Opportunities and challenges for ChatGPT and large language models in biomedicine and health. *Brief. Bioinform.* **2024**, *25*, bbad493. [CrossRef]
29. Chen, Y.P.; Lo, Y.H.; Lai, F.; Huang, C.H. Disease concept-embedding based on the self-supervised method for medical information extraction from electronic health records and disease retrieval: Algorithm development and validation study. *J. Med. Int. Res.* **2021**, *23*, e25113. [CrossRef]
30. Feldman, R.; Rosenfled, B.; Soderland, S.; Etzioni, O. Self-supervised relation extraction from the web. In Proceedings of the Foundations of Intelligent Systems: 16th International Symposium, ISMIS 2006, Bari, Italy, 27–29 September 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 755–764.
31. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *Stat* **2017**, *1050*, 2.
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
33. Wheeler, J.M.; Cohen, A.S.; Wang, S. A Comparison of Latent Semantic Analysis and Latent Dirichlet Allocation in Educational Measurement. *J. Educ. Behav. Stat.* **2023**, 10769986231209446. [CrossRef]

34. Xiong, J.; Li, F. Bilevel Topic Model-Based Multitask Learning for Constructed-Responses Multidimensional Automated Scoring and Interpretation. *Educ. Meas. Issues Pract.* **2023**, *42*, 42–61. [CrossRef]
35. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef] [PubMed]
36. Hammersley, J. *Monte Carlo Methods*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
37. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT press: Cambridge, MA, USA, 2016.
38. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
39. Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv* **2017**, arXiv:1701.02434.
40. Shapiro, A. Monte Carlo sampling methods. In *Handb. Oper. Res. Manag. Sci.* **2003**, *10*, 353–425.
41. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
42. Bridle, J.S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, Architectures and Applications*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 227–236.
43. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 3319–3328.
44. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part I 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
45. Castro, J.; Gómez, D.; Tejada, J. Polynomial calculation of the Shapley value based on sampling. *Comput. Oper. Res.* **2009**, *36*, 1726–1730. [CrossRef]
46. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
47. Feldhus, N.; Schwarzenberg, R.; Möller, S. Thermostat: A Large Collection of NLP Model Explanations and Analysis Tools. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Virtual Event, 7–11 November 2021; Adel, H.; Shi, S., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2021.
48. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.