Discovering Opinion Changes in Online Reviews via Learning Fine-grained Sentiments

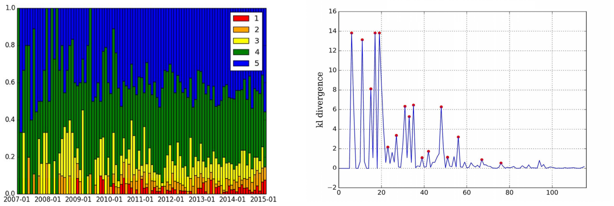
Zihang Liu*, Yu-Ru Lin†, Maoquan Wang‡ and Zhao Lu‡§
*Jinyuan Senior High School, Shanghai, China
†School of Information Science, University of Pittsburgh, Pittsburgh, USA
‡Dept. of Computer Science&Technology, East China Normal University, Shanghai, China
Email: †yurulin@pitt.edu, §zlu@cs.ecnu.edu.cn

***Abstract*—Recent years have shown rapid advancement in understanding consumers' behaviors and opinions through collecting and analyzing data from online social media platforms. While abundant research has been undertaken to detect users' opinions, few tools are available for understanding events where user opinions change drastically. In this paper, we propose a novel framework for discovering consumer *opinion changing events*. To detect subtle opinion changes over time, we first develop a novel fine-grained sentiment classification method by leveraging word embedding and convolutional neural networks. The method learns sentiment-enhanced word embedding, both for words and phrases, to capture their corresponding syntactic, semantic, and sentimental characteristics. We then propose an opinion shift detection algorithm that is based on the Kullback-Leibler divergence of temporal opinion category distributions, and conducted experiments on online reviews from Yelp. The results show that the proposed approach can effectively classify fine-grained sentiments of reviews and can discover key moments that correspond to consumer opinion shifts in response to events that relate to a product or service.**

***Keywords*-Fine-grained sentiment classification; Distribution representation; Opinion shift; Social media**

I. Introduction

In recent years, social media platforms, such as Twitter, have become important sources for users to report real-world events or to express their "opinions" on various aspects of their lives, such as an experience with a product or service. More and more studies have focused on extracting opinions or sentiments from such user-generated content. Existing sentiment classification approaches have been employed; most of them have focused on a classification into overall binary polarities (i.e., positive and negative) [1][2]. However, these methods lack the ability to distinguish complicated or fine-grained sentiments that users may express. Distinguishing fine-grained or multiple sentiment categories is a critical step for understanding how a real social event unfolds over time or how the public expresses feelings with respect to a significant event (such as a crisis event or a political debate). In the consumer domain, drastic changes in user sentiments may also reflect changes in user opinions about service quality, product prices, or the potential injection of fake reviews in the business sites.



(a) (b)

Figure 1: (a) Timed-stamped opinions generated by our approach for a restaurant from Yelp. (b) The Kullback-Leibler divergence of opinions about the restaurant. The red dots reveal the time points of opinion changes.

There are many existing change point detection algorithms [3][4]. MOA-TweetReader is a system to detect changes in binary categories [3]. This method uses ADWIN (ADaptive sliding WINdow) as a change detector to detect opinion changes from positive to negative, or vice versa. Other work focuses on detecting anomalies or detecting numerical changes [4]. Compared with these studies, we propose a new task, which is to detect an opinion change event – such as opinion changes that can range from "very positive" to "positive", "positive" to "neutral", "neutral" to "negative", "negative" to "very negative", and so forth. All these changes may be perceived qualitatively differently from one another, and could reflect various issues with the provided services or products.

To tackle this task, we propose a method for detecting customer opinion changes through analyzing social media users' reviews that reflect fine-grained sentiment changes over time. We aim to detect the opinion change of the users, as well as times in which opinions deviate significantly from those of other users at the previous moment.

Figure 1 shows a real world example of opinion changes discovered from Yelp data regarding a particular restaurant. The left plot shows the original time-stamped opinions extracted from the site, with colors indicating the different level of positive reviews from 1 to 5 – they are *very negative*, *negative*, *neutral*, *positive* and *very positive*, respectively. The height of each bar represents the volume of these opinions at the current time. The right plot shows three time periods of *opinion changes* detected by our method. These opinion changes occurred between January 2007 and January 2010. After January 2010, the opinion changes became less prominent, and eventually stabilized at the beginning of 2015. As we will show in Section 5, these opinion changes occurred due to problems with the food quality or the services offered by the restaurant.

Our approach aims to detect the time points at which users' opinions deviate from the previous ones. This can have broad applications in monitoring consumer sentiments about certain products or services, or general public opinions. A technical challenge involved in this monitoring process is how to identify more fine-grained differences in sentiments, instead of simple binary sentiment categories.

Recent research on sentiment classification approaches can be divided into two broad categories: lexicon-based and non-lexicon based approaches. The first approach achieves good performance, as they leverage the typical sentiment polarities of words in lexica [5][6]. However, these methods need manually labeled lexica or a background corpus to build lexica. The quality and domain difference of the lexicon have a great influence on the classification result.

The latest work shows the effectiveness of word embedding (namely word representation) in binary polarities sentiment classification tasks, as it can capture both syntactic and semantic relationships among words. To classify short texts (e.g., Twitter), two methods [1][2] are proposed: however, they are designed for binary classification. Binary sentiment classification approaches cannot be directly applied to fine-grained sentiment classification for two reasons: (1) Binary classification methods identify opinions of reviews into two polarities, while models for distinguishing multiple categories require more sophisticated learned decision boundaries. Classifying multiply polarities is more difficult than that of classifying binary polarities. (2) Features extracted by existing word embedding are not effective at identifying slight differences among fine-grained user opinions.

For multi-category sentiment classification, recursive neural networks (RNNs) and recursive neural tensor networks (RNTNs) and their variants [7][8][9], convolutional neural networks (CNNs) [10] are used to construct sentence vectors using pre-trained word embeddings. However, their methods face the issues of time complexity for higher dimensions (they don't scale well with more than 300 features).

In this work, we propose a framework of Detecting opinion Change from Temporal Reviews (DoCTeR). Our framework has two components – classifying opinions from reviews and detecting opinion shifts from the detected opinions. First, we propose a novel fine-grained sentiment classification approach that leverages Sentiment-Enhanced Word Embedding (SEWE) and convolutional neural networks. Second, we detect users' opinion changes by using a sliding window-based detection algorithm. Our experimental results on two real-world datasets show that the proposed framework not only effectively classifies opinions of reviews into fine-grained categories, but is also useful to detect changes in consumers' opinions. Three **main contributions** are:

1) We propose a novel sentiment-enhanced word embedding learning algorithm that captures not only syntactic and semantic, but also sentimental characteristics of online reviews. Learned word embedding is used to identify slight differences among user opinions.
2) We propose a new method of classifying opinions of reviews into fine-grained sentiment categories by using the proposed sentiment-enhanced word embedding and convolutional neural networks. Comparisons with several state-of-the-art approaches on real datasets show the effectiveness of our method.
3) We propose a sliding window-based algorithm of opinion shift detection that is based on the differences between opinion distribution measured by the Kullback-Leibler divergence. Our inspection of the reviews shows that we can use those multi-class sentiments to discover the ways in which users shift their opinions of products or services.

The remainder of the paper is organized as follows. Section 2 discusses the related work of multi-class sentiment classification. Section 3 details problem statement. Section 4 describes our framework. In section 5, we show our experimental results and discuss our discoveries in opinion shift analysis. The conclusion and future work are presented in Section 6.

II. Related Work

The dynamics of user opinions and opinion shifts about specific products or services are difficult to characterize. The task of multi-class sentiment classification classifies sentiments into multiple categories, which is helpful to model the features of user opinions and may provide further assistance in detecting opinion shift events.

Current work pays more attention to non-lexicon based approaches. Some machine learning algorithms [11][12] are introduced for sentiment analysis of texts. To classify sentiments of long texts into five sentiment categories, Ortigosa et al. [13] used native Bayes to train multi-dimensional vectors. Wang et al. [14] compared SVM and Native Bayes on sentiment classification of texts, and proposed the NBSVM (support vector machines with native Bayes) algorithm. Hermann et al. [15] proposed the CCAE (combinatorial category autoencoders) approach for sentiment analysis. Socher et al. [16] proposed the MV-RNN (matrix-vector recursive neural network) model, which learns distribution representations of sentences using a syntax tree. However, these methods are sensitive to data sparseness.

Since the success of word embedding in text mining, word embedding has been increasingly used in sentiment classification. Most of the existing related work are for binary polarity sentiment classification. Santos et al. [2] classified sentiments of short text using word embedding and deep convolutional neural networks. Tang et al. [1] improved the traditional word embedding algorithm and proposed the SSWE (sentiment-specific word embedding) algorithm for Twitter sentiment classification. For multi-class sentiment classification, recursive neural network, recursive neural tensor network, and its variant model, the convolutional neural network, were used to construct sentence vectors using pre-trained word embeddings [7][8][9].

In this work, we first examine the problem of multi-class sentiment classification. Similar to related work [8][17], we define five sentiment polarities. Compared with existing works, we first propose a sentiment-enhanced word embedding process for general word embedding by adding the sentiment of each text. We use CNN which show their effectiveness in the domain of text classification as the classifier [10][18]. Furthermore, to our best knowledge, the work presented in this study is the first attempt at detecting opinion shift about a product or service using fine-grained sentiment categories.

III. Problem Statement

Without a loss of generality, we consider a discrete m-level sentimental label system (i.e., sentiment categories from 1 to m), which is extensively used on today's online e-commerce sites, such as Yelp and Amazon, which adopt one-to-five-star sentimental labeling systems. We view k as the sentiment label of the k-th sentiment category.

We first denote a corpus as $T = [X, Y] \in \mathbb{R}^{n \times (u+m)}$, where $X \in \mathbb{R}^{n \times u}$ is a review matrix and $Y \in \mathbb{R}^{n \times m}$ is a sentiment category matrix. Three variables, n, u and m, refer to the number of reviews, the number of features contained in reviews, and the number of sentiment categories contained in the corpus, respectively. With no less generality, in this work, we limit $m = 5$; that is to say, the number of sentiment categories of all reviews is five. Five columns in vector y refer to the five sentiment categories, they are *very negative*, *negative*, *neutral*, *good* and *very good*, separately. For each review $T = \{t_1, t_2, ..., t_n\}$ in the corpus, $t_i = (x_i, y_i) \in \mathbb{R}^{u+m}$ consists of review vectors $x_i \in \mathbb{R}^u$ and sentiment class vectors $y_i \in \mathbb{R}^m$.

With the above notations, we formally define the problem of multi-class sentiment classification for reviews of products as follows: given a review corpus T, our aim is to automatically detect sentiment categories Y for unlabeled reviews X; namely a map of X to Y: $f : X \rightarrow Y$.

After we obtain sentiment classes for online reviews, we detect opinion changes using these detected sentiment classes.

IV. Our Approach

Two components of our framework are the opinion classifier, which employs the proposed SEWE model, and the proposed sliding window-based detector for opinion shift detection.

The first component consists of four main steps. First, the phrases-learning step identifies phrases as pseudo-words from a large-scale corpus and constructs a phrase-pseudo-word list. After this step, all sentences are formed by words and/or pseudo-words. Second, the sentiment-enhanced word embedding construction step learns sentiment-enhanced word embeddings for words and pseudo-words from a large-scale corpus at the same time, and produces a word-word embedding list. Third, the word embedding integrating step integrates word embeddings for words and pseudo-words in a review. We keep same lengths of all reviews through removing the rests of long reviews (if the length of a review is longer than η) and adding random word embeddings for short reviews (if the length of a review is shorter than η). We use the forth step to classify reviews into m categories (namely, sentiment labels) through employing a convolutional neural network.

The second component is a sliding window-based detector; we use it to detect opinion shifts from sentiment categories.

A. Opinion Classification from Reviews

1) Phrases learning: It is well known that the overall meaning of phrases are not simply a composition of the meanings of their individual words. For instance, *New York* is a city in the United States and not a prefecture in the United Kingdom, while *The New York Times* is a famous daily newspaper and not a position on a clock. Based on previous observations, if we can detect those phrases and learn their word embedding by treating them as pseudo-words, it may be possible to achieve a higher performance of classification.

Our next observation is: given a sentence, such as *I like New York*, if we directly use a word embedding model to capture the features of the sentence, four word vectors, *I* (w_1), *like* (w_2), *New* (w_3), and *York* (w_4), are obtained. In contrast, if we learn phrases first, we will obtain the word vector of *New York* (w'_3) since *New York* is a phrase that we view as a pseudo-word. Thus, the representation of the example sentence is converted to $\{w_1, w_2, w'_3\}$ from $\{w_1, w_2, w_3, w_4\}$. That is to say, using w'_3 to represent *New York* is more significant than that of $w_3 + w_4$.

Originated from [19], we judge whether two adjacent words are a phrase or a part of a phrase by using

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)}, \tag{1}$$

where $w_i w_j$ refers to two adjacent words, $count(x)$ is the number of occurrences of x in the dataset, and δ is a parameter used to prevent too many phrases that consist of infrequently found words. The scores of two words that are larger than a predefined threshold are chosen as phrases. We

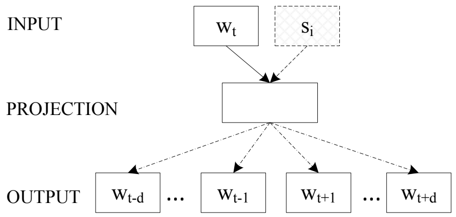


Figure 2: An example of the proposed SEWE model. S_i is the sentimental feature of a review.

then combine pseudo-words with normal words to construct word embeddings.

2) Sentiment-enhanced word embedding construction: Although syntactic and semantic similarities obtained by existing word embedding algorithms are quite useful for sentiment classification, syntactic and semantic similarities are not equal to sentimental similarities. As a result, we cannot use both syntactic and semantic similarities to classify sentiments of texts. For example, two words, *satisfied* and *disappointed*, convey distinct sentiments but are similar in syntactic structure. In this work, we improve the exiting word embedding model, the continuous skip-garm [19], by introducing sentimental features of reviews. We call the new model sentiment-enhanced word embedding (SEWE). We show an example of SEWE in Figure 2.

The traditional continuous skip-gram predicts surrounding words (d words before and after w_t) based on the current word w_t; that is, $P(w_{t-d:t+d}|w_t)$. Our observation shows that users write reviews that include their potential opinions about products or services. With the influences of the opinions, they will choose certain words that are suitable to unconsciously express their opinions. From this observation, we extend the continuous skip-garm model to SEWE model from two aspects: given a corpus T that consists of n reviews, and each review t_i with sentiment S_i contains a word sequence $(w_1, w_2, ..., w_t, ..., w_k)$, we first use a pseudo-word to represent the sentiment S_i of review t_i. Second, we learn both word representations for the surrounding words (d words before and after w_t) of word w_t, and the pseudo-word S_i. That is, our aim is to have $P(w_{t-d:t+d}, S_i|w_t)$. The objective of the SEWE model is to maximize the average log probability:

$$\frac{1}{V}\sum_{i=1}^{n}\sum_{t=1}^{k}\left(\sum_{-d\leq j\leq d}\log p\left(w_{t+j}|w_t\right)+\log p\left(S_i|w_t\right)\right) \tag{2}$$

here, $j \neq 0$, V is the total number of words in corpus T.

The basic Skip-gram model uses the probability function $p(w_O|w_I)$ as,

$$p\left(w_O|w_I\right)=\frac{\exp\left({v'}_{w_O}{}^{\top}v_{w_I}\right)}{\sum_{w=1}^{V}\exp\left({v'}_{w}{}^{\top}v_{w_I}\right)} \tag{3}$$

here v_w and v'_w mean the input and output vector representations of word w separately. Since the cost of computing $\bigtriangledown \log p\left(w_O|w_I\right)$ is proportional to V, this causes the above formulation to be impractical. A computationally efficient approximation of the full softmax is the hierarchical softmax [19]. The hierarchical softmax uses a binary tree representation of the output layer with the V words as its leaves, and for each node, explicitly represents the relative probabilities of its child nodes. Similar to previous work [19], we employ the definitions of hierarchical softmax, which uses a binary Huffman tree and negative sampling to train word embeddings.

3) Word embedding integration for reviews: We then construct review embeddings for reviews, based on the word embeddings of words contained in reviews. For instance, given the review, *I like New York very much and I will go there again*, we use the pseudo-word *New_York* to represent both *New* and *York*. After all word vectors in the review are found, according to the pre-trained word embeddings, they are sequentially linked. Considering that the length of each review is different, we limit the length to r (filled up or cut). Thus, a review is represented as $v_{1:r} \in \mathbb{R}^{rk}$, where $v_i \in \mathbb{R}^k$ is the i-th word in a review, $v_{i:j}$ is a concatenation of $v_i, v_{i+1}, ..., v_j$.

4) Fine-grained sentiment classification: Convolutional neural networks have been applied to NLP, such as semantic parsing and sentence classification [10]. The CNN model can take word sequences as input. To extract features that contain word sequences, sentimental characteristics, and syntactic and semantic similarities, we propose to combine CNN and SEWE. We extend the CNN model proposed in [10]. Figure 3 shows the steps of fine-grained sentiment classification using CNN.

We refer $\boldsymbol{g} \in \mathbb{R}^{h\times k}$ to the filter matrix, which is applied to a window of h words to produce the feature c_i as

$$c_i = f\left(\boldsymbol{g}\cdot v_{i:i+h-1}+b\right). \tag{4}$$

This filter is applied to each possible windows of words in the review $x_{1:h}, x_{2:h+1}, \cdots, x_{n-h+1:n}$ to produce a *feature map*. For example, given a window of words $v_{i:i+h-1}$, a feature c_i is generated by a convolution operation applied to a window of h words.

$$\boldsymbol{c} = [c_1, c_2, c_3, ..., c_{r-h+1}] \tag{5}$$

with $c \in \mathbb{R}^{r-h+1}$.

To capture the most important feature (the one with the highest value) for each feature map, a max-over-time pooling operation [20] is performed over the feature map, and we take the maximum value $\hat{c} = \max\{\boldsymbol{c}\}$ as the feature corresponding to this particular filter. The pooling scheme naturally deals with variable sentence lengths.

To reduce the computational complexity and capture the most important feature, we apply a max pooling operation over the feature map in Eq. (5).

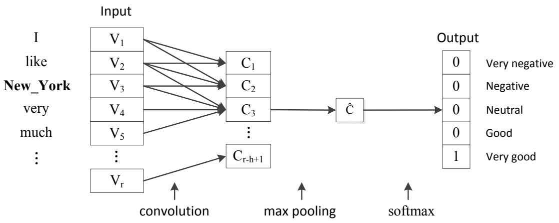


Figure 3: The procedure of the fine-grained sentiment classifier using CNN.

Finally, we use a softmax to obtain the classified results. For training data $\{(x_i, y_i), ..., (x_k, y_k)\}$, y is divided into m classes, namely $y_i \in \{1, 2, ..., m\}$. The hypothesis function of softmax h_θ is,

$$h_\theta = \begin{bmatrix} p(y_i = 1|x_i; \theta) \\ p(y_i = 2|x_i; \theta) \\ ... \\ p(y_i = m|x_i; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{m} e^{\theta_j^T x_i}} \begin{bmatrix} e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ ...e^{\theta_m^T x_i} \end{bmatrix} \tag{6}$$

where $\theta_i, \theta_2, ..., \theta_m$ are the parameters.

The cost function of softmax J is defined as,

$$J(\theta) = -\frac{1}{l}\left[\sum_{i-1}^{l}\sum_{j=1}^{m} 1\{y_i = j\} \log \frac{e^{\theta_j^T x_i}}{\sum_{i=1}^{m} e^{\theta_i^T x_i}}\right] v. \tag{7}$$

Here $1\{\cdot\}$ is the indicator function. So that 1{a true statement} = 1, and 1{a false statement} = 0.

B. Opinion Shift Detector

After we identify opinions from reviews, we detect user opinion shifts using the Kullback-Leibler divergence [21] and a sliding window-based detection algorithm.

1) Kullback-Leibler Divergence: For two discrete probability distributions P and Q, the Kullback-Leibler divergence of Q from P is defined as

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \tag{8}$$

It is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilies P. The Kullback-Leibler divergence is defined only if $Q(i) = 0$ implies $P(i) = 0$, for all i (absolute continuity). Whenever $P(i)$ is zero, the contribution of the i-th term is also interpreted as zero.

An example is shown in Figure 1(b). We use the KLD curve to represent the temporal opinions of a restaurant's comments. Note that the KLD curve is measured over time, $D = (d_1, d_2, \cdots, d_i, \cdots, d_Z)(1 \leq i \leq Z)$, where the height d_i is the KLD distance of the detected multi-class opinion distributions between the current month and the previous one. We can see that there are several prominent points, compared to the rest of the points in the curve. These prominent points indicate that user opinions would change.

To detect the points that are prominent in relative time ranges, we propose a sliding window-based detection algorithm. First, we obtain the maximum height d_i, when we scan the curve D using an initialized window size of ω_1. Then, the rest of d_i are detected, when we iteratively update the window size ω_k, until we meet the exit condition of the window size Z. Next, we compute the mean value h_a of all d_i. Here, the mean value h_a is the optimized parameter we take to scan the curve D again. Those heights d_j, which are larger than h_a, are defined as the prominent points.

V. Experiments and Analysis

In order to evaluate the proposed framework, we conducted two groups of experiments. The objective of the *first group experiments* was to evaluate the SEWE model. We designed the *second group experiments* to validate our opinion detector using the sliding window-based algorithm. All experiments were conducted on commodity hardware with 2.5 GHz CPUs and 8 GB of main memory.

The parameters in two groups of experiments are set as follows: There are two parameters in the phrase-learning step. The parameter η in the phrase learning step is set to 5 according to [19] in the aim of preventing more less common phrases. We set the threshold g to 1.17×10^{-4} according to our experiments. We set the length of review in the review vector to 72, according to the characteristics of two experimental dataset. Same to [10], we also set the window length of CNN to 3, 4, 5, and the probability of 1 in the bernoulli random variable parameter α. We set window size ω_1 to 2 in our experiments.

A. Experimental Dataset

We prepared three datasets to evaluate our approaches. Statistical information on these datasets is given in Table I.

Table I: Statistical information of three datasets

	YelpNY	MR [11]	SST [7]
#Reviews	25,000	5,331	11,855
#Sentiment categories	5	2	5

The first dataset is the *Yelp NewYork* (YelpNY) dataset. We collected 1,445,308 reviews for 15,781 venues in New York City using the Yelp Search API * from before November 19, 2014. Each review has a rating score from 1 to 5, which can be viewed as sentiment categories expressed by users. That is to say, these ratings refer to five sentiment categories, *very negative*, *negative*, *neutral*, *positive*, and *very positive*, respectively.

We then selected 50,000 textural reviews (10,000 for each sentiment class) as the *review corpus* to build sentiment word vectors. The YelpNY dataset is constructed by choosing 22,500 reviews of almost 6,000 venues for training (4,500

*http://www.yelp.com/developers/documentation/v2/searchapi for each category) and 2,500 reviews for testing (500 for each sentiment class). In this paper, we only report on the experiment results of a training set of a fixed size. However, based on our sensitivity test on accuracy vs. training size, we found that the accuracy would not be further improved by using a larger training set. The length of review is always short and many reviews are full of non-alphabetic characters, websites, emoticons, and numbers. To get better results, our preprocessing included removing punctuation, websites, numbers, emoticons, and converting uppercase letters to lowercase.

The second one, *Movie Review* (MR), is introduced in [11]. We used the sentence polarity dataset v1.0 of the MV dataset. Each review of MV has one sentence, and each sentence is classified into two sentiment categories, either negative or positive. Each category consists 5,331 sentences. Since there are not training dataset and testing dataset in the MR dataset, we measure its classification accuracies by 10-fold cross-validation.

The third dataset, *Stanford Sentiment Treebank* (SST), is introduced in [7]. The Sentiment Treebank includes fine-grained sentiment labels for 11,855 sentences. The training set contains 8,544 reviews that are labeled with 5 sentiment categories. The test set has 2,210 reviews.

B. Experiments on Opinion Detection

We first evaluate the steps of the SEWE model, and we evaluate opinion classification employing the SEWE model on the YelpNY dataset to compare it with other state-of-the-art methods.

1) Evaluate the SEWE model: To verify that the word vectors trained by SEWE have higher sentimental similarities, we compare similar words obtained by SEWE and word2vec for pairs of words. For two pairs of words, {satisfied, disappointed} and {good, bad}, we count the top 5 and top 10 similar words contained in word vectors generated by SEWE and word2vec. Their *error rate* is shown in Figure 4. We view the error rates of Word2vec as the baseline. As Figure 4 shows, the error rate for two pairs of words obtained with SEWR is lower than that obtained with Word2vec. We conclude that SEWE obtains more sentiment information that that of Word2vect during its training process.

2) Evaluate opinion detection employing SEWE: We compared four models, Ini_CNN, Phrases_CNN, Pnrases_Word2vector_CNN and Phrases_SEWE_CNN, on the Yelp dataset. The description and experimental results of each model are depicted in Table II.

We viewed the Ini_CNN model as the baseline. Two other models, Ini_CNN and Phrases_CNN, are used to show the effectiveness of the proposed phrases learning step. Moreover, we compare two models, Phrases_SEWE_CNN and Phrases_Word2vec_CNN, on the Yelp dataset. The first model uses SEWE and the second one uses Word2Vec.

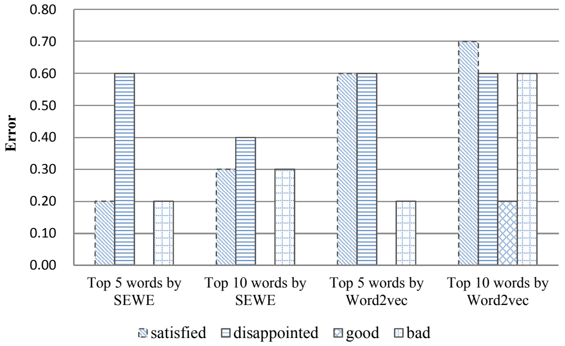


Figure 4: The error rates of two pairs of words, {satisfied, disapointed} and {good,bad}, learned by SEWE and Word2vec, respectively. The left two bars are the error rates obtained by SEWE; the right two bars are the error rates of Word2vec.

Table II: Model Descriptions and Experimental Results

Model	Model description	Accuracy

Ini_CNN	Baseline;
	Without phrases learning;
	Initialize word embedding randomly;
	Classified by CNN
		48.1%

Phrases_CNN	Adding phrases learning; Initialize word vectors randomly; Classified by CNN	51.5%

Phrases _Word2vec_CNN	Adding phrases learning;
	Word embedding by Word2vec;
	Classified by CNN.
		52.9%

Phrases_SEWE _CNN	Adding phrases learning;
	Word embedding by SEWE;
	Classified by CNN.
		53.8%

The experimental results of two models are shown in Table II. The accuracies of Phrase_CNN, Phrases_SEWE_CNN, and Phrases_Word2vec_CNN are higher than that of Ini_CNN. This performance verifies that phrase learning and word embedding can improve the classification accuracy. We notice that the accuracy of Phrases_SEWE_CNN is higher than that of Phrases_Word2vec_CNN. The reason is that word embedding learned by SEWE captures more sentimental information, and it is more suitable for sentiment classification. However, the improvement is lower than our expectations indicated. After analyzing the reviews in the dataset, we find the reason: the degrees of sentiments expressed in reviews is subtle, e.g., {poor, fair}, {very good, outstanding}. The subtle differences cause the learning of similar word vectors.

The improvement shown in Table II seems to be small; however, what is more useful is that the error rates are different across different opinion pairs. As shown in Figure 4, there are considerable differences, especially in the "satisfied" and "bad" categories. We believe that these subtle differences are particularly important in fine-grained sentiment analysis.

3) Comparison with competitive methods: To verify the effectiveness of our approach for the tasks of binary polarity sentiment classification and multi-category sentiment classification, the outcome of seven widely adopted models are compared on the MR dataset and the SST dataset, including two kinds of techniques, one based on word embedding and one without word embedding. Some brief introductions of these methods are given:

MV-RNN: Matrix-vector recursive neural network with parse trees [16].

RNTN: Recursive neural tensor network with tensor-based feature function and parse trees [7].

CNN-multichannel: Convolutional neural networks that are trained on top of two set of pre-trained word vectors [10].

NBSVM: Support vector machines with native Bayes with uni-bigrams [14].

CCAE: Combinatorial category autoencoders with combinatorial category grammar operators [15].

RAE: Recursive autoencoders with pre-trained word vectors from Wikipedia [22].

Paragraph-Vec: Logistic regression on top of paragraph vectors [23].

Comparison experiments on the MR dataset: We compare our approach to other four popular sentiment classification methods for the task of binary polarity sentiment classification in Table III. Experimental analysis shows that the SEWE approach provides better accuracy on the MR dataset.

Table III: Comparison with other approaches on MR

Model	Accuracy
MV-RNN [16]	79.0%
NBSVM [14]	79.4%
CCAE [15]	77.8%
RAE [22]	77.7%
Our approach	**79.8**%

Comparision experiments on the SST dataset: We compare our approach with other four state-of-the-art sentiment classification methods for the task of fine-grained sentiment classification on the SST dataset in Table IV.

Table IV: Comparison with other approaches on SST

Model	Accuracy
MV-RNN [16]	44.4%
RNTN [7]	45.7%
CNN-multichannel[10]	47.4%
Paragraph-Vec [23]	48.7%
Our approach	**49.1**%

We pre-train word vectors on the experimental corpus from the YelpNY dataset, which contains 50,000 reviews (33.3M size and 115,708 words). CNN-multichannel uses the publicly available word2vec vectors that were trained on 100 billion words from Google News. The vectors have 300 dimensions. Paragraph-Vec learned the word vectors and paragraph vectors using 75,000 training documents (25,000 labeled and 50,000 unlabeled instances). The learned vectors have 400 dimensions. Table IV shows that our approach outperforms CNN-multichannel and all other baseline methods. It is remarkable that the accuracy of our approach is based on such a small training experimental corpus.

Fine-grained sentiment classification for textural reviews remains a challenging task. Despite the small overall improvement (approximate 2%) over the second baseline, our method outperforms all current state-of-the-art methods with statistical significance assessed by McNemar's test ($p < 0.05$).

C. Experiments on Opinion Shift Detection

In the following, we demonstrate the effectiveness of our DoCTeR in discovering users' opinions shifts from reviews of products and service.

1) Data preparation: We selected the top 48 out of 21,892 restaurants contained in the YelpNY dataset for their plethora of textual reviews (with more than 1,000 reviews) posted by users. We constructed the *resYelpNY* dataset for manual annotation and opinion change detection. The reviews of these restaurants were posted during January 2005 to January 2015. The months of these restaurants registered at Yelp.com are different, from 26 months to 121 months. We statistic the resYelpNY dataset in Table V.

Table V: Statistic of the resYelpNY dataset

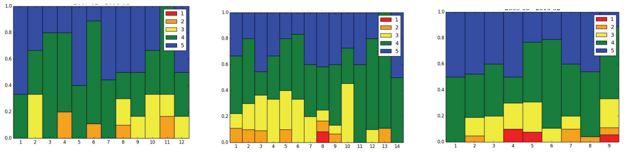
	Number
All reviews	79,670
All sentences	938,108
Average length of sentence	11
Length of maximum sentence	337
Length of minimum sentence	1

We submit all reviews of the 48 restaurants to the crowdsourcing platform, http://icrowd.ica.stc.sh.cn/crowd/, to manually annotate all opinion changes of these restaurants. The task of the crowdsourcing workers is to judge the opinion changes of each restaurant. Considering there are five opinion categories on Yelp.com, we preset four choices for the workers: *great*, *major*, *minor* and *no*, respectively. The four choices are defined in Table VI.

We set 40 sets of tasks for the 48 restaurants. Each set contains more than 90 tasks with the following items: times of reviews per month, reviews, and four choices. Each task will be done by three workers. After all sets of tasks are finished, we judge the answers according to the following rules: for a task, the maximum choice is viewed as the correct answer; if the choices are distributed, then we will

Table VI: Four kinds of opinion changes for manually annotation.

	very positive	positive	neutral	negative	very negative
very positive	no	minor	major	major	great
positive	minor	no	minor	major	major
neutral	major	minor	no	minor	major
negative	major	major	minor	no	minor
very negative	great	major	major	minor	no



(a) (b) (c)

Figure 7: (a), (b) and (c) are users' fine-grain opinion distribution between 2007-02 to 2008-01, between 2008-05 to 2009-06, and between 2009-06 to 2010-02, respectively.

check these tasks ourselves. We show the statistics of the labeling results of the resYelpNY dataset in Table VII.

Table VII: Statistics for the labeling results in the resYelpNY dataset.

	#all changes	#average changes
great change	144	3
major change	576	12
minor change	1,296	27

2) Experimental analysis of opinion detection: To evaluate the effectiveness of our DoCTeR approach, we first compare our approach to the task of binary opinion change classifier with the other two completive methods; then we show the effectiveness of the DoCTeR approach on the task of fine-grained opinion change detection. We report both accuracy and average accuracies for two kinds of opinion change detection.

Binary opinion change detection: Considering that the SEWE model is a important part of the DoCTeR approach, we compared the SEWE model with other two popular binary sentiment classifiers, MV-RNN [16] and NBSVM [14], for the task of binary opinion detection. After using the three sentiment classifiers, we employed the proposed KLD algorithm to detect the changes in opinion. Two binary polarities of opinions are "negative" and "positive." We view *very negative* and *negative* as a "negative" polarity, while *positive* and *very positive* as a "positive" polarity. The accuracies of the three sentiment classifiers are shown in Figure 5.

The average accuracies of the three opinion classifiers, SEWE, MV-RNN, and NBSVM are 82.7%, 80.6%, and 80.2%, respectively. The experimental results show that our SEWE approach with the proposed KLD algorithm provides the highest average accuracy.

Fine-grained opinion change detection: For the reviews of the 48 restaurants contained in the resYelpNY dataset, we employed DoCTeR to detect the time points at which reviewer opinions changed. In this experiment, we detected three kinds of opinion change: they are great, major, and minor change. The experimental results of detecting three kinds of opinion changes are shown in Figure 6.

The average accuracies of detecting great, major, and minor changes are 84.2%, 80.9%, and 77.6%, respectively. Figure 6 shows that DoCTeR is easily able to detect the majority of user opinion changes.

3) Our discoveries: We then demonstrated our discoveries by checking user reviews with respect to products or services. A restaurant located in Las Vegas was selected as the example. It has 4,137 reviews in the YelpNY dataset. We show its KLD curve in Figure 1(b). We observe that there are four time frames in which the KLD values have small fluctuations. The detailed opinion distributions of the first three frames is shown in Figure 7.

2007-02 to 2008-01 (Figure 7(a)): there is a lesser proportion of review opinions labeled 5, while opinion classes 3 and 4 scale up. Such a distribution indicates that there is an opinion shift. The trend of opinions expressed by users is from *very good* or *good* to *neutral*. Through reading the textual reviews, we find an opinion shift towards complaints about the restaurant during the given time period. Neutral and negative reviews began to appear. At first, customers expressed sentiments like "*Great food and great service......*", and "*Good place to watch sports. My friends stated burger were good*". Then, complaints appeared, such as "*...... the prices have went up*", and "*The kitchen was very slow*" .

2008-05 to 2009-06 (Figure 7(b)): there are more review opinions labeled 3 and 2, and some review opinions were labeled 1. The distribution means that the trend of users' opinion about the restaurant has moved from *neutral* to *negative*, and even to *very negative*. We notice that negative opinions suddenly increased with several very negative reviews during the time frame. Those reviews include "*......service was rude*", and "*......items were cold and uninspiring*". These reviews indicate that the level of service of the restaurant has dropped in the given months.

2009-06 to 2010-02 (Figure 7(c)): there were more and more review opinions labeled 1, 2, and 3. The trend suggests that during this period, the users felt that the service of the restaurant was not very satisfactory. Through checking relevant textural reviews, we found that both very negative and negative reviews grew in numbers. Most of reviewers criticized "*......the service is HORRIBLE*", and "*The rest of the burger was terrible*". These reviews indicate that the service of the restaurant was unsatisfactory during the given time period.

Interestingly, we have observed there are inconsistencies

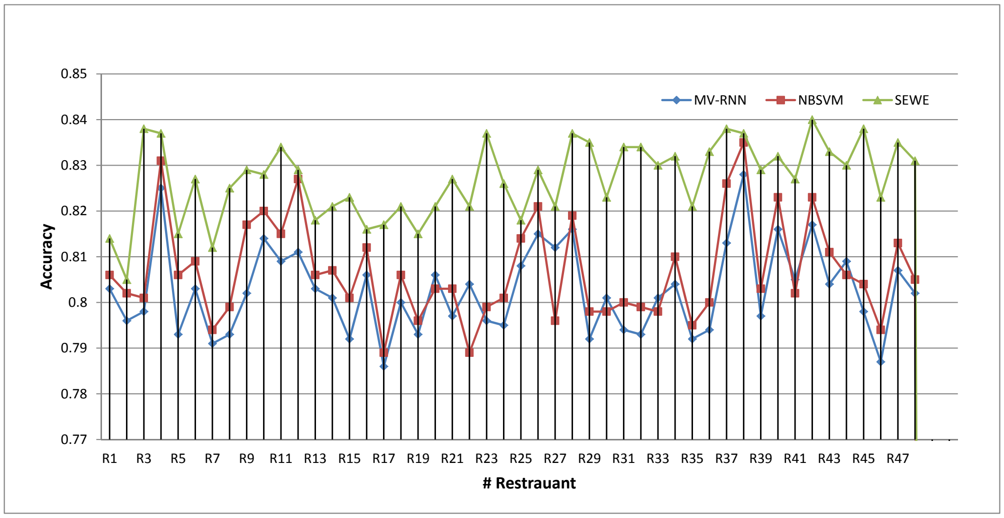


Figure 5: Experimental results on the resYelpNY dataset for binary opinion change detection.

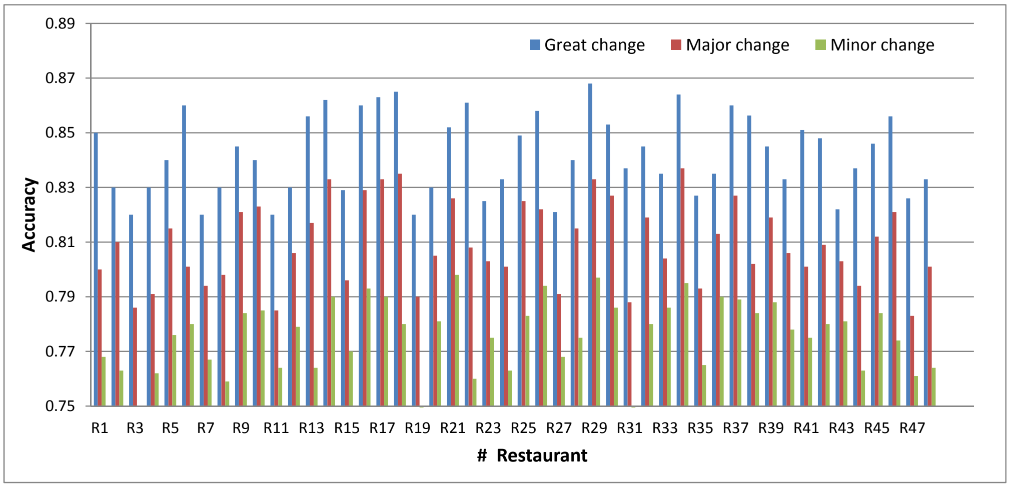


Figure 6: Experimental results on the resYelpNY dataset for fine-grained opinion change detection.

between the review content and their corresponding rating scores from same users. For example, *"Liz was awesome! Totally great service and attentive..."* is associated with a rating score '3' given by the same user.

VI. Conclusions and Future work

In this work, we proposed a framework of opinion shift detection from online reviews. Two components of the framework are the fine-grained opinion classifier and the opinion shift detector. The first component detects multi-class opinions using the SEWE model and convolutional neural networks. Our experimental results show the effectiveness of our approach. The second component of our framework is a sliding window-based detecting algorithm that identifies the time intervals in which opinions shift. Our experiments and discoveries of reviews show that the proposed approaches are helpful to identify opinion shifts about products or services.

In future, we will further improve the accuracy of opinion classification from online reviews using our multi-class sentiment classification algorithm. Moreover, to better understand the strengths and limitations of our method, we will conduct more quantitative evaluations of the proposed opinion shift detection approach.

Acknowledgments

This research was supported in part by National Key Technology Support Program (No.2015BAH01F02), in part by Science and Technology Commission of Shanghai Municipality (No.16511102702). The authors gratefully acknowledge the support of Yu-Ru Lin's Lab at the University of Pittsburgh, supported in part by NSF grant #1423697, #1634944, and the CRDF and CIS. Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of the funding sources.

References

[1] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2014, pp. 1555–1565.

[2] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland*, 2014.

[3] A. C. Bifet Figuerol, G. Holmes, B. Pfahringer, and R. Gavaldà Mestre, "Detecting sentiment change in twitter streaming data," in *Journal of Machine Learning Research: Workshop and Conference Proceedings Series*, 2011, pp. 5–11.

[4] S. Günnemann, N. Günnemann, and C. Faloutsos, "Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 841–850.

[5] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1275–1284.

[6] Y. Li, X. Li, F. Li, and X. Zhang, "A lexicon-based multi-class semantic orientation analysis for microblogs," in *Web Technologies and Applications*. Springer, 2014, pp. 81–92.

[7] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, p. 1642.

[8] H. Pouransari and S. Ghili, "Deep learning for sentiment analysis of movie reviews," 2014.

[9] R. Xu, T. Chen, Y. Xia, Q. Lu, B. Liu, and X. Wang, "Word embedding composition for data imbalances in sentiment and emotion classification," *Cognitive Computation*, vol. 7, no. 2, pp. 226–240, 2015.

[10] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[11] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[12] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 115–124.

[13] J. Ortigosa-Hernández, J. D. Rodríguez, L. Alzate, M. Lucania, I. Inza, and J. A. Lozano, "Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers," *Neurocomputing*, vol. 92, pp. 98–115, 2012.

[14] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012, pp. 90–94.

[15] K. M. Hermann and P. Blunsom, "The role of syntax in vector space models of compositional semantics." in *ACL (1)*, 2013, pp. 894–904.

[16] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1201–1211.

[17] M. Lee and R. Grafe, "Multiclass sentiment analysis with restaurant reviews," *Final Projects from CS N*, vol. 224, 2010.

[18] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proc. ACL*, 2015.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[20] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[21] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7893–7897.

[22] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 151–161.

[23] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv preprint arXiv:1405.4053*, 2014.