# Supervised Gradual Machine Learning for Sentence-Level Sentiment Analysis

**Jing Su**[1], **Qun Chen**[1], **Yanyan Wang**[1], **Lijun Zhang**[1], **Wei Pan**[1], **and Zhanhuai Li**[1]

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an Shaanxi, 710072, China

## ABSTRACT

The task of Sentence-Level Sentiment Analysis (SLSA) aims to detect the general sentiment polarity of an entire sentence. Even though many Deep Neural Network (DNN) models have been proposed for SLSA, the task remains very challenging because the efficacy of these deep models depends on the i.i.d (Independent and Identically Distributed) assumption; but, in real scenarios, the distributions of training and target data are almost certainly different to some extent. To alleviate this limitation resulting from distribution misalignment, this paper proposes a supervised approach based on the non-i.i.d paradigm of Gradual Machine Learning (GML) for SLSA. Beginning with the labeled training instances, the proposed approach gradually labels target instances in the order of increasing hardness by iterative knowledge conveyance. It leverages DNNs for feature extraction to supervise gradual knowledge conveyance. Specifically, it trains a sentence-level polarity classifier, which can detect polarity similarity between close neighbors in a deep embedding space, and separately a binary semantic network, which can extract implicit polarity relations between two arbitrary instances. Then, it fulfills knowledge conveyance by modeling the detected relations as binary features in a factor graph. We have empirically evaluated the performance of the proposed approach on real benchmark workloads by a comparative study. Our extensive experiments show that it achieves the state-of-the-art performance across all the test workloads. Our work clearly demonstrates that by leveraging DNN for feature extraction, GML can easily outperform the pure DNN solutions.

## Introduction

Sentence-Level Sentiment Analysis (SLSA)[1] aims to analyze the opinions and emotions expressed in a sentence. Unlike Aspect-Level Sentiment Analysis[2], which reasons about the local sentiment polarity expressed towards a specific aspect, SLSA needs to detect the general sentiment orientation of an entire sentence. Helpful to informed decision making, SLSA is widely considered to be a very important task in natural language processing.

The state-of-the-art performance of SLSA has been achieved by various DNN models. Especially, over the last few years, it has been empirically shown that by semantic learning on large-scale corpus, the pre-trained models (e.g., Bert[3], Robert[4] and Xlnet[5]) can automatically capture implicit sentimental features, thus effectively improve the performance of SLSA. However, the efficacy of these DNN models depends on the i.i.d assumption; but in real scenarios, there may not be sufficient labeled training data, and even if provided with sufficient training data, the distributions of training data and target data are almost certainly different to some extent. We illustrate the challenge of SLSA by the running examples as shown in Figure 1, in which we indicate the sentimental importance of different words by color depth. It can be observed that the Bert model mislabels all of the example sentences. In $S_0$, the model misidentifies the first part, ***"one still works fine"***, as the more important part, and thus mispredicts the polarity of the sentence as ***positive***. In $S_2$, even though the model correctly identifies the second part as the more important one, it still mispredicts the polarity as ***positive***.

To alleviate the limitation resulting from distribution misalignment between training and target data, this paper proposes a supervised approach for SLSA based on the recently proposed paradigm of Gradual Machine Learning. In general, GML begins with some easy instances, which can be automatically labeled by the machine with high accuracy, and then gradually labels more challenging instances by iterative knowledge conveyance in a factor graph. It has been applied to the task of Aspect-Level Sentiment Analysis (ALSA)[6,7] as well as entity resolution[8]. It is noteworthy that the existing GML solutions are unsupervised. For instance, the existing GML solutions for aspect-level sentiment analysis leverage sentiment lexicons, explicit polarity relations indicated by discourse structures, and unsupervised DNNs to enable sentimental knowledge conveyance. On one hand, whether based on sentiment lexicons or unsupervised DNNs, the extracted features may be noisy, resulting in inaccurate and insufficient knowledge conveyance; on the other hand, the local aspect-level sentimental features may not indicate the sentimental polarity of an entire sentence. Therefore, they can not serve as effective medium for sentence-level sentiment analysis.

Specifically, the proposed approach uses binary polarity relations, which are the most direct way of knowledge conveyance, to enable supervised gradual learning. Since it has been widely recognized that Bert-based DNN models can capture sentimental

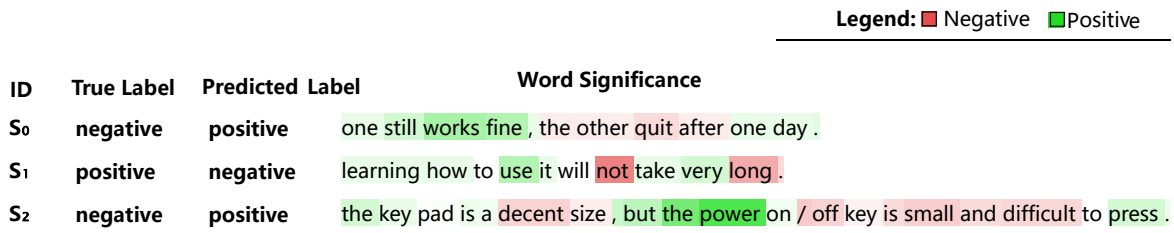


**Figure 1.** The illstrative examples of SLSA

features more accurately than manually crafted features (e.g., sentiment lexicons), the proposed approach leverages labeled training data to extract sentiment features by Bert-based models. Similar to the existing DNN models, it trains a sentence-level polarity classifier such that the sentences with similar polarities can be clustered within local neighborhood in a deep embedding space. To enable knowledge conveyance beyond local neighborhood, it also separately trains a semantic network to extract implicit polarity relations between two arbitrary sentences. All the extracted features are then modeled as binary factors in a factor graph to fulfill gradual learning.

The major contributions of this paper can be summarized as follows:

- We propose a supervised GML approach for SLSA, which can effectively exploit labeled training data to enable effective gradual learning;
- We present two types of DNN models to capture implicit sentimental features, and model them as binary factors in a factor graph to fulfill supervised knowledge conveyance for SLSA;
- We empirically validate the efficacy of the proposed solution on real benchmark workloads by a comparative study. Our extensive experiments have shown that it consistently achieves the state-of-the-art performance across all the test workloads.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 defines the task of SLSA and introduces the GML framework. Section 4 presents the proposed solution. Section 5 empirically evaluates the proposed solution. In Section 6, we conclude this paper with some thoughts on future work.

## Related work

The existing works on sentiment analysis can be broadly categorized into document-level, sentence-level and aspect-level. At the document level, the goal is to detect the sentiment polarity of an entire review. The state-of-the-art solutions have been built upon various deep neural networks, including contextual sentiment neural network[9], GLRNN[10], CNN-BiLSTM[11], SR-LSTM[12] and BAE[10]. Aspect-level sentiment analysis instead aims to detect sentiment polarities towards certain aspects of an entity. The most recent work, e.g., LCF-BERT[13], PTMs[14] and RGAT[15], were focused on how to leverage pre-trained language models (e.g, BERT). In this paper, we consider sentiment analysis on the sentence level. Unlike aspect-level sentiment analysis, sentence-level sentiment analysis needs to extract global sentiment features instead of local ones. Global features usually involve sentence structures, constituents and relations among different sub-parts. Therefore, compared with local features, global feature extraction is usually more challenging.

Early work on SLSA mainly focused on extracting different sentiment hints (e.g., n-gram, lexicon, pos and handcrafted rules) for SVM classifiers[16–19]. Unfortunately, these features are either sparse, covering only a few sentences, or not highly accurate. The advance of deep neural networks made feature engineering unnecessary for many natural language processing tasks, notably including sentiment analysis[20–22]. More recently, various attention-based neural networks have been proposed to capture fine-grained sentiment features more accurately. For instance, the authors of[23] proposed an attention-based LSTM for sentence-level sentiment classification. The authors of[24,25] proposed hierarchical attention networks to simultaneously capture word-level and sentence-level content for sentiment analysis.

Most recently, large pre-trained Language models (e.g., Bert, Robert and Xlnet) have been leveraged to improve the performance of SLSA[4,5,26,27]. On the other hand, many researchers investigated how to integrate the traditional language features (e.g., part-of-speech, syntax dependency tree and knowledge-base) into DNN models for SLSA[26,28,29]. For instance, the authors of[26,29] proposed to fuse part-of-speech tag into raw Bert embeddings. Observing that the attention mechanism of Bert is vulnerable to noise while capturing semantic relations between words, the authors of[28] exploited a probability matrix of all the dependency arcs extracted by a dependency parser to compensate semantics derived from attention. What is more, there exist a lot of works on designing new networks for sentiment analysis based on the standard transformer structure[27,30].

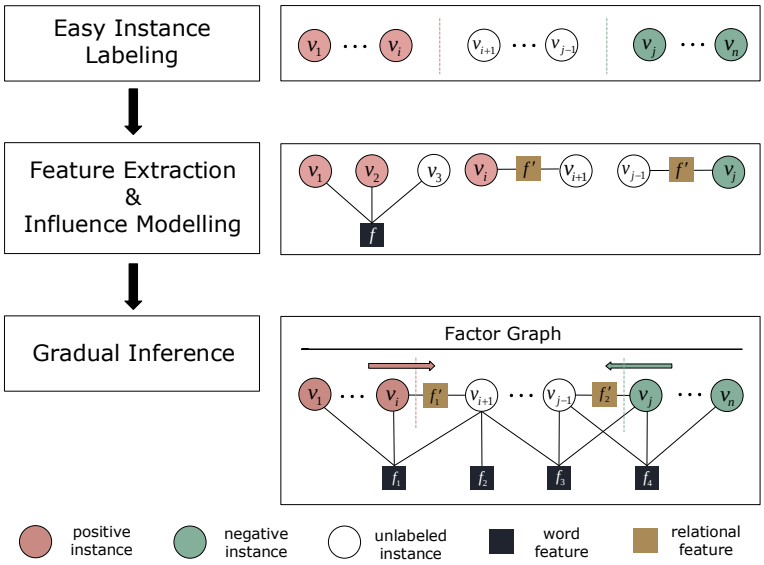


**Figure 2.** The general GML framework

Typically, they fed the outputs of the Bert model to a new network, reloading the parameters of the original pre-trained model to a new network. Several new pre-training proposals have also been presented to mitigate the mismatch between a new network structure and a pre-trained model. For instance, the work of[26] encoded sentiment score as part of input embedding and performed post-pretraining on the yelp datasets to get its own pre-trained model. The work of[27] modified the pre-training process to generate a new pre-trained model skep_ernie_2.0_large_en.

To better adapt a pre-trained model to downstream tasks, some researchers designed their own pre-training tasks[27,31]. For instance, the authors of[31] designed specific pre-training tasks to guide a model to predict phrase-level sentiment label. The authors of[27] reformulated multiple NLP tasks, which include sentence-level sentiment analysis, into a unified textual entailment task. So far, this approach achieved the state-of-the-art performance on sentence-level sentiment analysis. It is noteworthy that in this paper, we also integrate the external lexicon knowledge into DNN models to obtain sentiment-aware features for gradual learning.

It is noteworthy that all the above-mentioned deep learning solutions for SLSA were built upon the i.i.d learning paradigm. These efficacy depends on sufficiently large quantities of labeled training data. However, in real scenarios, there may not be sufficient labeled training data, and even if provided with sufficient training data, the distributions of training data and target data are almost certainly different to some extent. The gradual learning solution proposed in this paper is instead built upon the non-i.i.d learning paradigm. It can effectively alleviate the limitation resulting from distribution misalignment between training and target data.

## Preliminaries

In this section, we first define the SLSA task, and then provide a brief overview of the GML framework.

### Task Definition

As usual, this paper considers SLSA as a binary classification problem, in which a classifier needs to label each sentence as *positive* or *negative*. Formally, we define the task of SLSA as follows:

**Definition 1 (Sentence-Level Sentiment Analysis)** *Given a corpus of reviews* $\{r_0, r_1, r_2, \ldots, r_n\}$, *each review* $r_j$ *consisting of a sequence of sentences,* $\{s_{j1}, s_{j2}, \ldots, s_{jm}\}$, *the goal of SLSA is to predict the label of each sentence, where the label can be either positive* ($label = 1$) *or negative* ($label = 0$).

### The General GML Framework

As shown in Figure 2, the general GML framework consists of the following three essential steps:

#### *Easy Instance Labeling.*

Given a classification task, it is usually very challenging to accurately label all the instances in the task without good-coverage training examples. However, the work can become much easier if we only need to automatically label some easy instances in the task. In real scenarios, easy instance labeling can be performed based on the simple user-specified rules or the existing

**Algorithm 1** Scalable Gradual Inference Algorithm

**while** *there exists any unlabeled variable in* $G$ **do**
  $V' \leftarrow$ all the unlabeled variables in $G$;
  **for** $v \in V'$ **do**
    Measure the evidential support of $v$ in $G$;
  Select top-$m$ unlabeled variables with the most evidential support (denoted by $V_m$);
  **for** $v \in V_m$ **do**
    Approximately rank the entropy of $v$ in $V_m$;
  Select top-$k$ most promising variables in terms of entropy in $V_m$ (denoted by $V_k$);
  **for** $v \in V_k$ **do**
    Compute the probability of $v$ in $G$ by factor graph inference over a subgraph of $G$;
  Label the variable with the minimal entropy in $V_k$;

unsupervised learning techniques. For instance, in unsupervised clustering, an instance close to a cluster center can usually be considered as an easy instance, because it has only a remote chance to be misclassified. Gradual machine learning begins with the label observations of easy instances. Therefore, high accuracy of automatic easy instance labeling is critical for GML's ultimate performance.

For aspect-level sentiment analysis, as presented in our previous work[6], if a sentence contains some strong positive (res. negative) sentiment words, but no negation, contrast and hypothetical connectives, it can be reliably reasoned to be positive (res. negative). Please refer to[6] for algorithmic details. ***It is noteworthy that since this paper considers SLSA in the supervised setting, in which some labeled training data are supposed to be available, these training data with ground-truth labels can naturally serve as initial easy instances.***

#### *Feature Extraction and Influence Modeling.*

Feature serves as the medium for knowledge conveyance in the process of gradual learning. This step extracts the common features shared by labeled and unlabeled instances. To facilitate effective knowledge conveyance, it is desirable that a wide variety of features are extracted to capture diverse information. For each extracted feature, this step also needs to model its influence over the labels of its relevant instances. It is noteworthy that different applications usually require different features. In our previous work on unsupervised GML for aspect-level sentiment analysis[6], we extracted sentimental words and explicit polarity relations indicated by discourse structures to enable knowledge conveyance. In this paper, provided with labeled training data, we instead leverage DNNs to extract implicit sentiment features for SLSA.

#### *Gradual Inference.*

This step gradually labels the instances with increasing hardness in a task. Since the scenario of gradual learning does not satisfy the i.i.d assumption, gradual learning is fulfilled from the perspective of evidential certainty. Specifically, gradual learning is conducted over a factor graph, which consists of the labeled and unlabeled instances and their common features, by iterative factor inference. At each iteration, it chooses to label the unlabeled instance with the highest degree of evidential certainty.

In practice, GML is usually implemented by scalable gradual inference[8], which is sketched in Algorithm 1. It consists of three steps: 1) measurement of evidential support; 2) approximate ranking of entropy; 3) factor subgraph inference. Given a factor graph $G$, it first selects the top-$m$ unlabeled variables with the most evidential support in $G$ as the candidates for probability inference. To reduce the invocation frequency of factor graph inference, it then approximates entropy estimation by an efficient algorithm on the $m$ candidates and selects only the top-$k$ most promising variables among them for factor graph inference. Finally, it infers the probabilities of the chosen $k$ variables by factor subgraph inference.

In each iteration, GML generally chooses to label the inference variable with the highest degree of evidential certainty. Suppose that the total number of class labels, denoted by $\{L_1, L_2, \ldots, L_t\}$, is $t$. Given an inference variable $v$, GML measures its evidential certainty by the inverse of entropy as follows

$$E(v) = \frac{1}{H(v)} = \frac{1}{-\sum\limits_{1 \le i \le t} P_i(v) \cdot \log_2 P_i(v)}, \tag{1}$$

in which $E(v)$ and $H(v)$ denote the evidential certainty and entropy of $v$ respectively, and $P_i(v)$ denotes the inferred probability of $v$ having the label of $L_i$. The iteration is repeatedly invoked until all the instances in a task are labeled. It is noteworthy that in the process of gradual inference, a newly labeled instance at the current iteration would serve as an evidence observation in the following iterations.

| Notation | Description |
|---|---|
| $v$ | affective vector |
| $h$ | hidden states vector |
| $h_a$ | affective-aware hidden states vector |
| $w$ | affective attention weight vector |
| $w^i$ | affective attention weight of a word i |
| $w_d$ | the $d$-th dimensional affective weight(e.g., the weight of Evaluation) |
| $v_d^i$ | the $d$-th dimension affective value of a word i(e.g., the sentiment score of word i in Evaluation) |

**Table 1.** Frequently Used Notations.

## Supervised GML Solution

The supervised solution directly uses labeled examples in training data as easy instances. As mentioned in the introduction, it leverages two types of DNN models for knowledge conveyance: a polarity classifier, which is supposed to extract polarity-sensitive vector representations for detecting polarity similarity between close neighbors in a deep embedding space, and a semantic deep network, which can detect both similar and opposite polarity relations between two arbitrary sentences. We

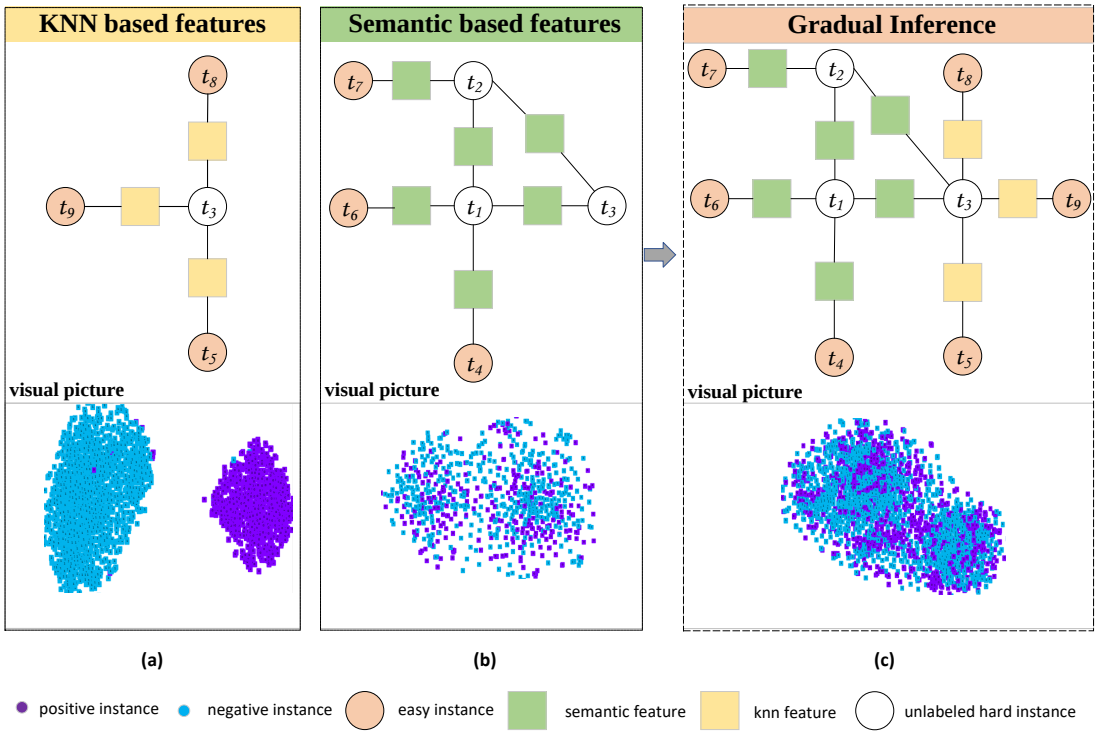


**Figure 3.** Supervised GML Solution for SLSA.

illustrate the proposed solution by the example as shown in Figure3. In the example, there are three unlabeled instances: $t_1$, $t_2$ and $t_3$. The subfigures (a) and (b) show their neighborhood-based similarity features and semantic relation features respectively, and the subfigure (c) shows the constructed factor graph. In the rest of this section, we first present the DNN models to extract polarity relation features, and then describe how to model them as factors in a factor graph to facilitate gradual learning. For presentation simplicity, we have summarized the frequently used notations in Table 1.

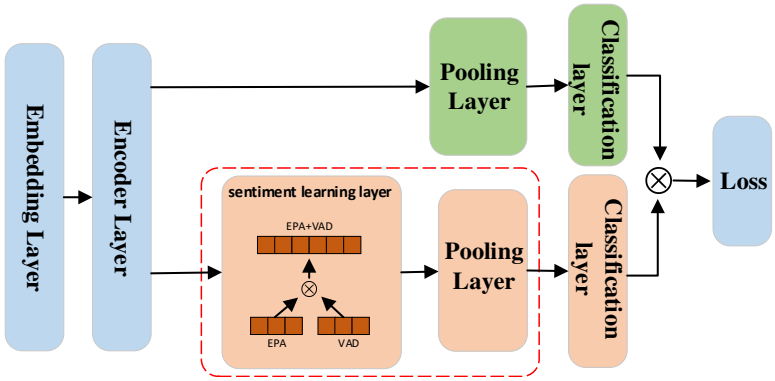


**Figure 4.** The EFL-based polarity classification model

## Feature Extraction: Polarity Relations

### *Similarity Relations By Polarity Classifier*

Motivated by the work presented in[32], we enhance the training of polarity classifier for SLSA by encoding affective knowledge handcrafted in sentiment lexicons. As shown in Figure 4, our solution adds a new branch of sentiment attention upon the EFL-based DNN model[27].

The new branch of sentiment attention, which consists of a sentiment learning layer and a pooling layer, serves to reflect the explicit sentiment polarities of each word as indicated by sentiment lexicons. In the sentiment learning layer, we build affective vectors by using two sentiment lexicons, EPA(Evaluation, Potency, and Activity) and VAD (Valence, Arousal, and Dominance), both of which measure sentiment orientation by three separate dimensions of continuous numerical values. Specifically, we concatenate the two word-level vectors of each word into a 6-dimensional affective vector, $v$, where the six dimensions correspond to Evaluation, Potency, Activity, Valence, Arousal, and Dominance respectively. In the model, the output of the encoder layer is a vector of hidden states denoted by $h \in \mathscr{R}^{b\times m\times e}$, which is then fed to the pooling layer, in which $b$ denotes the batch size, $m$ denotes the max length of sequence, $e$ denotes the embedding dimension. Before conveying $h$ to the pooling layer, we transform $h$ into a new vector of affective-aware hidden states, which is denoted by $h_a$. Specifically, we measure the attention weight of each sentiment word by the weighted sum of its sentiment dimension values as follows:

$$w^i = 1 + \sum_{d=0}^{5} (w_d \times v_d^i), \tag{2}$$

in which $w^i$ denotes the attention weight of a sentiment word, $w_d$ denotes the $d$-th affective dimension weight, and $v_d^i$ denotes the word's sentiment value in the $d$-th affective dimension. Note that the values of $w_d$ represents the weights of six dimensions (namely Evaluation, Potency, Activity, Valence, Arousal and Dominance); in our implementation, we set their values at [0.2,0.2,0.3,0.3,0.2,0.2] as suggested by[32]. The value of $v_d^i$ denotes a word's sentiment value in the $d$-th affective dimension, which can be directly extracted from the EPA and VAD lexicons. It is also noteworthy that the dimension value domain of the EPA lexicon is [0,1], while the dimension value domain of the VAD is [-5,5]. Therefore, we use a mapping function to unify the domains of EPA and VAD at [0,1]. Based on the setting of $w_d$ and $v_d^i$, the value domain of $w^i$ is between 1 and 2.4. If a word is absent in the sentiment lexicons, we set its attention weight as 1, or $w^i = 1$, effectively ignoring the lexicon influence.

Next, we concatenate the attention weights of all the words in a sentence to obtain its affective attention weight vector $w$ as follows:

$$w = \left[w^1, w^2, ..., w^m\right] \tag{3}$$

where $w \in \mathscr{R}^{m\times 1}$. Then, we compute the new affective-aware hidden states vector $h_a$ by weighting the original hidden states vector $h$ with $w$ as follows:

$$h_a = w \times h. \tag{4}$$

In the final classification layer, we fuse the features from both branches as follows:

$$l^{'} = l_\eta + \theta \times l_\xi, \tag{5}$$

in which $l_\eta$ and $l_\xi$ denote the losses generated by the representation learning branch and the sentiment attention branch respectively, and $\theta$ denotes the penalty weight parameter to balance the contributions from two branches. Since the sentiment

attention branch is supposed to complement the main representation learning branch, we suggest to set the value of $\theta$ at less than 0.5. In our implementation, we set $\theta = 0.3$.

Specifically, we have

$$l_\eta = y_\eta \log \hat{y} + (1 - y_\eta) \log(1 - \hat{y}), \tag{6}$$

and

$$l_\xi = y_\xi \log \hat{y} + (1 - y_\xi) \log(1 - \hat{y}). \tag{7}$$

We fine-tune the polarity classification model as shown in Figure 4 using labeled training data, and then exploit the resulting vector representations (the last-layer embeddings) for polarity similarity detection. In the implementation, we have constructed the DNN of polarity classification based on the state-of-the-art EFL model[27]. For each unlabeled sentence in a target workload, we extract its $k$-nearest neighbors from both the labeled and unlabeled instances. We use cosine distance to measure similarity. The value of $k$ is usually set to a small number to ensure the accuracy of extracted relations. Furthermore, we use a threshold (e.g., 0.001 in our experiments) to filter out the nearest neighbors not close enough in the embedding space. Our experiments have demonstrated that the performance of supervised GML is robust w.r.t the value of $k$ provided that it is set within a reasonable range (between 1 and 9).

***Similarity/Opposite Relations By Semantic Deep Network***

Built upon the Transformer architecture, the semantic deep network aims to detect the polarity relation between two arbitrary sentences. The backbone of a transformer is an encoder consisting of multiple multi-head self-attention layers. Each layer

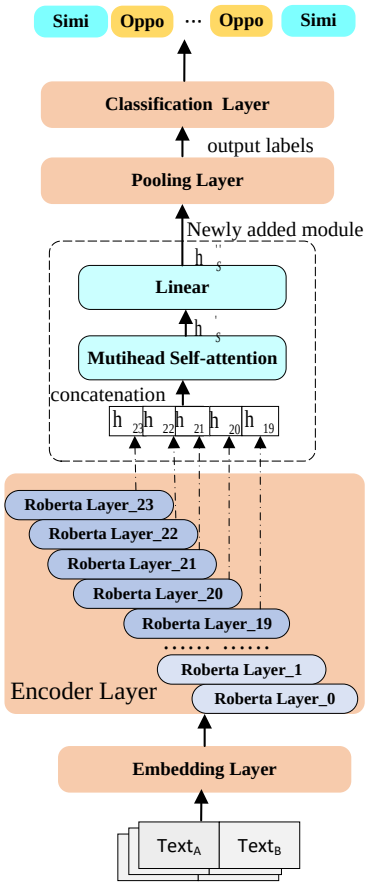


**Figure 5.** The Semantic Deep Network.

has the same network structure but different parameter weights. It has been well recognized that in a transformer, besides the last hidden layer, other layers also contain sentimental information. Therefore, we add a self-attention layer to aggregate the information present in the last five layers of transformer, and use a super feature vector to comprehensively capture sentimental features.

Specifically, as shown in Figure 5, the structure of the semantic deep network can be represented by the following three equations:

$$h_s = h_{19} \oplus h_{20} \oplus h_{21} \oplus h_{22} \oplus h_{23}, \tag{8}$$

$$h_s^{'} = f_m(h_s), \tag{9}$$

$$h_s^{''} = w_s \times h_s^{'} + b_s. \tag{10}$$

In Eq. 8, $\oplus$ denotes the concatenation operation, $h_i$ represents the vector output of the previous ith hidden layer, $i \in \{19,20,21,22,23\}$ and $h_i \in \mathscr{R}^{b \times m \times e}$, in which $b$ denotes the batch size, $m$ denotes the max length of sequence, $e$ denotes the embedding dimension, $h_s$ denotes the super vector after concatenation, $h_s \in \mathscr{R}^{b \times m \times 5e}$. Eq. 9 exploits the muti-head self attention function $f_m$ on the super vector to learn more different features. Finally, Eq. 10 employs a linear function to map dimension size of $b \times m \times 5e$ to the original dimension size of $b \times m \times e$ for subsequent layers, where $w_s \in R^{5e \times e}$ denotes a weight matrix and $b_s$ denotes the bias, both of which are supposed to be learned by training .

Provided with the super vector of $h_s$, the first step of the multi-head self-attention function is to map the raw input vector to a query vector, a key vector and a value vector by linear transformation as follows:

$$q_s = w_q \times h_s + b_q, \tag{11}$$

$$k_s = w_k \times h_s + b_k, \tag{12}$$

$$v_s = w_v \times h_s + b_v, \tag{13}$$

in which $q_s$, $k_s$ and $v_s$ denote query vector, key vector and value vector respectively, $w_q$, $w_k$ and $w_v$ denote weight matrix with the size of $R^{b \times m \times 5e}$, and $b_q$, $b_k$, $b_v$ denote bias.

Then, it exploits a softmax function to convert the query vector and key vector into an attention probability as follows:

$$a_p = Softmax\left(\frac{q_s \times {k_s}^T}{\sqrt{\gamma}}\right) \tag{14}$$

in which $a_p \in \mathscr{R}^{b \times m \times m}$. Finally, it multiplies $a_p$ with $v_s$ to get a combined context-informed word feature $h_s^{'}$ as follows:

$$h_s^{'} = a_p \times v_s \tag{15}$$

Then, it transforms $h_s^{'}$ to $h_s^{''}$ according to Eq. 10. It subsequent processing is similar to the traditional transformer architecture.

For SLSA, we construct polarity relations between labeled and unlabeled sentences based on a trained semantic deep network. In the training phase, we randomly extract $r$ labeled sentences from training data for each labeled sentence to fine-tune the semantic network. Then, in the feature extraction phase, we randomly extract $r$ sentences from labeled training data for each unlabeled sentence in the target workload, and construct its relations w.r.t them based on the semantic network. Our experiments has demonstrated that the performance of supervised GML is very robust w.r.t the value of $r$ provided that it is set within a reasonable range ($3 \leq r \leq 8$).

## Factor Modeling of Binary Relations

A factor graph for gradual machine learning consists of evidential variables, inference variables and factors. In the case of SLSA, a variable corresponds to a sentence and a factor defines a binary relation between two variables. In the process of GML, the labels of inference variables need to be gradually inferred. The label of a variable once inferred remains unchanged. The factor graph of the illustrative example has been shown in Figure 3.

In the supervised setting, all the labeled training data serve as the easy instances. The labels of the sentences in a target workload need to be gradually inferred by knowledge conveyance through the extracted binary relations. Specifically, we define the binary factor of a KNN-based similarity relation, $f_k$, as

$$\varphi_{f_k}(v_i, v_j) = \begin{cases} e^{w_{f_k}} & if\ v_i = v_j; \\ 1 & otherwise; \end{cases} \tag{16}$$

where $v_i$ and $v_j$ denote the two variables sharing the KNN-based similarity relational feature $f_k$, and $w_{f_k}$ denotes the weight of $f_k$. Similarly, we define the binary factor of a semantic relation between two variables, $f_s$, as

$$\varphi_{f_s}(v_i, v_j) = \begin{cases} e^{w_{f_s}} & if\ v_i = v_j; \\ 1 & otherwise; \end{cases} \tag{17}$$

| Dataset | Train | Validation | Test |
|---|---|---|---|
| MR | 8534 | 1066 | 1050 |
| CR | 2262 | 754 | 754 |
| twitter2013 | 5098 | 915 | 2034 |
| SST | 6920 | 872 | 1821 |

**Table 2.** The statistics of the test datasets.

where $v_i$ and $v_j$ denote the two variables sharing the semantic relational feature $f_s$, and $w_{f_s}$ denotes the weight of $f_s$.

In our implementation, the same type of factors are supposed to have the same weight. Initially, the weights of the similarity factors (whether KNN-based or semantic factors) are set to be positive (e.g., 1 in our experiments) while the weights of the opposite semantic factors are set to be negative (e.g., -1 in our experiments). It is noteworthy that the weights of three parameters would be continuously learned based on evidential observations in the inference process.

## Experiments

In this section, we empirically evaluate the performance of the proposed solution by a comparative study. The subsection  describes the experimental setup. The subsection  presents the comparative evaluation results. The subsection  evaluates performance sensitivity of the proposed solution w.r.t algorithmic parameters.

### Experimental Setup

For comparative evaluation, we use the benchmark datasets of MR, CR, twitter2013 and SST. Both MR and SST are movie review collections, CR contains the customer reviews of electronic products, while Twitter2013 contains microblog comments, which are usually shorter than movie and product reviews. The detailed statistics of the test datasets are presented in Table 2.

| Model | CR | | MR | | SST | | twitter2013 | |
|---|---|---|---|---|---|---|---|---|
| | Acc | Macro-F1 | Acc | Macro-F1 | Acc | Macro-F1 | Acc | Macro-F1 |
| EFL | 93.94%±0.04 | 95.36%±0.12 | 92.27%±0.32 | 92.23%±0.22 | 94.51% ±0.02 | 94.60%±0.40 | 93.36%±0.55 | 95.38%±0.45 |
| SentiLare | 92.41%±0.14 | 94.03% ±0.11 | 91.52%±0.10 | 91.42%±0.15 | 94.56%±0.80 | 94.69% ±0.23 | 92.90% ±0.26 | 95.02% ±0.24 |
| Roberta-large | 93.73%±0.43 | 95.06%±0.27 | 92.13%±0.25 | 92.12%±0.28 | 95.66%±0.91 | 95.75%±0.64 | 94.36%±0.44 | 96.10% ±0.30 |
| Xlnet-large | 93.44%±0.07 | 94.83%±0.04 | 91.31% ±0.26 | 91.33% ±0.25 | 95.30%±0.19 | 95.40%±0.73 | 93.80%±0.55 | 95.72% ±0.50 |
| Robert-base | 93.04%±0.46 | 94.50%±0.31 | 90.23%±0.23 | 90.27%±0.21 | 94.82%±0.19 | 95.00%±0.13 | 93.51%±0.37 | 95.50% ±0.31 |
| Xlnet-base | 92.84%±0.28 | 94.40%±0.15 | 90.09%±0.29 | 90.16%±0.73 | 93.00%±0.38 | 93.17%±0.38 | 92.51% ±0.09 | 94.84%±0.09 |
| SBERT | 92.93 %±0.10 | 93.55 %±0.32 | 92.38%±0.45 | 92.58%±0.40 | 95.09% ±0.80 | 95.22%±0.21 | 93.25%±0.11 | 94.31%±0.32 |
| AGN | 91.89%±0.13 | 91.24%±0.20 | 87.60%±0.32 | 87.57%±0.13 | 92.72%±0.20 | 90.94%±0.29 | 91.26%±0.30 | 91.26%±0.45 |
| DualCL | 92.19%±0.28 | 92.68%±0.85 | 89.41%±0.54 | 89.06%±0.13 | 93.41%±0.10 | 93.58%±0.77 | 88.94%±0.24 | 89.05%±0.22 |
| SLSA-GML | **95.62%**±0.21 | **96.54%**±0.29 | **93.16%**±0.30 | **93.04%**±0.32 | **96.27%**±0.12 | **96.30%**±0.15 | **94.94%**±0.20 | **96.49%**±0.20 |

**Table 3.** Comparative evaluation results: the best accuracy on each dataset is highlighted in bold.

Because the recent proposed DNN solutions based on pre-trained language models have been empirically shown to outperform earlier proposals, we compare the proposed solution, denoted by GML, with the following state-of-the-art models:

- SentiLare[26]. As a language representation model, it introduces word-level linguistic knowledge, which include part-of-speech tag and sentiment polarity, into pre-trained models and uses a label-aware masked language model to construct knowledge-aware language representation.
- Roberta-large[4]. It purposedly removes the next sentence prediction objective and dynamically changes masking pattern to improve the performance on downstream tasks.
- Roberta-base[4]. It is a simpler version of Roberta-large with only 12 hidden layers.

| Model | CR | | MR | | SST | | twitter2013 | |
|---|---|---|---|---|---|---|---|---|
| | Acc | Macro-F1 | Acc | Macro-F1 | Acc | Macro-F1 | Acc | Macro-F1 |
| SLSA-GML(w/o knn) | 95.36%±0.20 | 96.32%±0.35 | 93.07% ±0.17 | 92.97% ±0.72 | 96.16%±0.71 | 96.20% ±0.45 | 94.74% ±0.23 | 96.34%±0.85 |
| SLSA-GML(w/o semantic) | 94.56% ±0.75 | 95.69% ±0.25 | 93.06% ±0.89 | 92.94% ±0.65 | 95.61%±0.27 | 95.66% ±0.55 | 93.90%±0.86 | 95.78%±0.95 |
| SLSA-GML | **95.62%**±0.74 | **96.54%**±0.06 | **93.16%**±0.28 | **93.04%**±0.90 | **96.27%**±0.39 | **96.30%**±0.43 | **94.94%**±0.41 | **96.49%**±0.11 |

**Table 4.** The evaluation results of ablation study.

| $k_n$ | $k_s$ | CR | | MR | | SST | | twitter2013 | |
|---|---|---|---|---|---|---|---|---|---|
| | | Acc | Macro-F1 | Acc | Macro-F1 | Acc | Macro-F1 | Acc | Macro-F1 |
| 3 | 3 | 95.09%±0.13 | 96.13%±0.21 | 92.88%±0.20 | 92.76%±0.11 | 96.21%±0.21 | 96.24%±0.19 | 95.03%±0.47 | 96.56%±0.29 |
| 3 | 5 | 95.76%±0.23 | 96.64%±0.27 | 93.35%±0.17 | 93.22% ±0.17 | 96.32%±0.20 | 96.36%±0.22 | 94.54%±0.43 | 96.21%±0.27 |
| 3 | 7 | 95.62%±0.10 | 96.54% ±0.42 | 93.34%±0.35 | 93.22% ±0.44 | 96.27%±0.25 | 96.30%±0.17 | 94.74%±0.35 | 96.35%±0.29 |
| 3 | 8 | 95.49%±0.51 | 96.43%±0.65 | 93.35%±0.48 | 93.22%±0.52 | 96.05%±0.32 | 96.09%±0.61 | 94.44%±0.33 | 96.14%±0.66 |
| 5 | 3 | 95.23% ±0.34 | 96.25%±0.50 | 93.53%±0.32 | 93.46%±0.19 | 96.43%±0.81 | 96.45%±0.21 | 94.99%±0.34 | 96.53%±0.65 |
| 5 | 5 | 95.36%±0.78 | 96.32%±0.34 | 93.16%±0.56 | 93.03%±0.41 | 96.00% ±0.71 | 96.04%±0.80 | 94.59%±0.91 | 96.23%±0.87 |
| 5 | 7 | 95.62%±0.40 | 96.53%±0.61 | 93.25%±0.21 | 93.14%±0.53 | 96.21%±0.10 | 96.25%±0.23 | 94.94%±0.31 | 96.48%±0.40 |
| 7 | 3 | 95.36%±0.81 | 96.36% ±0.21 | 93.63%±0.30 | 93.56%±0.22 | 96.21%±0.57 | 96.24%±0.45 | 94.89% ±0.18 | 96.46%±0.15 |
| 7 | 5 | 95.62% ±0.20 | 96.54%±0.30 | 93.44%±0.82 | 93.36%±0.44 | 95.94%±0.20 | 95.96%±0.19 | 94.49%±0.21 | 96.17%±0.41 |
| 7 | 7 | 95.76%±0.29 | 96.64% ±0.30 | 93.44%±0.29 | 95.94%±0.91 | 95.97%±0.30 | 96.00%±0.10 | 94.59%±0.90 | 96.24%±0.12 |
| 7 | 8 | 95.49% ±0.12 | 96.42%±0.20 | 93.16%±0.51 | 93.03%±0.11 | 95.99% ±0.29 | 96.04%±0.21 | 94.44%±0.34 | 96.14%±0.37 |

**Table 5.** Sensitivity evaluation.

- Xlnet-large[5]. It is based on a generalized autoregressive pre-trained model that can learn bidirectional contexts by maximizing the expected likelihood over all the permutations of factorization order.
- Xlnet-base[5]. It is a simpler version of Xlnet-large with only 12 hidden layers.
- SBERT[31]. It use the siamese and triplet network structures to derive semantically meaningful sentence embeddings for sentimental polarity detection.
- EFL[27]. Converting class labels into auxiliary sentences, it is a unified model that can model multiple NLP tasks as a textual entailment task.
- AGN[33]. It integrates statistical information with semantic representation to train a robust classifier for sentiment analysis.
- DualCL[34]. It is a recently proposed framework for sentiment analysis that can simultaneously learn the features of input samples and the parameters of classifiers in the same embedding space.

Our implementation uses the affective-aware EFL model[27] as the baseline polarity classifier and leveraged external affective knowledge to extract the KNN-based similarity relations. It uses the improved Roberta-large model[4] to extract similar and opposite semantic relations between two arbitrary sentences. Specifically, the Roberta-large model consists of 16 heads and 24 layers with the hidden layer size of 1024. Our implementation keeps the dropout probability at 0.1 and sets the number of epochs to 3. It set the initial learning rate at $2e^{-5}$ for all the layers and the batch size at 32. For the training of semantic deep network, we generate 6 semantic relations (3 with similar labels and 3 with opposite labels) for each labeled example. For GML factor graph construction, we randomly select 6 labeled examples in the training set for each unlabeled sentence, and use the trained binary semantic model to predict their polarity relations. As usual, we measure the performance of different solutions by the metrics of Accuracy and Macro-F1. All the reported results are averages over 5 runs. We report both the means and standard deviations (STD).

## Comparative Evaluation

The detailed evaluation results have been presented in Table 3. It can be observed that GML consistently achieves the state-of-the-art performance across all the test workloads in terms of Accuracy and Macro-F1. Specifically, in terms of accuracy, GML outperforms the existing best performer, EFL, by the margins of 1.68%, 0.89%, 1.76%, 1.58% on CR, MR, SST and Twitter2013 respectively. Similarly, in terms of macro-F1, GML outperforms EFL by the margins of 1.18%, 0.81%, 1.7%, 1.11% on them respectively. In terms of accuracy, GML outperforms the existing state-of-the-art by around 1.6% on CR, 0.7% on MR, 0.6% on SST, and 0.58% on twitter2013. In terms of Macro-F1, the improvement margins over the state-of-the-art on the four test workloads are 1.18%, 0.46%, 0.55%,0.40% respectively. It is noteworthy that the performance of two recent approaches, AGN and DualCL, is similar to other DNN models, but worse than GML. We have observed that the statistical features (e.g., word frequency), which are the focus of AGN, are not very helpful to sentiment analysis. As an augmentation approach, DualCL usually performs well in the circumstance where there are only a few labeled training data. In our scenarios of benchmark workloads, the efficacy of DualCL is however rather limited. By leveraging the state-of-the-art DNNs for feature extraction, non-i.i.d gradual learning has a clear advantage over i.i.d learning. It can also be observed that for both GML and deep models, the fluctuations of different runs remain low (STD values $< 0.5$ in most cases). Due to the well recognized challenge of SLSA, these observations clearly indicate the efficacy of the proposed approach.

| #Id | #Text | Ground_Truth_Label | SLSA_GML | DNN |
|---|---|---|---|---|
| $t_1$ | this camera is perfect for an enthusiastic amateur photographer . | Pos | Pos | Pos |
| $t_2$ | but i 've already emailed creative tech support about it , and gotten timely responses - they will fix it for me if necessary. | Pos | Pos | Neg |
| $t_3$ | this did manage to cut the number of pop ups during start-up from 4 down to 2. | Pos | Pos | Neg |
| $t_4$ | i imagine if i left my player untouched ( no backlight ) it could play for considerably more than 12 hours at a low volume level. | Pos | Pos | Neg |

**Table 6.** Example sentences.

**Ablation Study.** We have also conducted an ablation study on the proposed GML solution. The detailed evaluation results have been reported in Table 4. It can be observed that without either KNN-based relations or binary semantic relations, the performance of GML drops on all the test workloads. This observation clearly indicates that KNN-based relations and binary semantic relations are complementary to each other: their combined modeling in GML achieves better performance than either of them. However, it can also be observed that compared with knn relations, the performance of GML drops more considerably without binary semantic relations. The knn relations capture only similarity features, while the binary semantic relations can capture both similarity and opposite, or more diverse, relations. It is noteworthy that our experimental results are consistent with the expected characteristic of GML that more diverse features can usually facilitate knowledge conveyance more effectively.

**Illustrative Examples.** We illustrate the efficacy of GML by the examples from CR as shown in Table 6. On $t_1$, both GML and the deep model give the correct label; however, on all the other examples, GML gives the correct labels while the deep model mispredicts. The four subfigures show the constructed factor subgraphs of the examples respectively. It can be observed that $t_2$ has three relational factors, two of which are correctly predicted while the remaining one is mispredicted. However, GML still correctly predicts the label of $t_2$ because the majority of its relational counterparts indicate a positive polarity. The similar result has also been observed on $t_3$. It is noteworthy that GML labels these examples in the order of $t_1$, $t_2$, $t_3$ and $t_4$. Since the predicted labels of $t_2$ and $t_3$ provide $t_4$ labeling with correct polarity hints, $t_4$ is also correctly labeled as positive.

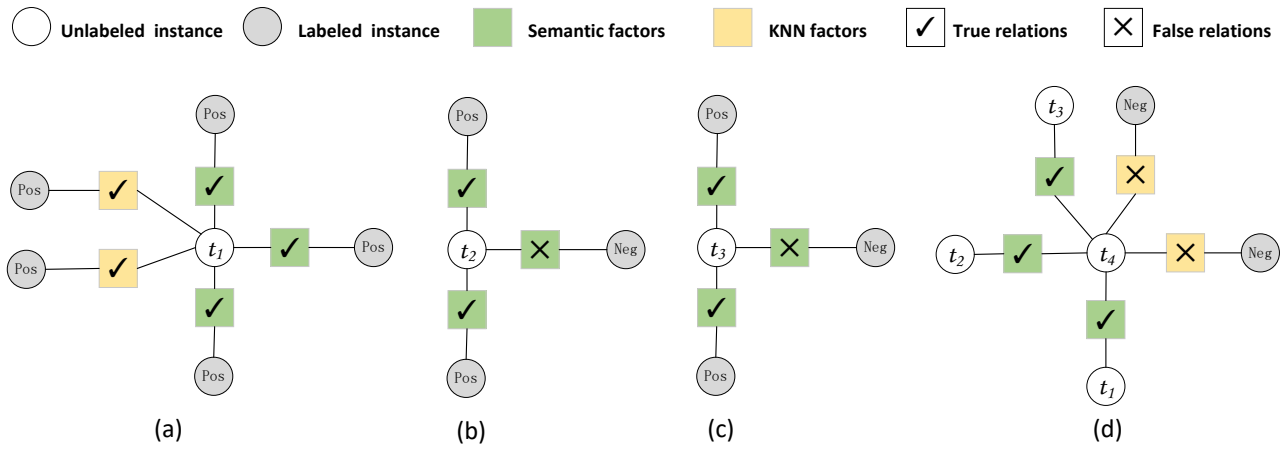


**Figure 6.** An illustrative example.

### Sensitivity Evaluation

We have also evaluated the performance sensitivity of GML w.r.t the number of extracted semantic relations and the number of extracted KNN relations respectively. Both parameters are set within the range between 3 and 8. The detailed evaluation results have been presented in Table 5. It can be observed that the performance of GML is very robust w.r.t both parameters. These experimental results bode well for its applicability of GML in real scenarios.

## Conclusion

In this paper, we have presented a novel approach based on GML for the task of sentence-level sentiment analysis. The proposed solution extensively leverages the existing DNN models to extract polarity-aware features, which are then used to enable effective gradual knowledge conveyance. Our extensive experiments on the benchmark datasets have shown that it achieves the state-of-the-art performance on all the test workloads. Our work clearly demonstrates that gradual machine learning, in

collaboration with DNN for feature extraction, can perform better than pure deep learning solutions on sentence-level sentiment analysis.

## Data availability

All the datasets used in this study are publicly available. MR dataset is available from https://www.cs.cornell.edu/people/pabo/movie-review-data/. CR dataset is available from https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#datasets. Twitter2013 dataset is available from https://www.dropbox.com/s/byzr8yoda6bua1b/2017_English_final.zip?file_subpath=%2F2017_English_final%2FGOLD%2F\Subtask_A. SST dataset is available from http://nlp.stanford.edu/sentiment. The sentiment lexicon EPA used in our paper is available from http://www.indiana.edu/~socpsy/public_files/EnglishWords_EPAs.xlsx, and another sentiment lexicon VAD is available from https://saifmohammad.com/WebPages/nrc-vad.html.

## References


1. Bongirwar, V. K. A survey on sentence level sentiment analysis. *Int. J. Comput. Sci. Trends Technol. (IJCST)* 110–113 (2015).
2. Pang, B. & Lee, L. Opinion mining and sentiment analysis. *Foundations Trends Inf. Retr.* 1—-135 (2008).
3. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert pre-training of deep bidirectional transformers for language understanding (2018).
4. Liu, Y. *et al.* Roberta a robustly optimized bert pretraining approach. *arXiv preprint arXiv1907.11692* (2019).
5. Yang, Z. *et al.* Xlnet generalized autoregressive pretraining for language understanding (2019).
6. Wang, Y. *et al.* Aspect-level sentiment analysis based on gradual machine learning. *Knowledge-Based Syst.* **212**, 106509 (2021).
7. Ahmed, M. *et al.* Dnn-driven gradual machine learning for aspect-term sentiment analysis (2021).
8. Hou, B., Chen, Q., Wang, Y., Nafa, Y. & Li, Z. Gradual machine learning for entity resolution. *IEEE Transactions on Knowl. Data Eng.* **34**, 1803–1814 (2022).
9. Ito, T., Tsubouchi, K., Sakaji, H., Yamashita, T. & Izumi, K. Contextual sentiment neural network for document sentiment analysis. *Data Sci. Eng.* **5** (2020).
10. Garg, S. & Ramakrishnan, G. Bae bert-based adversarial examples for text classification (2020).
11. Rhanoui, M., Mikram, M., Yousfi, S. & Barzali, S. A cnn-bilstm model for document-level sentiment analysis. *Mach. Learn. Knowl. Extr.* **1**, 832–847 (2019).
12. Rao, G., Huang, W., Feng, Z. & Cong, Q. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing* **308**, 49–57 (2018).
13. Zeng, B., Yang, H., Xu, R., Zhou, W. & Han, X. Lcf a local context focus mechanism for aspect-based sentiment classification. *Appl. Sci.* **9**, 3389 (2019).
14. Dai, J., Yan, H., Sun, T., Liu, P. & Qiu, X. Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta (2021).
15. Bai, X., Liu, P. & Zhang, Y. Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE Transactions on Audio, Speech, Lang. Process.* **29**, 503—-514 (2021).
16. Tripathy, A., Agrawal, A. & Rath, S. K. Classification of sentiment reviews using n-gram machine learning approach. *Expert. Syst. with Appl.* **57**, 117–126 (2016).
17. Fang, J. & Chen, B. Incorporating lexicon knowledge into svm learning to improve sentiment classification (2011).
18. Kumari.U, Sharma.A & Soni.D. Sentiment analysis of smart phone product review using svm classification technique (2017).
19. Chikersal, P., Poria, S. & Cambria, E. Sentu sentiment analysis of tweets by combining a rule-based classifier with supervised learning (2015).
20. Wang, J., Yu, L.-C., Lai, K. R. & Zhang, X. Tree-structured regional cnn-lstm model for dimensional sentiment analysis. *ACM Transactions on Audio, Speech, Lang. Process.* **28**, 581–591 (2019).

**21.** Li, W., Zhu, L., Shi, Y., Guo, K. & Cambria, E. User reviews sentiment analysis using lexicon integrated two-channel cnn-lstm family models. *Appl. Soft Comput.* **94**, 106435 (2020).

**22.** Minaee, S., Azimi, E. & Abdolrashidi, A. Deep-sentiment sentiment analysis using ensemble of cnn and bi-lstm models. *arXiv preprint arXiv 1904.04206* (2019).

**23.** Zhou, X., Wan, X. & Xiao, J. Attention-based lstm network for cross-lingual sentiment classification (2016).

**24.** Li, Z., Wei, Y., zhang, Y. & yang, Q. Hierarchical attention transfer network for cross-domain sentiment classification (2018).

**25.** Stappen, L. *et al.* Context modelling using hierarchical attention networks for sentiment and self-assessed emotion detection in spoken narratives (2019).

**26.** Ke, P., Ji, H., Liu, S., Zhu, X. & Huang, M. Sentilare sentiment-aware language representation learning with linguistic knowledge (2020).

**27.** Wang, S., Fang, H., Khabsa, M., Mao, H. & Ma, H. Entailment as few-shot learner. *arXiv preprint arXiv 2104.14690* (2021).

**28.** Zeng, J. *et al.* Improved review sentiment analysis with a syntax-aware encoder (2019).

**29.** Cheng, K., Yue, Y. & Song, Z. Sentiment classification based on part-of-speech and self-attention mechanism. *IEEE Access* **8**, 16387–16396 (2020).

**30.** Reimers, N. & Gurevych, I. Sentence-bert sentence embeddings using siamese bert-networks (2019).

**31.** Yin, D., Meng, T. & Chang, K.-W. Sentibert a transferable transformer-based architecture for compositional sentiment semantics (2020).

**32.** Xiang, R. *et al.* Affective awareness in neural sentiment analysis. *Knowledge-Based Syst.* **226**, 107137 (2021).

**33.** Li, X., Li, Z., Xie, H. & Li, Q. Merging statistical feature via adaptive gate for improved text classification (2021).

**34.** Chen, Q., Zhang, R., Zheng, Y. & Mao, Y. Dual contrastive learning text classification via label-aware data augmentation. *CoRR* (2022).


## Acknowledgements


The work is supported by the National Natural Science Foundation of China under Grant No. 62172335, No. 61972317, No.61732014, and No. 61672432, the Fundamental Research Funds for the Central Universities of China under Grant No.3102019DX1004.


## Author contributions

J.S. designed and implemented the algorithms, and prepared for the manuscript. Q.C. is responsible for devising the research plan, and revising the manuscript. Y.W. helped with algorithm implementation. Z.L. and W.P. helped with algorithm design and empirical evaluation. L.Z. edited the manuscript. All authors have reviewed the manuscript.

## Competing interests

The authors declare no competing interests.