

Implicit Sentiment Analysis with Event-Centered Text Representation

Deyu Zhou^{1*}, Jianan Wang¹, Linhai Zhang¹, Yulan He^{2 3}

¹School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China

²Department of Computer Science, University of Warwick, UK

³Alan Turing Institute, UK

{d.zhou, jn wang, lzhang472}@seu.edu.cn,
yulan.he@warwick.ac.uk

Abstract

Implicit sentiment analysis, aiming at detecting the sentiment of a sentence without sentiment words, has become an attractive research topic in recent years. In this paper, we focus on event-centric implicit sentiment analysis that utilizes the sentiment-aware event contained in a sentence to infer its sentiment polarity. Most existing methods in implicit sentiment analysis simply view noun phrases or entities in text as events or indirectly model events with sophisticated models. Since events often trigger sentiments in sentences, we argue that this task would benefit from explicit modeling of events and event representation learning. To this end, we represent an event as the combination of its event type and the event triplet <subject, predicate, object>. Based on such event representation, we further propose a novel model with hierarchical tensor-based composition mechanism to detect sentiment in text. In addition, we present a dataset¹ for event-centric implicit sentiment analysis where each sentence is labeled with the event representation described above. Experimental results on our constructed dataset and an existing benchmark dataset show the effectiveness of the proposed approach.

1 Introduction

Sentiment analysis aims at automatically detecting the sentiment of given text. Explicit sentiment analysis methods detect the sentiment mainly based on the occurrence of sentiment-related words and have been extensively explored (Agarwal et al., 2011; Tang et al., 2014, 2015a). However, sentiment could also be implicitly expressed. For example, the sentence ‘*I won the first place in the speech contest*’ does not contain any sentiment words, but the event of ‘*winning the first place*’ reflects the positive sentiment. Implicit sentiment analysis was

thus proposed to detect sentiment in the absence of sentiment words (Liu, 2012). Compared with explicit sentiment analysis, implicit sentiment analysis is more challenging as there is a lack of explicit cues for inferring the sentiment polarity.

Research on implicit sentiment analysis can be broadly classified into two categories: metaphor-based and event-centric. Metaphor/rhetoric-based implicit sentiment analysis methods typically detect sentiment based on a metaphoric sentiment dictionary and some manually designed rules (Zhang and Liu, 2011), while event-centric approaches assume sentiment is triggered by events described in sentences. Examples include an early approach which regarded noun phrases or entities in sentences as events (Greene and Resnik, 2009), and more recent approaches which indirectly model events by capturing contextual information using graph convolutional networks (Zuo et al., 2020) or attention mechanism (Wei et al., 2020). We argue that in the former, the event representation is oversimplified and consequently event-related knowledge would be largely lost; while in the latter, events are not directly modeled which makes it less effective in detecting sentiments triggered by events.

To overcome the limitations of existing event-centric approaches, we propose to construct a corpus in which event triplets in the form of <subject, predicate, object> and their corresponding types are annotated in sentences. In addition, each sentence is also assigned with a sentiment class label. An example sentence and its annotation are shown below:

‘*You abandon me for a week to go off on holiday with daddy, come back and barely 2 days later you go off out with him again.*’

Event triplet: <you, abandon, me>

Event type: abandonment

Sentiment: negative

The event triplet and the event type are combined as

*Corresponding author.

¹<https://github.com/FloatingIsland2/Implicit-Sentiment-Analysis>

the final representation of an event. Based on such event-centered text representation, we further propose a method for implicit sentiment analysis built on hierarchical tensor-based compositions, which effectively employs Tensor Composition (Socher et al., 2013; Weber et al., 2017) to encode the interaction between the subject, the predicate, the object and the sentence. Moreover, we adopt a multi-task learning framework to perform event type classification and sentiment classification simultaneously. Our experimental results show that event type classification benefits sentiment classification.

In summary, our contributions are as follows:

- We propose a novel model with hierarchical tensor-based compositions to detect sentiment based on event-centered text representations to explicitly model events and capture the interaction between the subject, the predicate, the object and the sentence.
- We further develop a multi-task learning framework to improve sentiment analysis with event type classification.
- We present a dataset, called EveSA, with annotated event triplets, event types, and sentence-level sentiment polarity labels for implicit sentiment analysis.
- Our model outperforms several competitive baselines on both EveSA and SemEval17-Task4 Subtask A (Rosenthal et al., 2017).

2 Related Work

Implicit Sentiment Analysis Liu (2012) first classifies sentiment analysis into explicit and implicit sentiment analysis. Generally speaking, implicit sentiment analysis can be further classified into metaphor-based and event-centric approaches. In metaphor-based approaches, sentences containing keywords found in a metaphor dictionary are considered implicitly expressing positive or negative emotional tendencies (Zhang and Liu, 2011). In event-centric approaches, events mentioned in text may imply positive or negative sentiments. Balahur et al. (2011) presented an approach for detecting event-triggered sentiment based on common-sense knowledge, EmotiNet, a knowledge base of concepts with associated affective value. Greene and Resnik (2009) used grammatical structures to mine language features related to implicit sentiments, and used similarity calculations to improve the performance of sentiment classification. Zuo

et al. (2020) proposed a context-specific heterogeneous graph convolutional network to address the problem of the absence of sentiment words.

Event-related Sentiment Analysis In recent years, researchers also pay attention to the importance of event information in sentiment analysis. Deng and Wiebe (2015) encoded a set of sentiment inference rules in a probabilistic soft logic framework for entity/event-level sentiment analysis tasks. Hofmann et al. (2020) encoded properties of events as latent variables following theories of cognitive appraisal of events to improve emotion classification performance. Gaonkar et al. (2020) tracked label-label correlations through label embeddings in sentiment classification tasks to maintain the consistency of emotion caused by the same type of event. Ding and Riloff (2016) first defined affective events as triples <subjective, verb, objective>, and created a dataset containing affective events with sentiment polarity labels. Ding and Riloff (2018b) expanded affective events as <subjective, predicate, objective, prepositional phrase>, and introduced a weakly supervised method for affective events classification. Ding and Riloff (2018a) categorized affective events into *physiological, health, leisure, social, financial, cognition, and freedom*, based upon human need related to people’s motivations and desires. Yang et al. (2019) regarded the topic distribution of a sentence as an implicit event, and proposed an event-driven attention with topics of sentences for emotion ranking. Zhuang et al. (2020) presented a discourse-enhanced self-training method that iteratively improved the classifier with unlabeled affective events.

3 Dataset: EveSA

Since the existing sentiment analysis datasets only contain text and sentiment polarity labels, but no event-related annotations, we construct an event-centered dataset for implicit sentiment analysis. We first identify event types from FrameNet (Baker et al., 1998) and then crawl tweets which contain the triggering words of the corresponding event types.

3.1 Construction of an Event Type Library

FrameNet (Baker et al., 1998) is an English vocabulary knowledge base that contains more than 1200 semantic frames (each frame can be regarded as an event type) and lexical units (each lexical units can be regarded as a predicate), and more than 200,000

labeled sentences under all frames, which can be used for learning models for NLP tasks such as information extraction and event detection.

With the FrameNet knowledge base, we build an event library consisting of event-related components: event types, argument roles (*subjects* and *objects*), and event triggers (*predicate*). Since not all frames contained in FrameNet are suitable as event types or have obvious sentiment inclinations, we define the filtering rules as follows:

- Filter frames that do not contain the *subject* and *object* argument roles;
- Remove frames that contain more than two arguments to control the complexity of the events;
- Artificially remove some frames that do not have sentiments to ensure the selected event types are balanced in the three sentiment categories (*positive*, *negative* and *neutral*).

According to these rules, 18 frames are selected out of a total of 1200 frames in FrameNet as our identified event types, listed in Table 1. Finally, we build a dictionary of trigger words for each event type by only keeping the verb and other lexical units with no ambiguous word senses as the predicates of each event type.

Event Type	Total	Positive	Neutral	Negative
Abandonment	135	37	24	74
Affirm_or_deny	151	29	29	93
Certainty	163	51	21	91
Cause_harm	451	12	61	378
Endangering	100	1	22	77
Expectation	234	99	73	62
Experiencer_focused_emotion	100	63	9	28
Judgement	354	132	49	173
Stimulate_emotion	865	270	60	535
Work	118	80	25	13
Assistance	215	152	31	32
Attention(ignore)	104	8	20	76
Detaching	107	23	39	45
Killing	279	18	16	245
Leadership	160	56	56	48
Offering	112	80	18	14
Supply	158	82	33	43
Theft	175	12	38	125
Total	3981	1205	624	2152

Table 1: Annotated event types and their sentiment class distributions in our constructed dataset – EveSA.

3.2 Data Collection and Cleaning

We employ `tweet_scrapper`² to crawl tweets containing at least one of the trigger words of the event types identified in Section 3.1. To clean the data, first, the Python package `preprocessor`³ is used to remove the URLs, HashTags, and various emojis in text. Next, HTTP tokens, other special tokens, non-English characters, and consecutively repeated characters are removed from text. Afterwards, each tweet is processed using `CoreNLP`⁴ to obtain the word segmentation results. Finally, tweets with more than 40 words are filtered.

3.3 Annotation of Event Triplets and Sentiment Polarity

Given the word segmentation result of each text $sen = \{w_1, \dots, w_i, \dots, w_n\}$, in which n is the sentence length. Two annotators first verify the word segmentation results. Next, the annotators assign the event type $etype$, and mark the position of the predicate or the event trigger words in the sentence. Afterwards, the arguments are annotated for each event. The j th argument role under the event type $etype$ has the following annotations: $argrole_j = \{w_u, w_{u+1}, \dots, w_v\}$, $0 < j \leq m$, where m represents the number of argument roles under this $etype$. Each sentence is also annotated with event triplets, $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ and one of the sentiment categories, *positive*, *neutral* or *negative*. The independently annotated results are compared and any inconsistent annotations are resolved through discussions. The final dataset contains 18 event types and 3,981 sentences, each with an annotated event triplet. Detailed data set statistics are shown in Table 1.

4 Model Description

4.1 Problem Setting

Given (1) a sentence $sen = \{w_1, w_2, \dots, w_n\}$ associated with a sentiment category; (2) its event triplet $\langle \text{subject: } subj = \{w_1, w_2, \dots, w_{n_{subj}}\}, \text{predicate } pred = \{w_1, w_2, \dots, w_{n_{pred}}\}, \text{object } obj = \{w_1, w_2, \dots, w_{n_{obj}}\} \rangle$; and (3) its corresponding event type $etype$, the task of fact-based implicit sentiment analysis model is to predict the sentiment distribution over three sentiment polari-

²https://github.com/5h1rish/tweet_scrapper

³<https://github.com/s/preprocessor>

⁴<https://stanfordnlp.github.io/CoreNLP/>

ties. We propose an event-centered text representation model to solve this problem.

4.2 Overview

Traditional sentence-level sentiment analysis infers the sentiment category of a sentence from text directly. We instead propose to detect the sentiment category of a sentence based on its associated event, the triplet $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ and the event-related information apart from the textual information of the sentence, as depicted in Figure 1.

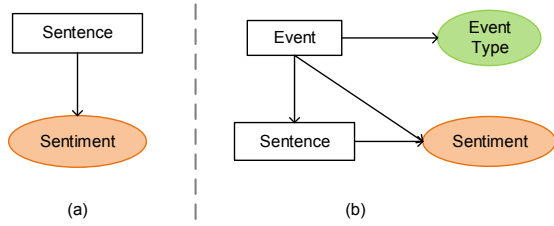


Figure 1: (a) Traditional sentence-level sentiment analysis where sentiment is inferred from text directly. (b) our event-centered approach in which sentiment is detected based on the event triplet, the event type in addition to the textual information of the sentence. Z is the noise introduced to increase sentence diversity.

The overall architecture of our proposed method is presented in Figure 2. It consists of three parts. (1) To integrate the event triplet $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ with the sentence, hierarchical tensor-based compositions are employed. Both event triplet and the sentence are first fed into a BERT encoder (Devlin et al., 2018). Then the bottom-level tensor composition (Weber et al., 2017) is used to obtain the event representation e , which is further combined with the BERT-encoded sentence representation by the top-level tensor composition to generate the output r_{final} . (2) To model the event type information, a multi-task learning framework is used to perform event type classification and sentiment classification simultaneously, since in our preliminary experiments, accurate classification of event types can benefit the sentiment classification task. (3) Since the same event could be described in sentences with diverse surface forms, we assume the BERT-encoded sentence representation h_{sen} follows a Gaussian distribution with its mean determined by the event representation, e , i.e., $h_{sen} \sim \mathcal{N}(e, \sigma^2)$. Such a constraint is added to the loss function to make the sentence representation in the embedding space to be close to the learned event representation.

4.3 Hierarchical Tensor-based Compositions

The event triplet input to the BERT encoder is: [CLS] subject [SEP] predicate [SEP] object [SEP]. The hidden states of these three parts are $\mathbf{H}_{subj} \in \mathbb{R}^{L_{subj} \times d}$, $\mathbf{H}_{pred} \in \mathbb{R}^{L_{pred} \times d}$ and $\mathbf{H}_{obj} \in \mathbb{R}^{L_{obj} \times d}$, where L_{subj} , L_{pred} and L_{obj} are the maximum sequence length of the subject, predicate and the object, respectively, and d is the dimension of the BERT hidden states. After averaging the constituent token embeddings of each event element separately, the hidden states of the three parts are $h_{subj} \in \mathbb{R}^d$, $h_{pred} \in \mathbb{R}^d$ and $h_{obj} \in \mathbb{R}^d$, which are then fed into Tensor-based Compositions to model the relationship among the subject, the predicate, and the object.

First, the interaction between the subject and the predicate, $r_{s-p} \in \mathbb{R}^d$, is computed as follows:

$$r_{s-p} = f \left(h_{subj}^T T_1^{[1:k]} h_{pred} + W \begin{bmatrix} h_{subj} \\ h_{pred} \end{bmatrix} + b \right) \quad (1)$$

where $T_1^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ is a tensor, which consists of a set of k matrices, each with $d \times d$ dimensions. The tensor product $h_{subj}^T T_1^{[1:k]} h_{pred}$ is a vector $r \in \mathbb{R}^k$, and each entry is computed by one slice of tensor T_1 : $r_i = h_{subj}^T T_1^{[1:k]} h_{pred}$, ($i = 1, \dots, k$). Other parameters are trainable in a standard feed-forward neural network, where $W \in \mathbb{R}^{k \times 2d}$ is the weight matrix, $b \in \mathbb{R}^{k \times k}$ is the bias vector. The calculation diagram of Tensor Composition is shown in Figure 3:

Similarly, to compute the interaction between the predicate and the object, $r_{p-o} \in \mathbb{R}^d$ is computed by:

$$r_{p-o} = f \left(h_{pred}^T T_2^{[1:k]} h_{obj} + W \begin{bmatrix} h_{pred} \\ h_{obj} \end{bmatrix} + b \right) \quad (2)$$

Finally, the interaction between the subject and the object is calculated by r_{s-p} and r_{p-o} of the second layer in Tensor-based Compositions, with its output regarded as the event representation e :

$$e = f \left(r_{s-p}^T T_3^{[1:k]} r_{p-o} + W \begin{bmatrix} r_{s-p} \\ r_{p-o} \end{bmatrix} + b \right) \quad (3)$$

The sentence $h_{sen} \in \mathbb{R}^d$ is encoded with BERT. To better fuse the event representation e with the contextual information in the sentence, we model their interaction in the same way, using Tensor-based Composition, to derive the final representation r_{final} :

$$r_{final} = f \left(e^T T_4^{[1:k]} h_{sen} + W \begin{bmatrix} e \\ h_{sen} \end{bmatrix} + b \right) \quad (4)$$

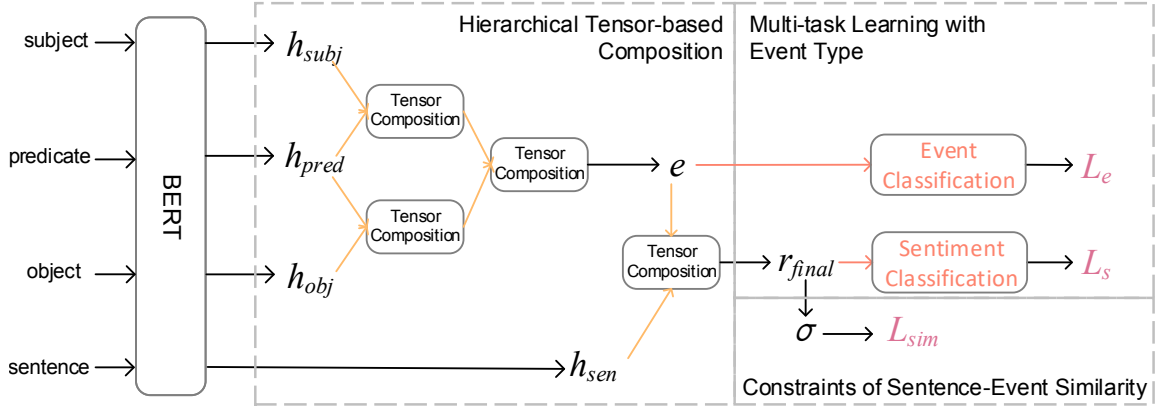


Figure 2: The event-centered text representation model with hierarchical tensor-based compositions.

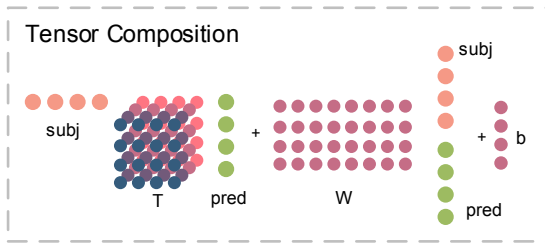


Figure 3: Diagram of Tensor Composition.

4.4 Multi-task Learning with Event Type Classification

Since each sentence in our dataset is annotated with an event type and a sentiment polarity category, we use a multi-task learning framework to make use of event type information in sentiment classification. Two tasks, event type classification and sentiment classification, share the same encoder and hierarchical Tensor-based Compositions. The event type classifier applies e to generate the event distribution, while the sentiment classifier uses the final representation r_{final} to compute the sentiment distribution.

The event type classification loss L_e is defined as:

$$L_e = -\frac{1}{N} \sum_{i=1}^N \sum_{e=1}^E y_i^e \log p(c_e | x_i) \quad (5)$$

where N and E denote the total number of training examples and event types respectively, $p(c_e | x_i)$ is the predicted probability of the i -th example x_i belonging to class c_e , and y_i^e is the ground truth label.

The sentiment classification loss L_s is defined in a similar way except that K denotes the total number of sentiment categories, which is 3 in our

dataset:

$$L_s = -\frac{1}{N} \sum_{i=1}^N \sum_{s=1}^K y_i^s \log p(c_s | x_i) \quad (6)$$

The multi-task framework is trained by minimizing the cross entropy loss of event classification L_e and sentiment classification L_s .

4.5 Constraint of Sentence-Event Similarity

Since the same event can be described in sentences with different surface forms, we assume that the sentence representation follows a Gaussian distribution with its mean determined by the event representation: $h_{sen} \sim \mathcal{N}(e, \sigma^2)$, where the variance σ can be computed based on the output of hierarchical Tensor-based Compositions.

$$\sigma = MLP(r_{final}) \quad (7)$$

We design a loss function to encourage the sentence representation to be close to the event representation, based on the probability density function of the Gaussian distribution:

$$\mathcal{L}_{sim} = \frac{1}{T} \sum_t \frac{1}{2\sigma_t^2} \|h_{sen} - e\|^2 + \frac{1}{2} \log \sigma_t^2 \quad (8)$$

where $\log \sigma_t^2$ is a variance regularization term that prevents the module from predicting too large variance σ .

Finally, the combined loss function is defined as follows:

$$\mathcal{L} = \lambda_e \mathcal{L}_e + \lambda_s \mathcal{L}_s + \lambda_{sim} \mathcal{L}_{sim} \quad (9)$$

where λ_e , λ_s , and λ_{sim} are hyperparameters controlling the relative contribution of the respective loss term.

5 Experiments

5.1 Datasets and Evaluations

To evaluate our proposed approach, we conducted experiments on the following two corpora: **EveSA**, the dataset we presented in Section 3 for Event-driven Sentiment Analysis, and **SemEval17 Task4** (Rosenthal et al., 2017). The SemEval-2017 Task 4 Subtask A aims to identify the overall sentiment of a tweet. The data was first cleaned by removing the hashtags, user mentions, irregular phrases and abbreviations. The original dataset does not have the annotations of event triplets. Existing event extraction models are mainly trained on the ACE 2005 dataset (Grishman et al., 2005), where triggers and arguments of news events are quite different from that of personal social events mentioned in the dataset. Therefore, we used a Semantic Role Labeling model (He et al., 2017) to extract a predicate from each sentence and mark its subject and object manually. We manually checked more than 20,000 tweets in this dataset, and retained 7,117 tweets which contain the full event triplets `<subject, predicate, object>`.

Category	Dataset	
	EveSA	SemEval17 Task4
Positive	1205	3247
Neutral	624	2001
Negative	2152	1869
Total	3981	7117

Table 2: Statistics of the datasets.

The statistics of the two corpora used in our experiments are shown in Table 2. It should be noted that since we use the SRL model instead of event extraction model to annotate the **SemEval 17 Task 4** dataset, there are no event types annotated. Thus the multi-task learning module is skipped in the experiments on this dataset.

For both EveSA and SemEval17 Task4, we employ accuracy and weighted-F1 as evaluation metrics. The weighted-F1 is computed by:

$$F1_{weighted} = \frac{\sum_{i=1}^N weight_i \times F1_i}{N} \quad (10)$$

where N is the total number of class categories, and $weight_i$ represents the weight of the i -th class category, which is the distribution of the i -th class in the training set.

5.2 Implementation Details

We implement the models in Pytorch 1.4.0. The BERT encoder is fine-tuned in the training process. The dimension of the hidden states is 768 and the batch size is 32. Number of matrices in tensor compositions are set to 100. The number of epochs is 5. \mathcal{L}_e , \mathcal{L}_s and \mathcal{L}_{sim} are set to 5, 1, and 1, respectively. The loss function is minimized using Adam optimizer (Kingma and Ba, 2014) with a learning rate of $1e - 5$ and a dropout rate of 0.1. All the parameters are chosen based on a validation set which is 20% of their respective training set.

5.3 Compared Methods

Since our model takes both event and sentence as input, we compare the proposed model with baselines taking three kinds of inputs:

Sentence as input Models in this category take a tweet (i.e., a sentence) as input:

BiLTSM (Schuster and Paliwal, 1997): A bidirectional LSTM neural network with GloVe embeddings (Pennington et al., 2014) for word sequence. **BERT** (Devlin et al., 2018): The state-of-the-art model for sentiment classification. We limit the sentence length to the last 40 tokens to allow a larger batch size. We use BERT-base due to the memory limit of our GPU.

BiLSTM+ orthogonalAtt (Wei et al., 2020): A recently proposed implicit sentiment analysis model with bidirectional LSTM neural network with BERT encoder and orthogonal attention.

Event triplet as input Models in the category take the event triplet in the form of `<subject, predicate, object>` as input:

The first two baselines, **BiLSTM** and **BERT**, take the concatenated words in subject, predicate and object as an input sequence.

NTN (Weber et al., 2017): A neural network with tensor-based compositions with subject, predicate and object in event triplet as inputs. We use GloVe embeddings (Pennington et al., 2014) for the word sequence.

BERTNTN: An NTN model with BERT as an encoder, using fine-tuned BERT to get hidden states of the word sequence.

Sentence and event triplet as input Models in this category consider both the input sentence and the extracted event triplet:

BERTNTN + Att (Vaswani et al., 2017): Built upon **BERTNTN**, the sentence is also encoded

Input	Model	EveSA				SemEval17 Task4			
		acc	P	R	F1	acc	P	R	F1
Sentence	BiLSTM	0.5670	0.4463	0.5670	0.4915	0.5864	0.5697	0.5864	0.5704
	BERT	0.7297	0.7260	0.7297	0.7239	0.7584	0.7530	0.7584	0.7528
	BiLSTM + orthogonalAtt	0.7459	0.7372	0.7459	0.7361	0.7563	0.7605	0.7563	0.7580
Event	BiLSTM	0.5645	0.4557	0.5645	0.4790	0.5414	0.5288	0.5414	0.4914
	NTN	0.6047	0.5622	0.6047	0.5570	0.5400	0.5504	0.5440	0.5035
	BERT	0.6438	0.6272	0.6438	0.6297	0.6163	0.6197	0.6163	0.6128
	BERTNTN	0.6754	0.6546	0.6754	0.6438	0.6233	0.6175	0.6233	0.6164
Sentence + Event	BERTNTN + Att	0.7419	0.7321	0.7419	0.7316	0.7547	0.7522	0.7547	0.7531
	BERTNTN \oplus sentence	0.7519	0.7569	0.7519	0.7307	0.7590	0.7578	0.7590	0.7582
	BERTNTN \oplus sentence +Att	0.7293	0.7260	0.7293	0.7180	0.7583	0.7549	0.7583	0.7555
	BERTNTN + Event Embedding	0.7218	0.7059	0.7218	0.7047	-	-	-	-
Ours (Sentence + event)	HTC + Sen-Event	0.7569	0.7486	0.7569	0.7494	0.7716	0.7700	0.7716	0.7690
	HTC + MTL+ Sen-Event	0.7657	0.7596	0.7657	0.7605	-	-	-	-

Table 3: Overall accuracy and weighted-F1 results.

with BERT. The results of NTN serves as queries, and hidden states of the sentence are keys and values to get the final attention output.

BERTNTN \oplus sentence: The event triplet and the sentence are fed into a BERT encoder. Then the BERT-encoded hidden state of the subject, predicate and object is concatenated with the sentence separately. The three concatenations are then fed into NTN.

BERTNTN \oplus sentence + Att: On the basis of **BERTNTN \oplus sentence**, the final attention output is obtained similar to **BERTNTN + Att**.

BERTNTN + Event Embedding (Tang et al., 2015b): Similar to **BERTNTN + Att**, each event type has its randomly initialized event embedding, similar to deriving user embeddings.

Due to the lack of event type labels in SemEval17 Task4, we are unable to conduct experiments using models making use of event types.

Experimental results on two corpora are shown in Table 3. It can be observed that: (1) The proposed model performs remarkably better than other baselines across all input categories and on all evaluation metrics, including a recently proposed implicit sentiment analysis model, which verifies the effectiveness of our model in capturing events that evoke sentiments in sentences. (2) Event representation in the form of event triplet leads to improved performance in sentiment classification, as evidenced by the lower performance of **BERT** (using sentence as the only input) compared to that of **BERTNTN + Att** and **BERTNTN \oplus sen-**

tence (encoding both sentence and event triplet). It should be noted that models using event triplets as the only input, such as **BERT** (using event triplet concatenation as input) and **NTN** give inferior results because the sentence contextual information is not modeled. (3) Event type, the extra annotated information in our dataset EveSA, plays a vital role in joint sentiment and event type prediction, since **HTC + MTL+ Sen-Event** outperforms **HTC + Sen-Event**.

5.4 Ablation Study

To validate the effectiveness of components in our approach, we performed ablation experiments on two corpora, and showed the ablation results in Table 4 and Table 5, respectively.

BERT only takes the sentence input, while BERTNTN only uses the event triplets as input. Event \oplus sentence concatenates the outputs of BERT and BERTNTN for classification. HTC, MTL and Sen_Event refer to hierarchical tensor-based compositions, multi-task learning with event type classification and the constraint of sentence-event similarity component, respectively. The full model are **HTC + MTL + Sen_Event** for EveSA, and **HTC + Sen_Event** for SemEval17 Task4.

It can be observed that using event triplets as input only (BERTNTN) gives the worst results. BERT trained on tweets directly improves upon BERTNTN quite substantially. Simply combining outputs from BERT and BERTNTN performs worse than BERT. Using the hierarchical tensor compositions (HTC) is more effective in encod-

Event < subject, predicate, object >	Event Type	Sentence	BERT → Ours
< The waterfront, offers, unparalleled views of the harbor and Hong Kong Island >	Offering	[S1] The waterfront here offers unparalleled views of the harbor and Hong Kong Island.	neu → pos
< it, abandoned, by its originators >	Abandonment	[S2] There were, ultimately, some findings that were distinctly incompatible with the theory and it was abandoned by its originators.	neu → neg
< Most of what the government does, serving, the public 's will >	Assistance	[S3] It isn't as easy to summarize federal regulation as it is to summarize federal spending , but the basic point is similar: Most of what the government does is actually serving , not opposing , the public 's will .	neu → pos
< US Department of Energy report, predicted, output >	Expectation	[S4] A US Department of Energy report has predicted output to rise from 890 million tonnes to 1,240 million tonnes in the next 20 years.	neu → pos
< He, affirmed, his support for the new fourth clause of the coronation oath >	Affirm_or_deny	[S5] He is said to have affirmed his support for the new fourth clause of the coronation oath at this time.	neg → pos
< he, revered, by the Surrealists >	Judgment	[S6] In the Twenties , he was an inspiration to the young muralists Diego Rivera and Jose Clemente Orozco and was revered by the Surrealists .	neg → pos

Figure 4: Case study of example classification outputs by BERT and our model. The event information including event triplet and event type is also shown. Instances incorrectly predicted by BERT but correctly predicted by our approach are selected. The last column, “BERT → Ours”, shows the prediction of BERT vs. the prediction of our approach.

Model	acc	P	R	F1
BERT	0.7297	0.7260	0.7297	0.7239
BERTNTN	0.6754	0.6546	0.6754	0.6438
Event \oplus sentence	0.7206	0.7174	0.7206	0.7125
HTC	0.7494	0.7459	0.7494	0.7425
HTC + MTL	0.7594	0.7530	0.7594	0.7509
HTC + Sen_Event	0.7569	0.7486	0.7569	0.7494
HTC + MTL + Sen_Event	0.7657	0.7596	0.7657	0.7605

Table 4: Ablation results of EveSA.

Model	acc	P	R	F1
BERT	0.7584	0.7530	0.7584	0.7528
BERTNTN	0.6233	0.6175	0.6233	0.6164
Event \oplus sentence	0.7456	0.7416	0.7456	0.7370
HTC	0.7639	0.7641	0.7639	0.7638
HTC + Sen_Event	0.7716	0.7700	0.7716	0.7690

Table 5: Ablation results of SemEval 2017.

ing both sentence contextual information and event triplets, outperforming BERT. Multi-task learning further improves the performance of HTC. Finally, the combination of all three components (or the two components without MTL for SemEval 2017) described Section 4 achieves the best results on the datasets.

5.5 Case Study

To further analyze whether the event information is the key factor to evoke sentiment, and to provide interpretable results for the prediction of our event-

centered text representation model, we compare the classification results of our proposed method with the results of BERT through a case study. The former employs both event and sentence as features, while the latter achieves state-of-the-art in sentence-level text classification without considering events which trigger sentiments expressed in text. Concrete descriptions of cases are shown in Figure 4 in which we show the instances that BERT fails to predict while ours gives correct results.

It can be seen from the examples [S1] to [S4] that in the absence of sentiment words, BERT tends to classify sentences as neutral. On the contrary, by utilizing event representations and jointly performing event type classification and sentiment analysis, our model benefits from emotional semantics contained in event triplets such as <Most of what the government does, serving, the public’s will > and the event type *Assistance*, and achieves better results in such sentences with implicitly expressed emotions. [S5] and [S6] are examples misjudged by BERT as *negative* but are correctly classified by our method. It can be observed that the event information is indeed helpful in implicit sentiment analysis.

6 Conclusion

In this paper, we propose a novel approach for fact-based implicit sentiment analysis, mainly for the situation where a sentence contains no sentiment words but only event descriptions. Our model employs event information, including the event triplets and event types as features, and detects sentiment

based on event representations learned with hierarchical tensor-based compositions. Moreover, we present a dataset with event annotations for implicit sentiment analysis. Experimental analysis demonstrates that both event triplets and event type benefit implicit sentiment classification. Our current approach assumes that events have already been extracted from text. Future research will explore automated event extraction in order to perform end-to-end implicit sentiment analysis.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments and helpful suggestions. This work was funded by the National Natural Science Foundation of China (61772132), and the UK Engineering and Physical Sciences Research Council (grant no. EP/T017112/1, EP/V048597/1). YH is supported by a Turing AI Fellowship funded by the UK Research and Innovation (grant no. EP/V020579/1).

References

- Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, pages 30–38.
- Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.
- Alexandra Balahur, Jesús M Hermida, and Andrés Montoyo. 2011. Detecting implicit expressions of sentiment in text based on commonsense knowledge. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 53–60.
- Lingjia Deng and Janyce Wiebe. 2015. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 179–189.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Haibo Ding and Ellen Riloff. 2016. Acquiring knowledge of affective events from blogs using label propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Haibo Ding and Ellen Riloff. 2018a. Human needs categorization of affective events using labeled and unlabeled data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1919–1929.
- Haibo Ding and Ellen Riloff. 2018b. Weakly supervised induction of affective events by optimizing semantic consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Radhika Gaonkar, Heeyoung Kwon, Mohaddeseh Bastan, Niranjan Balasubramanian, and Nathanael Chambers. 2020. Modeling label semantics for predicting emotional reactions. *arXiv preprint arXiv:2006.05489*.
- Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics*, pages 503–511.
- Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu’s english ace 2005 system description. *ACE*, 5.
- Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what’s next. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- Jan Hofmann, Enrica Troiano, Kai Sassenberg, and Roman Klinger. 2020. Appraisal theories for emotion classification in text. *arXiv preprint arXiv:2003.14155*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval ’17*, Vancouver, Canada. Association for Computational Linguistics.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

- Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934. Citeseer.
- Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015a. Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.
- Duyu Tang, Bing Qin, and Ting Liu. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- N. Weber, N. Balasubramanian, and N. Chambers. 2017. Event representations with tensor-based compositions.
- Jiyao Wei, Jian Liao, Zhenfei Yang, Suge Wang, and Qiang Zhao. 2020. Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing*, 383:165–173.
- Yang Yang, Deyu Zhou, Yulan He, and Meng Zhang. 2019. Interpretable relevant emotion ranking with event-driven attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 177–187.
- Lei Zhang and Bing Liu. 2011. Identifying noun product features that imply opinions. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies*, pages 575–580.
- Yuan Zhuang, Tianyu Jiang, and Ellen Riloff. 2020. Affective event classification with discourse-enhanced self-training.
- Enguang Zuo, Hui Zhao, Bo Chen, and Qiuchang Chen. 2020. Context-specific heterogeneous graph convolutional network for implicit sentiment analysis. *IEEE Access*, 8:37967–37975.