IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE AND COGNITIVE COMPUTING FOR COMMUNICATION AND NETWORK

Received December 18, 2018, accepted January 15, 2019, date of publication January 31, 2019, date of current version February 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2893806

Enhancing Attention-Based LSTM With Position Context for Aspect-Level Sentiment Classification

JIANGFENG ZENG[1], XIAO MA[2], AND KE ZHOU[1], (Member, IEEE)
[1]Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China
[2]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

Corresponding author: Ke Zhou (k.zhou@hust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61232004, Grant 61802440, and Grant 61502189, and in part by the National Key Research and Development Program of China under Grant 2016YFB0800402.

ABSTRACT Aspect-level sentiment classification is an interesting but challenging research problem, namely, the prediction of the sentiment polarity toward a specific aspect term of an opinionated sentence. Previous attention-based recurrent neural networks have been proposed to address this problem because attention mechanism is capable of finding out those words contributing more to the prediction than others and have shown great promise. However, the major drawback of these attention-based approaches is that the explicit position context is ignored. Drawing inspirations from the manner modeling the position context in information retrieval and question answering, we hypothesize that we should pay much more attention to the context words neighboring to the aspect than those far away, especially when one review sentence is a long sequence or contains multiple aspect terms. Based on this conjecture, in this paper, we put forward a new attentive LSTM model, dubbed PosATT-LSTM, which not only takes into account the importance of each context word but also incorporates the position-aware vectors, which represents the explicit position context between the aspect and its context words. We conduct substantial experiments on the SemEval 2014 datasets, and the encouraging results indicate the efficacy of our proposed approach.

INDEX TERMS Data analysis, natural language processing, sentiment classification, aspect-level, recurrent neural networks, long short-term memory, position context, attention mechanism.

I. INTRODUCTION

In this work we are dedicated in mining the sentiment expressed towards a specific aspect term (not aspect category) explicitly appearing in an opinionated text, which is the so-called aspect-level sentiment classification [1], [2]. For instance, in the opinionated review on a restaurant ''the food tastes so yummy, but the price is too high.'', ''food'' and ''price'' are two aspects of the restaurant, and the sentiment polarity towards aspect ''food'' is positive while the polarity towards aspect ''price'' is negative.

Aspect-level sentiment classification is a fine-grained NLP problem and has attracted a myriad of interest in the research community these years. While the standard sentiment classification entails assigning class labels to each sentence, it doesn't care whether there exists different aspects in one sentence. Aspect-level sentiment classification is considerably more difficult than sentence-level sentiment classification because it necessitates distinguishing which part of the sentence describes the corresponding aspect.

In the literature, there are two general paradigms in existing representative approaches: traditional machine learning models and neural network models. Traditional machine learning methods are dedicated to manually extracting an abundance of features like bag-of-words and sentiment lexicons which are used to train a sentiment classifier such as SVM [3]–[7]. However, effective features usually need to be built upon domain expert knowledge, which is labor-intensive and complicated. It is known to all that deep learning models win reputations because they automatically learn semantic representations from high dimensional original data without carefully designed feature engineering. This is the main reason why deep learning techniques are superior to traditional machine learning techniques [8]–[10]. As a powerful technique for sequence modeling, recurrent neural networks (RNNs) have been widely applied to a variety of NLP tasks [11]–[14]. Some RNN-based approaches have been investigated for sentiment analysis [15]–[17]. However, intuitively only some subset of context words are crucial

to inferring the sentiment towards an aspect. Inspired by the visual attention mechanism belonging to humans, attention mechanisms implemented by neural networks have been well-studied in substantial applications, such as image generation [18], [19], image caption [20], machine translation [21], [22], natural language inference [23] and text hashing [24]. There are already some works using attention mechanism to deal with aspect level sentiment analysis [25]–[29]. Compared with traditional machine learning models, neural networks models have achieved promising results on sentiment analysis for their capability to learn powerful and semantic feature representations from original data without carefully hand-crafted feature engineering.

To our best knowledge, all the aforementioned RNN models turn a blind eye to the position context since, in theory, sequence models are sensitive to word order. However, we argue that word order is incapable of elaborately revealing the position information between the given aspect and each of its context words. In other words, word order is not identical to the explicit position context. Based on this insight, in this paper, we design a new attentive LSTM model, dubbed PosATT-LSTM, which not only takes into account the importance of each context word but also incorporates the position-aware vectors which represents the explicit position context between the aspect and its context words. The most similar work to ours is that of [30] which incorporates the position context of the question words into the answer's attentive representations. Let's take a long sequence ''although the restaurant is blamed for its awful service, the reason why I always come to this restaurant is that the food are clearly among the best in the city and the price is affordable.'' as an example. As a man with normal intelligence, it is apparent that the sentiment polarities of ''service'', ''food'' and ''price'' are negative, positive and positive respectively because they are depicted by ''awful'', ''best'' and ''affordable'' respectively. As for ''service'', the distances between the aspect and the three instructive context words are -1, 19 and 27. The distances are -15, 5 and 13 for ''food'', and -26, -6, 2 for ''price''. It is clear that the corresponding context words of great importance are usually close to the given aspect. The intuition behind this is that the context words neighboring to the aspect should be paid much more attention since they are much more valuable than those far away, especially when one review sentence is a long sequence or contains multiple aspect terms. Consequently, the most important and challenging issues in our model are: (1) how to model the position context, and (2) how to exploit the position-aware vectors to enhance attention-based LSTM networks for aspect-level sentiment classification. In terms of the first issue, we model position-aware influence vectors with the Gaussian kernel. As far as I know, there exist some convolutional neural network (CNN) models addressing sequential tasks which simply concatenate the distance information into the input word vectors and get the performance improved [31], [32]. However, this manner doesn't match RNNs well perhaps because RNN models are instinctively capable of capturing word order within a sequence. Thus, we append position-aware influence vectors into the hidden representations of the context words on top of LSTM layer. Finally, attention mechanism is used to compute weights between the aspect embedding and the concatenated representations for ultimate aspect-specific attentive representations.

To sum up, our contributions are three-folds.

- In order to generate the aspect-specific attentive representations, we devise a novel attention-based Long Short-Term memory network for aspect-level sentiment classification. The model computes attention weights relying on both the importance of each context word and the position context between the aspect and its context words.
- To better make advantage of the explicit position context, we design position-aware influence vectors with the Gaussian kernel.
- Since RNN models are theoretically aware of word order in the sequence, it doesn't make any sense to concatenate the position vector into the input word vectors. Therefore, position-aware influence vectors are appended into the hidden representations of the context words on top of LSTM layer. At last, the ultimate aspect-specific attentive representations are obtained via computing attention weights between the aspect embedding and the concatenated representations.
- Compared with several baselines, we conduct qualitative experiments on two real-world datasets and the results evaluate the effectiveness of our proposed method for aspect-level sentiment classification.

The rest of this paper is organized as follows. We first briefly review the related work of aspect-level sentiment classification and survey the manners modeling position information in the field of NLP in section II. Afterwards, the proposed attentive model (PosATT-LSTM) is presented in section III. In section IV, extensive comparison experiments are conducted to prove the superiority of our proposed algorithm. At last, we draw a conclusion and envision the future in section V.

II. RELATED WORK

Sentiment analysis is a big suitcase since it handles many subproblems involved in extracting meaning and polarity from text. The majority of current approaches attempt to detect the overall sentiment polarity expressed in an opinionated text but ignore the target entities mentioned (e.g., MacBook, Mcdonalds) or their aspects (e.g., price, food taste, service). As investigated in the work of [33], Jiang *et al.* surprisedly find that ignoring the target entities or their aspects discussed in the opinionated text results in 40% of sentiment classification errors. Consequently, as fine-grained branches of sentiment analysis, target dependent sentiment analysis and aspect based sentiment analysis draw a lot of attention these years. As is often the case, approaches qualified for the task of target dependent sentiment analysis can be naturally transferred to the task of aspect based sentiment analysis.

As a result, in the following, we first survey aspect based sentiment analysis, and then include a more comprehensive study on the manner of modeling position information.

A. ASPECT BASED SENTIMENT ANALYSIS

Aspect based sentiment analysis can be divided into four subtasks: (1) Aspect term extraction aims to find out the aspect terms commented in the opinionated text; (2) Aspect term polarity knows the aspect terms discussed in the opinionated text and aims to determine the sentiment polarity towards each aspect term; (3) Aspect category detection identifies the aspect categories discussed in the opinionated text; (4) Aspect category polarity knows the aspect categories discussed in the opinionated text and aims at distinguishing the sentiment polarity towards each aspect category. Notice that an aspect term means to be a particular aspect of the target entity and explicitly appear in the opinionated text while aspect categories, typically much coarser compared with aspect terms, do not necessarily occur as part of the opinionated text. In this paper, we try to tackle the subtask 2.

Earlier methods for aspect-level sentiment analysis highly depend on the quality of the handcrafted feature engineering [34], [35]. Deep neural networks (DNNs), characterized by automatic representation learning, release the burden of harbor-intensive feature engineering. Over the past several years, recurrent neural networks (RNNs), especially long short-term memory networks (LSTMs) have shown a striking promise in the field of natural language processing (NLP). Dong *et al.* [15] devise an adaptive recursive neural network by modeling syntactic relations on tweet data. Poria *et al.* [36] propose a deep CNN model to extract aspects for opinion mining. Tang *et al.* [17] propose TD-LSTM and TC-LSTM both of which split the whole context into two components, i.e., left part with target and right part with target, followed by two long short-term memory networks to compute the target representations in forward and backward direction respectively. The authors concatenate the two target-specific representations as input for sentiment classification. The only difference between TD-LSTM and TC-LSTM is that TC-LSTM takes as input the concatenation of each word vector and target vector to incorporate the semantic relatedness of target with its context words. In the work of [37], a two stage approach is developed, i.e., aspect term extraction and sentiment classification. Years of research has witnessed the great success achieved by incorporating attention mechanism into deep neural networks. Attention mechanism has also been investigated for aspect-level sentiment classification. Motivated by deep memory networks [38], Tang *et al.* [39] attempt to address this issue using deep memory networks with multiple computational layers. Wang *et al.* [25] put forward aspect embedding which treats the aspect representations as training parameters and design several attention-based LSTM networks to capture the relevance between the aspect and its context words. Since all the previous models overlook the separate modeling of aspects, Ma *et al.* [26] utilize two attention-based LSTM networks to represent the aspect and its context interactively. The final sentiment classifier takes as input the combination of aspect representation and its context representation. In order to capture sentiment features separated by a long distance, Chen *et al.* [27] devise a deep model using multiple attention mechanisms. Ma *et al.* [40] attempt to model multi-aspects within one sentence at one time. Moreover, Saeidi *et al.* [41] and Ma *et al.* [28] are dedicated to tackling the challenges of both aspect based sentiment analysis and target dependent sentiment analysis. The latter proposes an attention-based hierarchical approach by further incorporating affective commonsense knowledge.

B. MODELING POSITION INFORMATION

Some researchers have realized the importance of position information, and model position information to improve performance for many NLP tasks. The manners to model position information can be roughly grouped into two categories: using a pre-defined strategy and leaning as model parameters. The first option computes a location vector for each context word using a pre-defined strategy. For example, Sukhbaatar *et al.* [38] develop end-to-end memory networks for question answering and give an equation to compute location vectors. In order to capture the order of the sequence, Vaswani *et al.* [42] use sine and cosine functions of different frequencies to compute positional encodings which are added to the input word embeddings. Liu *et al.* [43] and Chen *et al.* [30] model the position context using the Gaussian kernel in information retrieval (IR) and question answering (QA), respectively. The second option regards position vectors as model parameters to be learned during training. To offer substitutes for the famous encoder-decoder framework when addressing sequential data, Gehring *et al.* [44] introduce an architecture based entirely on convolutional neural networks, and defines position embeddings which are learned as model parameters during training. Note that Vaswani *et al.* [42] also experimented using learned position embeddings from [44] instead, and found that the two manners produced nearly identical results. Along this line, there are two studies attempt to introduce the position information for aspect-level sentiment classification. In the work of [39], Tang *et al.* introduce location attention using both simple pre-defined strategies and learned as model parameters four manners to model position information: the first two methods use the equation in [38] and a simplified version of it; and the remaining two methods regard position vectors as model parameters like [44]. However, no remarkable performance gain is obtained using such location attention. To generate position-weighted memories, Chen *et al.* [27] calculate the position weight for each context word using a pre-defined equation.

III. THE PROPOSED APPROACH

In this section we try our best to figure out the proposed approach for aspect-level sentiment classification and a high-level illustration of our proposed model is shown in Figure 1. We first formulate the problem of aspect-level

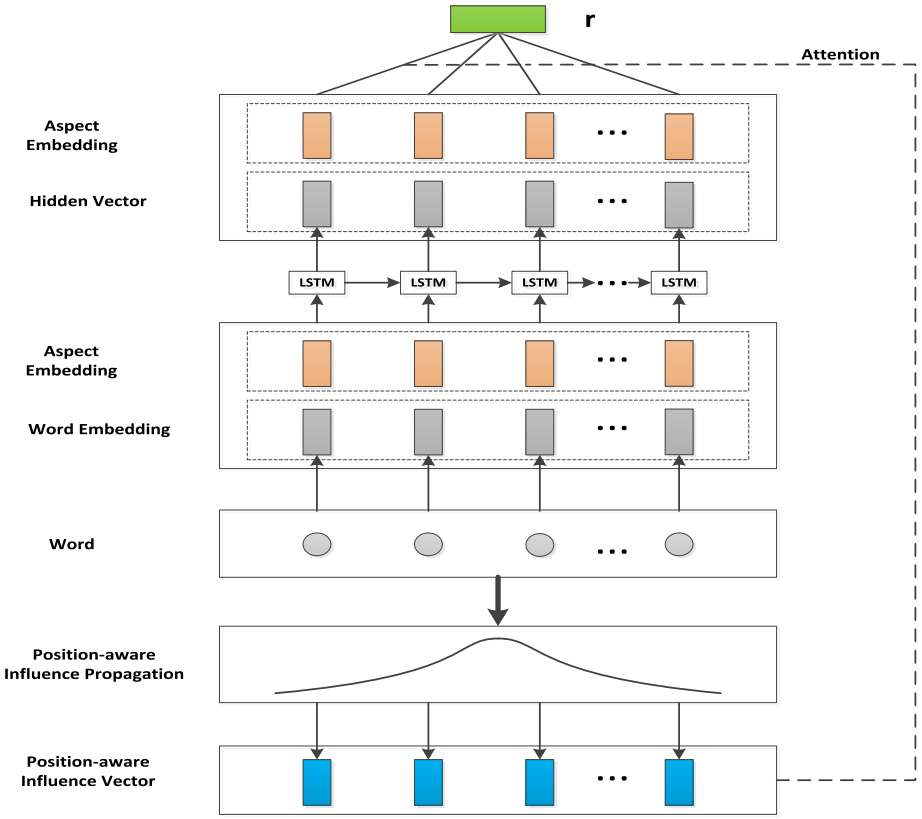


FIGURE 1. Overview of the proposed model for aspect-level sentiment classification.

sentiment classification. Then, a brief survey on Long Short-Term Memory networks (LSTMs) is given because a LSTM is used for sentence modeling in our implementation. Afterwards, we attempt to tackle two challenging issues: (1) how to model the position context, and (2) how to exploit the position-aware vectors. At last, the training details of our model are presented.

A. TASK DEFINITION

Suppose a sentence S contains n words and mentions m aspect terms. Taking the (aspect, sentence) pairs as input, aspect-level sentiment classification aims at predicting a sentiment category for each (aspect, sentence) pair. For instance, the sentence "Other than not being a fan of [click pads] and the lousy [internal speakers], it's hard for me to find things about this notebook I don't like, especially considering the $350 [price tag]." involves three aspect terms, i.e., [click pads], [internal speakers] and [price tag]. We first arrange the sentence into three (aspect, sentence) pairs as instances, and then infer a sentiment category for each instance. The desired outputs for [click pads], [internal speakers] and [price tag] are **negative**, **negative** and **positive**, respectively.

B. LONG SHORT-TERM MEMORY (LSTM)

Recurrent neural networks (RNNs) [11], [45], [46] have become a cornerstone for many natural language processing (NLP) applications. As a special kind of RNN, Long Short-Term Memory network (LSTM) is customized to deal with the long-term dependencies since the standard RNN suffers from gradient vanishing or exploding problem [47]. The key idea behind LSTM is that the cell state is regulated via three gates at each time step, each of which is implemented using sigmoid function.

Verified by the work of [33], aspect information plays a vital role in classifying the polarity when addressing aspect-level sentiment classification. However, most previous studies [17], [39] average all the aspect or target word representations as the representations of aspects or targets, which fails to actually represent aspects and leads to badly predict the sentiment polarities of these aspects. Thus, Wang *et al.* [25] proposed to learn an embedding vector for each aspect which is the so-called aspect embedding. Notice that aspect embeddings are regarded as model parameters to be learned during training. In our work, the LSTM network takes in both the pre-trained word embeddings like *word2vec* and aspect embeddings as input. Specifically, let us first formalize the notations. All the word vectors are stacked in a word embedding matrix $L \in R^{d\times|V|}$, where d is the dimension of word vector and $|V|$ is vocabulary size. All the aspect embeddings are denoted as a matrix $A \in R^{d_a\times|A|}$, where d_a is the dimension of aspect vector and $|A|$ is the size of aspect terms. Formally, given the input word embedding $w_t \in R^d$ from L which vectorizes the t-th word of a sentence and the given aspect embedding $v_a \in R^{d_a}$ from A, current cell

state c_t and current hidden vector h_t in a standard LSTM can be updated as follows:

$$i_t = \sigma(W_i^w \cdot [w_t, v_a] + W_i^h \cdot h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f^w \cdot [w_t, v_a] + W_f^h \cdot h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o^w \cdot [w_t, v_a] + W_o^h \cdot h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c^w \cdot [w_t, v_a] + W_c^h \cdot h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \odot tanh(c_t) \quad (5)$$

where i, f and o are input gate, forget gate and output gate respectively, which control the inflow of valuable information and the outflow of useless information at each time step via the sigmoid function, i.e., σ in the equations. $\Theta^{lstm} = \{W_i^w \in R^{d\times(d+d_a)}, W_f^w \in R^{d\times(d+d_a)}, W_o^w \in R^{d\times(d+d_a)}, W_c^w \in R^{d\times(d+d_a)}, W_i^h \in R^{d\times d}, W_f^h \in R^{d\times d}, W_o^h \in R^{d\times d}, W_c^h \in R^{d\times d}, b_i \in R^d, b_f \in R^d, b_o \in R^d, b_c \in R^d\}$ are denoted as the parameters of LSTM to be learned during training. $\odot$ stands for element-wise multiplication and $[w_t, v_a]$ means the concatenation operation. Then, we obtain the hidden vectors $[h_1, h_2, ..., h_n]$ as the final word representations of the context words with length n.

Many LSTM-based approaches regard the last hidden vector h_n or the average pooling from all the hidden values as the sentence embedding. Note that the benchmark approach that only uses LSTM in section IV regards the last hidden vector h_n as the representation of sentence which is taken as input by a softmax layer for sentiment classification.

C. POSITION-AWARE INFLUENCE VECTOR

Based on our previous assumption, the polarity of a given aspect is doomed to be severely affected by its neighboring context words. In this paper, the main reasons why we choose the Gaussian kernel to model the explicit position context between the aspect and its context words are: (1) The Gaussian kernel used by [30] achieves a remarkable performance gain for RNN based model; (2) The position embeddings devised by [44] and the sine/cosine functions of different frequencies exploited by [42] are used to help non-RNN based methods (convolution neural networks and self-attention neural networks) capture the order of the sequence. We first compute the position-aware influence propagation with the Gaussian kernel:

$$Kernel(\mu) = \exp(\frac{-\mu^2}{2\gamma^2}) \quad (6)$$

where μ is the distance between the current context word and the aspect, and γ means to be the propagation scope. Evidently, the longer the distance is, the lower the position-aware influence will be exerted on the context word. In order to represent the influence in a high-dimensional space, we assume that the influence for a specific distance complies with the Gaussian distribution over each dimension. Then the influence matrix P is formulized as:

$$P(i, \mu) \sim N(Kernel(\mu), \sigma') \quad (7)$$

where $P(i, \mu)$ defines the influence in the $i-$th dimension as to the distance of μ, and N represents the normal distribution with mean value of $Kernel(\mu)$ and standard deviation of σ'. Each column of P denotes the influence vector corresponding to a specific distance. For simplicity, we denote $p_j \in R^{d_p}$ from P as the influence vector for the $j-$th context word with a distance of μ, and d_p is the dimension of position vector.

D. ATTENTION

Attention mechanism is one of the most exciting advancements in deep learning and has a long history. But only recently has attention mechanism made its way into recurrent neural networks (RNNs) architectures that are widely developed for a variety of NLP applications. Previous attention-based approaches often compute the attention distribution depending on the relation between the aspect vector and the hidden vectors generated by a sequential model like LSTM, while the explicit position information between the aspect and each context word has not been well investigated. Here, we compute attention weights by incorporating the explicit position context into the attentive representations. Specifically, given the aspect embedding v_a, the position-aware vector p_j and the hidden vectors $[h_1, h_2, ..., h_n]$ of the context words, the attention weight corresponding to the context word at position j in the sentence can be denoted as:

$$\alpha_j = \frac{\exp(e(h_j, v_a, p_j))}{\sum_{k=1}^{n} \exp(e(h_k, v_a, p_k))} \quad (8)$$

$$e(h_j, v_a, p_j) = \eta^T \tanh(W_h h_j + W_p p_j + W_a v_a + b) \quad (9)$$

where $e(h_j, v_a, p_j)$ calculates a score which measures the semantic relatedness between the $j-$th context word and the aspect through incorporating the position-aware vectors. We define $\Theta^{att} = \{W_h \in R^{d\times d}, W_p \in R^{d\times d_p}, W_a \in R^{d\times d_a}, b \in R^d, \eta \in R^d\}$ as the training parameters to be learned during training. With the obtained attention distribution, this opinionated sentence is represented by the weighted sum of all the hidden vectors:

$$r = \sum_{j=1}^{n} \alpha_j h_j \quad (10)$$

E. SENTIMENT CLASSIFICATION

Here, we train the sentiment classifier. First, a nonlinear layer is used to project the aspect-specific attentive representation r into the target space of C classes:

$$\hat{r} = tanh(W_r r + b_r) \quad (11)$$

where C is the number of sentiment classes, and $\Theta^{(classifier)} = \{W_r \in R^{d\times C}, b_r \in R^C\}$ is the parameters to be learned. Afterwards, a softmax layer is used to compute the sentiment distribution:

$$g_c = \frac{\exp(\hat{r}_c)}{\sum_{z=1}^{C} \exp(\hat{r}_z)} \quad (12)$$

TABLE 1. Statistics of SemEval 2014 datasets.

Dataset	Positive		Negative		Neural	
	Train	Test	Train	Test	Train	Test
Laptop	994	341	870	128	464	169
Restaurant	2164	728	807	196	637	196

F. MODEL TRAINING

In our work, we need to optimize all the parameters notated as $\Theta = \{\Theta^{lstm}, \Theta^{att}, \Theta^{(classifier)}, A\}$. Note that A is the aspect embeddings to be learned. Cross entropy with L_2 regularization is defined as the loss function for optimization when training:

$$L = -\sum_{d \in D} \sum_{c=1}^{C} y_c(d) \cdot \log(g_c(d)) + \lambda L_2(\Theta) \tag{13}$$

where D is the dataset, d is one sample, $y_c(d)$ is the golden sentiment distribution and λ is the coefficient for L_2 regularization.

IV. EXPERIMENTS

In this section, we present our experiment settings and conduct experiments on the task of aspect-level sentiment classification.

A. EXPERIMENTAL SETTINGS

We evaluate the proposed method on two real-world datasets, i.e., Laptop and Restaurant, all of which are from SemEval 2014.[1] The statistics of the used datasets are summarized in Table 1. Each dataset is split into train and test set. It can be seen that each review is labeled with three sentiment polarities: positive, negative and neutral. Note that as done by previous work [25], [26], [39], we remove the fourth category, i.e. conflict, because each dataset contains a very tiny number of this category instances. Overall, in our experiments we use 2966 reviews and 4728 reviews from the Laptop dataset and the Restaurant dataset, respectively. To be specific, the number in Table 1 means the number of reviews for training and test in each sentiment category.

In our experiments, we exploit the 300-dimensional word embeddings initialized by GloVe [48], and randomly initialize the aspect embeddings and the out-of-vocabulary word vectors from $U(-\epsilon, \epsilon)$, where $\epsilon = 0.01$. The dimension of aspect embeddings are 300 too. According to the experimental settings in the work of [30], we set the propagation scope γ in Equation 6 to be 25, and the standard deviation σ' in Equation 7 to be 0.1. The dimension of position-aware vectors is also 300. Our experiments are conducted with a batch size of 25 reviews, L_2-regularization weight of 0.00001 and initial learning rate of 0.05 for AdaDelta. We implement our PosATT-LSTM using Theano and the results are obtained by a fixed random seed (10001).

[1] alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

Evaluation metric used here is classification accuracy. *Accuracy* measures the overall sentiment classification performance, is formalized as:

$$Accuracy = \frac{T}{N} \tag{14}$$

where T is the number of samples correctly predicted and N is the total number of test dataset.

B. BASELINES

In order to comprehensively evaluate the performance of our proposed PosATT-LSTM, the following baseline methods are used for comparison.

- **Majority** infers the sentiment polarities of the test dataset according to the majority sentiment polarity in the training dataset.
- **LSTM** [25] models opinionated sentences with a LSTM network without considering aspect terms, and treats the last hidden vector as the sentence representation which is taken into input by a softmax layer for final classification.
- **TD-LSTM** [17] first decomposes a sentence into left part with target and right part with target, and then models the them with a left-directed LSTM and a right-directed LSTM respectively. The classifier takes as input the concatenation of the left and right aspect-dependent representations to predict the sentiment polarity of the target.
- **TC-LSTM** [17] is structurally similar to TD-LSTM [17] and the only difference is that it combines input word embedding and aspect vector (average over multiple word vectors) to enhance the importance of aspect representations.
- **AE-LSTM** [25] is first to devise aspect embeddings which treat aspect representations as a part of training parameters.
- **ATAE-LSTM** [25] appends the aspect embedding into each word embedding with the goal of reinforcing the importance of aspect information. Then attention weights are computed with the guidance of aspect embeddings. Note that ATAE-LSTM corresponds to the non-position version of our PosATT-LSTM.
- **MemNet** [39] uses a deep memory network to replace RNN-based methods for sentence modeling, and captures the relevance between each context word and the depicted aspect through multiple computational hops, each of which is a neural attention model over an external memory.
- **IAN** [26] not only models the context words with an attention-based LSTM network, but also uses another attention-based LSTM network to generate the aspect representation. The output of each LSTM network is obtained via interactive attention. Then the context representation and the aspect representation are concatenated for final classification.

TABLE 2. Sentiment classification results of our model against competitor models on laptop and restaurant. Accuracy is the used evaluation metric. Best results are in bold.

Methods	Laptop	Restaurant
Majority	0.650	0.535
LSTM	0.665	0.743
TD-LSTM	0.681	0.756
TC-LSTM	0.682	0.760
AE-LSTM	0.689	0.762
ATAE-LSTM	0.687	0.772
MemNet	0.703	0.781
IAN	0.721	0.786
PosATT-LSTM	**0.728**	**0.794**

C. MODEL COMPARISONS

The classification accuracy results of our model compared with other competitive models are shown in Table 2. Comparing Majority and other LSTM based methods, we can conclude that LSTM networks have shown huge superiority over Majority in sequence modeling because they are capable of effectively generating feature representations without handcrafted feature engineering. Among all the LSTM based methods, the basic LSTM approach performs worst because it never takes aspect information into consideration, which is pointed out in the work of [33]. TD-LSTM, TC-LSTM, AE-LSTM, ATAE-LSTM, IAN and our proposed PosATT-LSTM all incorporate aspect information into model building at different levels. TD-LSTM focuses the aspect twice while TC-LSTM takes as input the combination of aspect vector (average over multiple word vectors) and input word embedding, which results in a tiny performance gain. Considering that it's naive to represent the aspect containing multi-words with average over multiple word vectors, AE-LSTM devises an embedding vector for each aspect in an explicit manner. This is the main reason why AE-LSTM slightly exceeds TD-LSTM and TC-LSTM. Further more, ATAE-LSTM, MemNet, IAN and PosATT-LSTM stably beat TD-LSTM and AE-LSTM in that attention mechanism is exploited. MemNet wins ATAE-LSTM by 1.6% and 0.9% on the Laptop dataset and Restaurant dataset respectively because it computes attention weights via multiple computational hops. Compared with ATAE-LSTM, IAN achieves absolute increases of 1.4% and 3.4% on the Laptop dataset and Restaurant dataset respectively since two LSTM networks are designed to generate more reasonable aspect representation and context representation with interactive attention.

As we stated above, ATAE-LSTM corresponds to the non-position version of our PosATT-LSTM. It can be observed from Table 2 that compared with ATAE-LSTM, our PosATT-LSTM model brings absolute increments of 4.1% and 2.2% on the Laptop dataset and Restaurant dataset respectively. What accounts for such increases in performance is that we not only incorporate the aspect information, but also encode the explicit position context into the aspect-specific attentive representations. Therefore, our proposed PosATT-LSTM achieves the best performance among all the competitive baselines. On the other hand, compared with the current state-of-the-art IAN, absolute increases of 0.7% and 0.8% are achieved on the Laptop dataset and Restaurant dataset respectively.

TABLE 3. Comparison with different position encoding methods. PosATT-LSTM+PE uses position embeddings to replace our proposed position-aware vectors obtained using the Gaussian kernel.

Methods	Laptop	Restaurant
ATAE-LSTM	0.687	0.772
PosATT-LSTM+PE	0.689	0.775
PosATT-LSTM	**0.728**	**0.794**

TABLE 4. Effects of position-aware vectors. IAN+pos intends to strengthen the context representation in IAN with the explicit position context.

Methods	Laptop	Restaurant
IAN	0.721	0.786
IAN+Pos	0.726	0.793
PosATT-LSTM	**0.728**	**0.794**

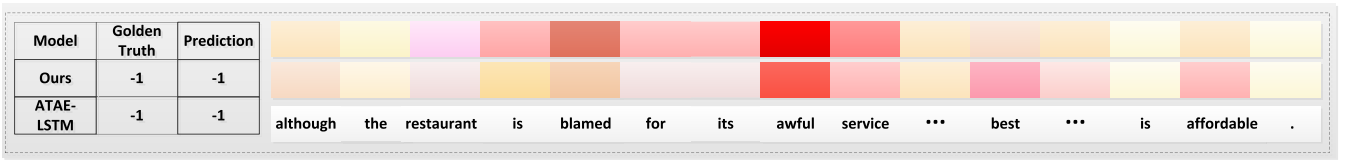
D. EFFECTS OF POSITION-AWARE VECTORS

In order to validate the advantage of modeling position information using the Gaussian kernel over other manners, we replace the position-aware vectors with position embeddings [39], [44] which are learned as model parameters during training. The model used for comparison is coined PosATT-LSTM+PE. Table 3 shows the results of ATAE-LSTM (the non-position version), PosATT-LSTM+PE (position embeddings) and PosATT-LSTM (position-aware vectors using the Gaussian kernel). From the results we can see that PosATT-LSTM+PE performs a little better than ATAE-LSTM but far below our proposed PosATT-LSTM.

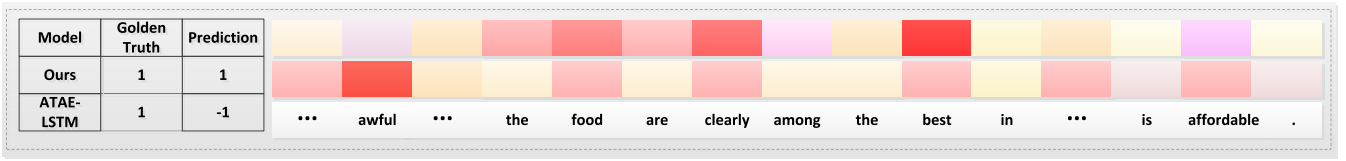
It is well accepted that IAN is the state-of-the-art approach for aspect-level sentiment classification before our work, and context representations are also generated by an attention-based LSTM similar to ours. We argue that strengthening the context representation in IAN with the explicit position context should get the performance boosted since modeling the explicit position information into our approach takes effect. Hence, an extra experiment is performed to verify the effectiveness of position-aware vectors and its accuracy results are presented in Table 4. It can be seen that IAN+Pos scores 0.5% and 0.7% higher on the Laptop and Restaurant dataset respectively. Now we can conclude that our designed position-aware vectors are effective when addressing sentiment-level classification. In our implements, IAN+Pos scores slightly worse than PosATT+LSTM. In addition, the number of training parameters of PosATT+LSTM is less than that of IAN+Pos. Thus we choose PosATT+LSTM as our proposal.

E. CASE STUDY

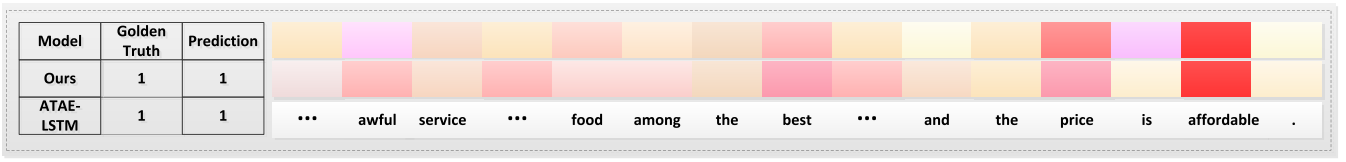
To have an intuitive understanding of the advantage of our proposed PosATT-LSTM over ATAE-LSTM (the non-position version), we apply the trained PosATT-LSTM and



(a)



(b)



(c)

FIGURE 2. Case Study: Illustration of attention weights on the words obtained by our PosATT-LSTM and ATAE-LSTM (the non-position version model). The deeper the color is, the higher the weight scores. (a), (b) and (c) show the results w.r.t. different aspect terms. In each subfigure, the left table shows the predictions of the two models and the right shows the attention weights on the words (the first row is produced by our PosATT-LSTM, the second row is produced by ATAE-LSTM, and the third row is the sentence). (a) Aspect term: *service*. (b) Aspect term: *food*. (c) Aspect term: *price*.

ATAE-LSTM to predict the polarities of three aspects, i.e., ''service'', ''food'', ''price'', in the opinionated sentence ''although the restaurant is blamed for its awful service, the reason why I always come to this restaurant is that the food are clearly among the best in the city and the price is affordable.''. Figure 2 visualizes the attention weights on the words computed by PosATT-LSTM and ATAE-LSTM. Obviously, it's difficult to infer the polarities of three aspects in a sequence of 37 words. We can see that PosATT-LSTM accurately predicts the sentiment polarities for ''service'', ''food'' and ''price'' because ''awful'', ''best'' and ''affordable'' are attended the most respectively while ATAE-LSTM makes a mistake when predicting the sentiment polarity for aspect term ''food''. It can be clearly observed that PosATT-LSTM concentrates on those words close to the given aspect term and reduces the influence of other sentiment words faraway. However, because of ignoring the explicit position context, ATAE-LSTM often chooses the sentiment lexicons appearing in the sentence to be focused, which lures the sentiment classifier to make erroneous judgements. To be specific, in terms of aspect ''food'', PosATT-LSTM focuses on three top-ranking words (''best'', ''clearly'' and ''food''), and pays little attention to farther sentiment lexicons like ''awful'', ''affordable''. By contrast, ATAE-LSTM attends the most on ''awful'', which leads to the misjudgement. Although ATAE-LSTM correctly infers the sentiment polarities for ''service'' and ''price'', it is disturbed worse by the sentiment words faraway than our proposed PosATT-LSTM. Therefore it can be concluded from the case study that PosATT-LSTM concerns much more around the given aspect term and reduces the impact of sentiment lexicons which are a bit farther from the given aspect. That is the main reason why our proposed approach beats all the competitive methods aforementioned.

F. ERROR ANALYSIS

According to the predictions on the testing dataset, we find our approach outperforms all the baselines when addressing sentences where multi-aspects are reviewed successively, which is displayed in the case study section. However, our approach still has limitations. An error analysis is carried out, and most of the errors could be summarized as follows. First, PosATT-LSTM performs bad facing long-distance dependencies between the aspect and the corresponding sentiment word. For instance, the aspect is placed in the beginning of the review, followed by a long string of words depicting other things, but commented at the end of the review. Second, PosATT-LSTM behaviors unstable when addressing cases where the sequential order of multi-aspects is misleading. Indeed, during experiments we find some prediction results of such misleading order cases are right. One possible explanation for this is that our method considers both the context semantics and the position information and balances between these two factors.

V. CONCLUSION

In this paper, we develop a new attention-based LSTM model, which not only takes into account the importance of each context word but also incorporates the explicit position information between the aspect and its context words into the

aspect-specific attentive representations. First, we model position-aware influence vectors with the Gaussian kernel just as processed in the field of information retrieval (IR). Then, we append both position-aware influence vectors and aspect embeddings into the sentence hidden representations on top of LSTM layer. In the end, substantial experiments have been conducted on two publicly open datasets, the results of which clearly demonstrate that the ultimate aspect-specific attentive representations improve the performance of aspect-level sentiment classification compared with several baselines.

Although it is empirically validated that our proposal has shown great potentials for aspect-level sentiment classification, this work overlooks the influence among different aspects when one opinionated sentence owns more than one aspect terms. Evidently, current deep attentive methods leave space for improvement. As a result, in the future, we intend to investigate the way of modeling more than one aspect simultaneously with our designed position attention mechanism.

ACKNOWLEDGEMENTS

It is highly appreciated for those helpful and thought-provoking advices of the anonymous reviewers.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, Jul. 2008.
[2] B. Liu, *Sentiment Analysis and Opinion Mining* (Synthesis Lectures on Human Language Technologies), vol. 5. San Rafael, CA, USA: Morgan & Claypool, 2012.
[3] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. ACL Conf. Empirical Methods Natural Lang. Process.*, vol. 10, Jul. 2002, pp. 79–86.
[4] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 412–418.
[5] N. Kaji and M. Kitsuregawa, "Building lexicon for sentiment analysis from massive collection of HTML documents," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, Jan. 2007, pp. 1075–1083.
[6] D. Rao and D. Ravichandran, "Semi-supervised polarity lexicon induction," in *Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Mar. 2009, pp. 675–682.
[7] V. Pérez-Rosas, C. Banea, and R. Mihalcea, "Learning sentiment lexicons in spanish," in *Proc. LREC*, May 2012, pp. 1–55.
[8] M. Xia, T. Li, T. Shu, J. Wan, C. W. de Silva, and Z. Wang, "A two-stage approach for the remaining useful life prediction of bearings using deep neural networks," *IEEE Trans. Ind. Inform.*, 2018.
[9] J. Wan, J. Yang, Z. Wang, and Q. Hua, "Artificial intelligence for cloud-assisted smart factory," *IEEE Access*, vol. 6, pp. 55419–55430, 2018.
[10] J. Zeng, X. Ma, and K. Zhou, "CAAE++: Improved CAAE for age progression/regression," *IEEE Access*, vol. 6, pp. 66715–66722, 2018.
[11] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, Sep. 2010, pp. 1045–1048.
[12] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.
[13] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2015, pp. 1556–1566.
[14] Q. Qian, B. Tian, M. Huang, Y. Liu, X. Zhu, and X. Zhu, "Learning tag embeddings and tag-specific composition functions in recursive neural network," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2015, pp. 1365–1374.
[15] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2014, pp. 49–54.
[16] T. H. Nguyen and K. Shirai, "PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 2509–2514.
[17] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. COLING 26th Int. Conf. Comput. Linguistics*, Dec. 2016, pp. 3298–3307.
[18] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.
[19] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1462–1471.
[20] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
[21] D. Bahdanau, K. Cho, and Y. Bengio. (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473
[22] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1412–1421.
[23] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, "Reasoning about entailment with neural attention," *CoRR*, vol. abs/1509.06664, pp. 1–9, Sep. 2015. [Online]. Available: http://arxiv.org/abs/1509.06664
[24] K. Zhou, J. Zeng, Y. Liu, and F. Zou, "Deep sentiment hashing for text retrieval in social CIoT," *Future Gener. Comput. Syst.*, vol. 86, pp. 362–371, Sep. 2018.
[25] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2016, pp. 606–615.
[26] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Sep. 2017, pp. 4068–4074.
[27] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2017, pp. 452–461.
[28] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. 32nd Conf. Artif. Intell. (AAAI)*, Apr. 2018, pp. 5876–5883.
[29] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 946–956.
[30] Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An, "Enhancing recurrent neural networks with positional attention for question answering," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 993–996.
[31] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1753–1762.
[32] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 2124–2133.
[33] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2011, pp. 151–160.
[34] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC-Canada-2014: Detecting aspects and sentiment in customer reviews," in *Proc. 8th Int. Workshop Semantic Eval.*, Aug. 2014, pp. 437–442.
[35] J. Wagner *et al.*, "DCU: Aspect-based polarity classification for SemEval task 4," in *Proc. 8th Int. Workshop Semantic Eval.*, Aug. 2014, pp. 223–229.
[36] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016.
[37] M. S. Akhtar, D. K. Gupta, A. Ekbal, and P. Bhattacharyya, "Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis," *Knowl.-Based Syst.*, vol. 125, pp. 116–135, Jun. 2017.

[38] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 2440–2448.
[39] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2016, pp. 214–224.
[40] X. Ma, J. Zeng, L. Peng, G. Fortino, and Y. Zhang, "Modeling multi-aspects within one opinionated sentence simultaneously for aspect-level sentiment analysis," *Future Gener. Comput. Syst.*, vol. 93, pp. 304–311, Apr. 2018.
[41] M. Saeidi, G. Bouchard, M. Liakata, and S. Riedel, "Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods," in *Proc. COLING 26th Int. Conf. Comput. Linguistics*, Dec. 2016, pp. 1546–1556.
[42] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6000–6010.
[43] B. Liu, X. An, and J. X. Huang, "Using term location information to enhance probabilistic information retrieval," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 883–886.
[44] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1243–1252.
[45] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, Apr./Jun. 1990.
[46] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[48] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1532–1543.

Authors' photographs and biographies not available at the time of publication.