Distantly Supervised Lifelong Learning for Large-Scale Social Media Sentiment Analysis

Rui Xia, Jie Jiang, and Huihui He

Abstract—Although sentiment analysis on traditional online texts has been studied in depth, sentiment analysis for social media texts is still a challenging research direction. In the social media that contains a huge amount of texts and a large range of topics, it would be very difficult to manually collect enough labeled data to train a sentiment classifier for different domains. Distant supervision that considers emoticons as natural sentiment labels in the microblog texts has been widely used in social media sentiment analysis. However, the previous distant supervision works were normally trained based on an isolate set of data, and they were not capable to deal with the scenario where the texts are continuously increasing and the topics are constantly changing. To address such challenges, in this work we propose a distantly supervised lifelong learning framework for large-scale social media sentiment analysis. The key characteristic of our approach is continuous sentiment learning in social media. It learns on past tasks sequentially, retains the knowledge obtained from past learning and uses the past knowledge to help future learning. The lifelong sentiment classifier is trained on two large-scale distantly supervised social media datasets respectively, and evaluated on nine benchmark datasets. The results prove that our lifelong sentiment learning approach is feasible and effective to tackle the challenges of continuously updated texts with dynamic topics in social media. We also prove that the belief "the more training data the better performance" does not hold in large-scale social media sentiment analysis. In contrast, by conducting continuous learning from past tasks, our approach beats the traditional way of using all training data in one task, in terms of both classification performance and computational efficiency.

Index Terms—Sentiment analysis, social media analysis, distant supervision, lifelong learning

1 Introduction

With the rise of social network services such as Twitter and Weibo in recent decades, sentiment analysis toward texts in social media has gained much attention in the field of natural language processing and data mining. However, in comparison with sentiment analysis of traditional online texts, new challenges have arisen in sentiment analysis for social media texts.

On one hand, the amount of text data in the social media is massive and continuously increasing. On the other hand, hot topics in social media are constantly changing. New topics appear continuously, along with the old ones' decay. As is known, sentiment analysis is closely related to the topics and domains. In the large-scale social media that contains a large range of topics, it would be very difficult to manually collect enough labeled data to train sentiment classifiers for different domains.

Distant supervision, which makes the use of emoticons to naturally label the sentiment of the microblog text, has been proposed to address such problems and achieved success in the previous studies of social media sentiment analysis [1]. Sentiment classifiers in these work were normally trained in a single task based on an isolate dataset. Statistical learning models learned on single tasks may exhibit good performance in the case of small data. However, in the era of big data, it is very difficult to construct a universal and effective statistical learning model via single-task learning. For instance, it was not capable to deal with the scenario where the amount of data is continuously increasing and the topics are constantly changing. Therefore, establishing an adaptive and continuous learning model has become a big issue for large-scale social media sentiment analysis.

Lifelong machine learning (lifelong learning for short), which considers systems that can learn many tasks from one or more domains over its lifetime, has drawn much attention in recent years in many fields. The goal is to sequentially retain the knowledge learned from past tasks and selectively transfer that knowledge so as to develop more accurate hypotheses when learning a new task. Big data in social media offers a golden opportunity for lifelong learning because its scale and diversity give us abundant information for discovering the commonsense knowledge automatically. It enables an intelligent learning system to perform continuous learning, accumulate knowledge learned previously, and become more and more knowledgeable in future learning [2].

In this work, we propose a distantly supervised lifelong learning framework, for large-scale social media sentiment analysis. Specifically, we treat the increasing content of texts in social media as a time-series data stream. We collect the text data over time and divide the data into different partitions. We consider each partition as a past dataset, conduct single-task machine learning on each past task and then

• *The authors are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China. E-mail: rxia@njust.edu.cn, njustjiangjie@163.com, hehuihui1994@gmail.com.*

Manuscript received 22 Jan. 2017; revised 26 July 2017; accepted 8 Aug. 2017. Date of publication 7 Nov. 2017; date of current version 5 Dec. 2017. (Corresponding author: Rui Xia.)
Recommended for acceptance by E. Cambria, A. Hussain, and A. Vinciarelli.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TAFFC.2017.2771234

store the domain knowledge (such as vocabulary, class-conditional distribution, sentiment lexicon, a family of hypotheses of the classification algorithms, etc.) in that task. As a new task comes, we first compute the similarity between the new task and each past task. If the two tasks are similar, we merge the knowledge of two tasks into one; otherwise, we consider it as a new past task and store the knowledge learned in it additionally. Finally, we aggregate the knowledge from all past tasks, for prediction on the future tasks in an ensemble learning manner. We propose two ways for integrating the knowledge learned from different tasks in different conditions, leading to a Lifelong Bagging model and a Lifelong Stacking model, respectively.

The key characteristic of our approach is continuous sentiment learning in social media. It retains the knowledge gained from past learning and uses the knowledge to help future learning. The whole learning process is maintained in a continuous learning manner.

We train the lifelong learning model on two large-scale distantly supervised social media text datasets (an English Twitter dataset and a Chinese Weibo dataset). Both datasets contain millions of microblog messages. We conduct the evaluation by testing our model on nine benchmark sentiment datasets (five English datasets and three Chinese datasets). The results prove our approaches' applicability and efficiency in sentiment analysis for large-scale social media texts.

The main contributions of this paper are:

- To the best of our knowledge, it is the first work toward lifelong learning for large-scale social media sentiment analysis. Our method offers a continuous sentiment learning manner to retain the knowledge gained from the past and uses them to learn in future.
- Our method totally depends on the distant supervision information in the social media texts. It could be therefore cheaply applied to a new domain without the need of any manually labeled data.
- Our method is compatible to any available single-task learning algorithm (e.g., naïve Bayes, logistic regression and support vector machines) in each past learning task.
- We empirically prove that the conclusion, "the more training data the better performance," does not hold in large-scale social media sentiment analysis. In contrast, by learning continuously on past tasks and combining the knowledge learned from them, our lifelong ensemble framework beats the traditional way of using all training data in one task, in both classification performance and computational efficiency.

The following of the paper is organized as follows. The related work of social media sentiment analysis, distant supervision and lifelong machine learning are reviewed in Section 2. Section 3 presents the details of two large-scale distantly supervised social media datasets. In Sections 4 and 5, we introduce the details of the lifelong learning process, and present our Lifelong Bagging model and Lifelong Stacking model, respectively. The experimental results are presented and analyzed in Section 6. Section 7 finally draw the conclusions.

2 Related Work

2.1 Social Media Sentiment Analysis

Although sentiment analysis on traditional online texts has been studied in depth [3], [4], [5], [6], the characters of social media texts such as short length, large scale and dynamic topics have brought new challenges to the research of sentiment analysis. In the early beginning, many researchers tried to represent social media text in terms of feature engineering. For example, the use of linguistic information such as morphology [7], [8], [9], syntax [8], [9], semantics [10], [11], and lexicons [7], [9] was employed in text representation. Furthermore, some information specific in social media, such as emoticons and rich text information [7], [9], user relationship [12], social media text normalization [13], [14], and feature selection [15] was also incorporated for text representation.

In traditional sentiment analysis, supervised learning algorithms are the mainstream methods. However, they depend on an enough amount of manually labeled data to train the sentiment classifier. In social media, the amount of texts is huge and the topics vary frequently. It is hence difficult to manually collect enough training data for different domains. Some researchers therefore tried to avoid the labor-intensive labeling, and proposed distant supervision by making the use of natural labeling information in social media, for example, emoticons. Go et al. [1] first proposed sentiment classification with distant supervision. They considered Twitter messages containing emoticons, which can be converted to pseudo labels, as training data. For instance, Twitter messages which have ":)" were supposed to be positive, while that containing ":(" were considered as negative. The classification performance on Twitter data was more than 80 percent, with naïve Bayes, maximum entropy and support vector machine (SVM) as classification algorithms. It is obvious that the use of distant supervision can easily construct a large-scale training dataset. Similarly, Pak et al. [16] also used emoticons to filter the natural annotation corpus consisted of Twitter messages, and compared the performance of classification systems with different treatment of negative sentence and feature selection methods. Kouloumpis et al. [7] took hashtags and emoticons as natural annotation information and compared the classification performance based on different feature sets. Zhao et al. [17] built a microblog emotion classification system based on emoticon annotated data. The model can be updated iteratively through the incremental Naïve Bayes under the continuous data stream. Liu et al. [18] presented a model called emoticon smoothed language model (ESLAM) to integrate manually labeled data and noisy labeled data into one framework. Experiments demonstrated that ESLAM effectively integrates both kinds of data and outperforms model that uses any single one of them. Xiang et al. [19] build the sentiment classifier by introducing rich microblog features in SVM, and then proposed a semi-supervised learning framework combining LDA and sentiment classification.

There were also some efforts attempting to learn a domain-specific sentiment lexicon based on the natural annotation data in social media, and use such lexicons to help sentiment classification. For example, Muhanmud et al. [9] took the advantage of point-wise mutual information (PMI) to learn a sentiment lexicon, and then constructed

features for fully supervised learning with the help of the lexicon. Relied on this method, they won the first place in SEMEVAL-2013 sentiment classification task.

In recent years, some researchers used deep neural network to either learn a deep text representation or directly solve sentiment classification based on a large amount of social media dataset [20], [21], [22], [23]. Tang et al. [20] learned the sentiment-specific word embedding (SSWE) from massive distantly supervised tweets by developing three neural networks. The performance on the test dataset of SEMEVAL-2013 has been further improved after concatenating SSWE with existing feature sets.

Although the work of feature engineering, lexicon construction and distant supervision can to some extent alleviate the problems in social media sentiment analysis, most of them still follow the traditional single-task sentiment classification framework. They are still impracticable or inefficient when dealing with real-world social media sentiment analysis where the text data are continuously increasing and the topics are constantly changing.

In contrast, the work proposed in this paper is a lifelong sentiment learning framework. The manner of continuously sentiment learning matches well with the need of large-scale social media sentiment analysis.

2.2 Lifelong Machine Learning

Most of the current statistical learning algorithms, such as logistic regression, naïve Bayes, SVM, maximum entropy as well as the deep neural networks, are normally learned based on an isolated dataset and a single task. That is, the learning process is conducted on an isolated dataset in one period. Obviously, such an isolated learning pattern have many limitations. First, the knowledge learned in past tasks cannot be applied to future tasks. Second, due to the lack of prior knowledge, a large number of training examples was normally required to train a more generalized and adaptive model.

With the attempt to make machine learning algorithms act more like human beings, for example, retaining knowledge acquired in past tasks and using the past knowledge to help learning in future tasks, researchers have proposed the concept of "lifelong machine learning" ("lifelong learning") in many research fields, including machine learning, data mining and natural language processing. [24], [25], [26].

Although lifelong machine learning is a relatively novel concept, many similar concepts has actually been proposed before, such as lifelong learning [24], transfer learning [27], [28], multi-task learning [29], never-ending learning [30], self-taught learning [31], and online learning [32].

Multi-task learning aims at jointly learning to optimize multiple tasks. The learner optimizes the performance across all of the n tasks through some shared knowledge.

Transfer learning uses the source-domain labeled data to help target-domain learning. The goal of transfer learning is to learn well only for the target task.

In comparison, lifelong machine learning is a more generalized concept laying the emphases on learning continuously, accumulating the previously learned knowledge and using such knowledge to learn more ones. Lifelong learning differs from multi-task learning and transfer learning in two aspects: First, it is different from multi-task learning because lifelong learning does not jointly optimize the learning of all tasks, which multi-task learning does. It is also different from transfer learning because transfer learning identifies prior knowledge using the labeled and unlabeled data in the target domain. Second, the learning process of both multi-task learning and transfer learning are not continuous and have no knowledge retention process. But the learning process of lifelong learning is continuous and the learning on future tasks is based on the knowledge gained on past tasks.

Chen and Liu [2] gives a detailed introduction of lifelong learning and surveys the related work in different fields. According to their definition, a lifelong machine learning algorithm should have the following four parts:

(1) Past Information Store (PIS): It stores the information resulted from the past learning. This may involve substores for information such as i) the original data used in each past task, ii) intermediate results from the learning of each past task, and iii) the final model or patterns learned from the past task, respectively.
(2) Knowledge Base (KB): It stores the knowledge mined or consolidated from PIS. This requires a knowledge representation scheme suitable for the application.
(3) Knowledge Miner (KM): It mines knowledge from PIS. This mining can be regarded as a meta-learning process because it learns knowledge from information resulted from learning of the past tasks. The knowledge is stored to KB.
(4) Knowledge-Based Learner (KBL): Given the knowledge in KB, this learner is able to leverage the knowledge and/or some information in PIS for the new task.

Carlson et al. [30] and Mitchell et al. [33] introduced a never-ending machine learning system, NELL (Never-Ending Language Learning), to extract structured information from unstructured web pages. NELL has been in continuous operation since January 2010. Till now, NELL has accumulated a knowledge base of more than 3,027,643 asserted instances of 1,182 different categories and relations.

Chen et al. [34] proposed the first lifelong learning approach to sentiment classification, based on stochastic gradient descent optimization in the framework of Bayesian probabilities. Penalty terms were introduced to effectively exploit the knowledge gained from past learning. The aim of their approach was to learn the knowledge in multiple domains of the product review and help predict the sentiment of a new target domain. Their approach is designed for multi-domain sentiment classification and the knowledge of multiple past domains are learned at one time. In contrast, our approach is designed for large-scale social media sentiment analysis. Our method offers a continuous sentiment learning manner and is compatible to any available single-task learning algorithm.

3 Distantly Supervised Training Datasets

The proposed lifelong sentiment learning work is based on a large-scale sentiment dataset and distant supervision. In this section, we first introduce the details of dataset construction and distantly supervised sentiment labeling.

3.1 Dataset Construction and Sentiment Labeling

We use two large-scale social media datasets, respectively in English and Chinese, to train our lifelong learning model.

TABLE 1
The Statistics of Two Social Media Datasets

	English Twitter Dataset	Chinese Weibo Dataset
Num of Positive Examples	4,500,000	500,000
Num of Negative Examples	4,500,000	500,000
Vocabulary Size	1616493	425821
Average Length of messages	15.2	20.5

For English, we directly make the use of the Twitter corpus collected by [35]. They employed the Twitter Developer API to crawl tweets containing emoticons between February 2014 and September 2014. For Chinese, we construct a dataset from Weibo, which is a famous social media in China. We used the Weibo API to collect the Weibo microblogs from May 1st 2015 to December 1st 2015. Both datasets contain microblogs published in about half a year.

The distantly supervised sentiment labeling procedure in the English Twitter dataset is conducted in the following three main steps:

(1) First, we label tweets which contain ":)", ":)", ":-)", ": D", "=)" as positive examples and tweets which contain ":(", ": (", ":-(" as negative ones, in the same way as [22];
(2) Second, we remove all emoticons that are used to collect the training data following the method in [7], and ignore tweets whose tokens are less than 7 according to [20];
(3) Third, we utilize a Twitter tokenizer [36] to preprocess all tweets. Rare words that occur less than 5 times in the vocabulary are removed. HTTP links and username are replaced by "<http>" and "<user>", respectively.

For the Chinese Weibo dataset, the pre-processing steps are:

(1) We first perform Chinese word segmentation by using the ICTCLAS toolkit;
(2) We then label the Chinese microblogs which contain "[哈哈]","[害羞]","[爱你]" as positive samples and tweets which contain "[鄙视]","[吃惊]","[吐]" as negative samples;
(3) With the purpose of reducing the noise, we discard microblogs that contain more than 5 different emoticons. For the same consideration, we only keep the microblogs containing only positive emoticons for the positive category and the microblogs containing only negative emoticons for negative category. After the removal of emoticons and hashtags, the microblogs whose length is less than 5 are also abandoned.

Table 1 summarizes the statistics of two datasets.

3.2 Dataset Partition

Let D denote the large-scale social media text dataset. We first segment D into several partitions of texts ($D^{(i)} \subset D$), where i denotes the index of time interval. For example, $D^{(1)}$ denotes the partition of texts that were published in time interval t_1. Subsequently, $D^{(2)}$ contains the texts published in time interval t_2. In this way, we get a collection of dataset partitions

$$D = \{D^{(1)}, \ldots, D^{(t)}, \ldots, D^{(N)}\}, \tag{1}$$

with the time interval from t_1 to t_N.

We conduct single-task sentiment learning on each of the partition sequentially. Suppose we have learned on each partition in $\{D^{(1)}, \ldots, D^{(t-1)}\}$ and stored the knowledge in each task respectively. Our goal in lifelong learning is to use the knowledge learned from $\{D^{(1)}, \ldots, D^{(t-1)}\}$ to help learning on $D^{(t)}$. This learning mechanism operates continuously until reaching the end of the whole dataset.

The time stamp information was not provided in the original dataset [35]. Therefore, in our experiments, we suppose the number of tweets in each time interval (e.g., one day) is the same. Under this assumption, the dataset partition according to tweet number is the same as that according to time stamp.

The size of each dataset partition is user-defined. In the Section of experimental studies, we will discuss the size of dataset partition and its influence to the lifelong sentiment learning performance.

It should be noted that in real-world lifelong learning, the partitions are not finite. As time goes by, the number of past tasks is growing, and we need to design a manner to update the ever-growing past knowledge and use them to help learning for the future tasks.

4 Distantly Supervised Lifelong Ensemble Learning

Given the dataset partitions, lifelong learning is defined as follows. A lifelong learner has performed learning on a sequence of partitions from $D^{(1)}$ to $D^{(t-1)}$, and stored the knowledge learned in each task. When faced with the partition $D^{(t)}$, it uses the knowledge gained in the past $(t-1)$ partitions to help learning for the tth task.

According to [2], in order to build such a lifelong learning system, we need to answer the following questions:

(1) What kinds of knowledge should be retained from the past learning?
(2) What forms of information will be used to help future learning?
(3) How does the system use the knowledge to help future learning?

In this work we present a distantly supervised life-long ensemble learning framework, based on distant supervision and ensemble learning. Fig. 1 presents the general structure of the framework. It contains three parts: 1) past-task learning and knowledge storing; 2) new-task learning and knowledge updating, and 3) future-task prediction. In the following sections, we will introduce them in detail, and answer the above mentioned three questions.

4.1 Past-Task Learning and Knowledge Storing

According to the dataset partition presented in Section 3, the collections $D^{(1)}, \ldots, D^{(t-1)}$ are considered as the datasets for past learning, at time t. Learning on these partitions in sequence is viewed as past learning.

On each of the past $(t-1)$ tasks,

$$D = \{x_1^{(k)}, x_2^{(k)}, \ldots, x_m^{(k)}\}, k = 1, \ldots, t-1, \tag{2}$$

we define the following four kinds of domain knowledge:

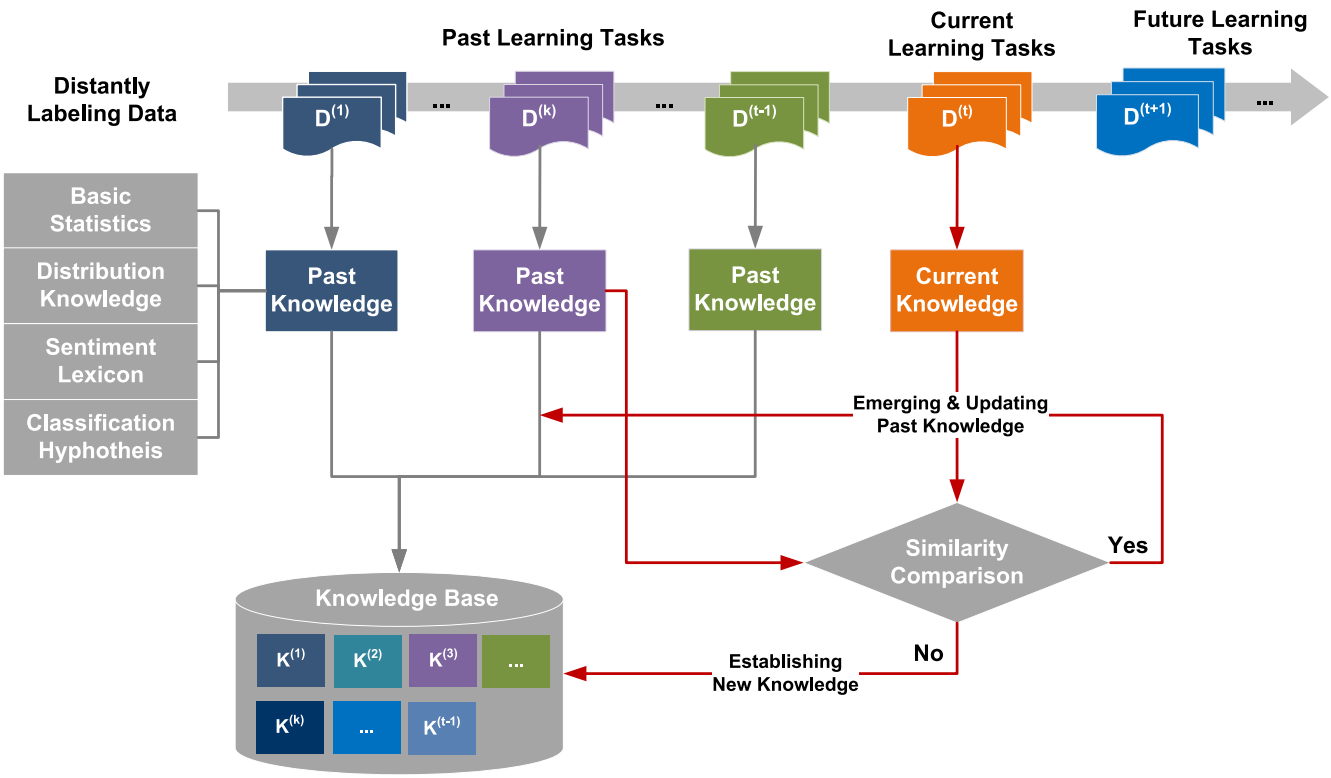


Fig. 1. The illustration of the lifelong learning framework for large-scale social media sentiment analysis.

(1) Basic Statistics. The vocabulary $V^{(k)} = \{w_1, \ldots, w_i, \ldots, w_v\}$, the number of training documents $N^{(k)}$, the document frequency in each class $N^{(k)}(c_j)$, the document frequency of each term in the vocabulary $N^{(k)}(w_i)$, and the document frequency of each term in each class $N^{(k)}(w_i, c_j)$ are considered as the basis statistic knowledge in past task $D^{(k)}$.

(2) Distribution Knowledge. According to the basic statistics, we can compute the following distribution knowledge: the distribution of each term $p^{(k)}(w_i) = \frac{N^{(k)}(w_i)}{N^{(k)}}$, the distribution of each class $p^{(k)}(c_j) = \frac{N^{(k)}(c_j)}{N^{(k)}}$, the joint distribution of term and class $p^{(k)}(w_i, c_j) = \frac{N^{(k)}(w_i, c_j)}{N^{(k)}}$, and the conditional distribution of each term given each class $p^{(k)}(w_i|c_j) = \frac{N^{(k)}(w_i, c_j)}{N^{(k)}(c_j)}$.

(3) Sentiment Lexicons. PMI is a widely used metric for construct domain sentiment lexicon. The PMI score of term w_i and class c_j is defined as

$$\begin{aligned} PMI^{(k)}(w_i, c_j) &= \log \frac{p^{(k)}(w_i, c_j)}{p^{(k)}(w_i)p^{(k)}(c_j)} \\ &= \log \frac{N^{(k)}N^{(k)}(w_i, c_j)}{N^{(k)}(w_i)N^{(k)}(c_j)}. \end{aligned} \tag{3}$$

On this basis, we can compute the semantic orientation (SO) for each term as follows

$$SO^{(k)}(w_i) = PMI^{(k)}(w_i, +) - PMI^{(k)}(w_i, -), \tag{4}$$

where $+$ and $-$ denote the class labels of positive and negative, respectively.

Using $SO^{(k)}(w_i)$ to denote the sentiment score of w_i, we obtain a sentiment lexicon $L^{(k)}$ in each past task. We could also observe the semantic change of words over time, by comparing $SO^{(k)}(w_i)$ across the past tasks over time.

We could further calculate the SO score of the whole message, by summing up the SO score of each word in it

$$SO^{(k)}(x) = \sum_{w_i \in x} SO^{(k)}(w_i), \tag{5}$$

and then determine the sentiment of the message as follows

$$s(x) = \begin{cases} + & \text{if } SO(x) > 0 \\ - & \text{if } SO(x) < 0 \\ random\{+, -\} & \text{if } SO(x) = 0. \end{cases} \tag{6}$$

(4) Statistical Learning Hypotheses. Based on each partition $D^{(k)} = \{(x_1^{(k)}, y_1^{(k)}), (x_2^{(k)}, y_2^{(k)}), \ldots, (x_m^{(k)}, y_m^{(k)})\}$ where m is the number of training examples, we train the single-task sentiment classification model $h^{(k)}$. Note that any type of sentiment classification algorithms (e.g., logistic regression, naïve bayes, SVM, etc.) can be used as the single-task classification algorithm. The model hypothesis $h^{(k)}$ as well as the parameters $\theta^{(k)}$ are stored as the supervised classification knowledge in the kth task.

In this work, we have tried two classic sentiment classification algorithms (i.e., logistic regression, naïve bayes). Due to the space limitation, here we only use logistic regression as an example for description.

Logistic regression is a classic linear classification model. It has the following hypothesis

$$p(y = 1|x; \theta^{(k)}) = h^{(k)}(x) = \frac{1}{1 + e^{-\theta^{(k)} \cdot x}}, \tag{7}$$

$$p(y = 0|x; \theta^{(k)}) = 1 - h^{(k)}(x), \tag{8}$$

where $y = 0$ and $y = 1$ denote that the class label is negative and positive respectively. The parameters $\theta^{(k)}$ are optimized by minimizing the negative log-likelihood loss function

TABLE 2
Four Kinds of Domain Knowledge Learned and Stored for Each Task

	The Domain Knowledge
Basic Statics	$V^{(k)}$, $N^{(k)}$, $N^{(k)}(c_j)$, $N^{(k)}(w_i)$, $N^{(k)}(w_i, c_j)$
Distribution Knowledge	$p^{(k)}(w_i)$, $p^{(k)}(c_j)$, $p^{(k)}(w_i, c_j)$, $p^{(k)}(w_i \mid c_j)$
Sentiment Lexicons	$PMI^{(k)}(w_i, c_j)$, $SO^{(k)}(w_i)$, $L^{(k)}$
Learning Hypotheses	$h^{(k)}$ and $\theta^{(k)}$ of each learning algorithm

$$\min l^{(k)}(\theta) = -\sum_{i=1}^{m} y_i^{(k)} \log h^{(k)}(x_i^{(k)}) + (1 - y_i^{(k)}) \log (1 - h^{(k)}(x_i^{(k)})). \tag{9}$$

For each partition $D^{(k)}$, we minimize the loss function by stochastic gradient descent and store the final parameter $\theta^{(k)}$ of model hypothesis.

Table 2 summarizes four kinds of domain knowledge defined in our approach. For each of the past task, we learn and store such domain knowledge. The knowledge will be updated as the new task comes and finally be used to help future task learning.

4.2 New-Task Learning and Knowledge Updating

So far we have introduced learning in past tasks. Note that the text data in the social media is continuously produced. Once the lifelong learning approach is fed with a new partition $D^{(t)}$, the previous domain knowledge will be updated. This process is called new-task learning.

For a new coming task, we still conduct one-task learning at first and acquire the domain knowledge in the same manner as past-task learning. If the knowledge learned in this new task remains similar to anyone in the past tasks, we merge the knowledge of the two similar tasks. Otherwise, we will consider it as a new "representative" domain, as store the domain knowledge in it. For this purpose, we need to answer the following two questions.

(1) How to measure the similarity of two tasks?
(2) How to merge the knowledge of two similar tasks?

For the first question, to measure the similarity between the current task and a past task, many similarity measures could be used. The type of measure depends on what kind of knowledge is used for calculation. For example, the Jaccard coefficient can be used to measure the similarity of the vocabulary knowledge of two tasks. The Kullback-Leibler (K-L) divergence can be used to measure the similarity of distributional knowledge of two tasks. And the cosine value can be used to measure the similarity of sentiment lexicons as well as the model hypotheses (e.g., weights in logistic regression).

For the second question, by using $D^{(k')}$ to denote the combined task of two similar tasks $D^{(k)}$ and $D^{(t)}$, we present the following knowledge updating rule, in correspondence with the four kinds of knowledge defined in Section 4.1.

(1) Basic Statistics Updating. First, the vocabulary $V^{(k')}$ will be union of $V^{(t)}$ and $V^{(k)}$

$$V^{(k')} = V^{(t)} \cup V^{(k)}, \tag{10}$$

The document frequency in $D^{(k')}$ will be the sum of that in $D^{(k)}$ and $D^{(t)}$

$$N^{(k')}(c_j) = N^{(t)}(c_j) + N^{(k)}(c_j), \tag{11}$$

$$N^{(k')}(w_i) = N^{(t)}(w_i) + N^{(k)}(w_i), \tag{12}$$

$$N^{(k')}(w_i, c_j) = N^{(t)}(w_i, c_j) + N^{(k)}(w_i, c_j). \tag{13}$$

(2) Distribution Knowledge Updating. Second, based on the basic statistic updating rule, we can update the distribution knowledge as follows:

$$p^{(k')}(w_i) = \frac{N^{(t)}(w_i) + N^{(k)}(w_i)}{N^{(k')}}, \tag{14}$$

$$p^{(k')}(c_j) = \frac{N^{(t)}(c_j) + N^{(k)}(c_j)}{N^{(k')}}, \tag{15}$$

$$p^{(k')}(w_i, c_j) = \frac{N^{(t)}(w_i, c_j) + N^{(k)}(w_i, c_j)}{N^{(k')}}, \tag{16}$$

$$p^{(k')}(w_i \mid c_j) = \frac{N^{(t)}(w_i, c_j) + N^{(k)}(w_i, c_j)}{N^{(t)}(c_j) + N^{(k)}(c_j)}. \tag{17}$$

(3) Sentiment Lexicon Updating. Accordingly, the PMI and SO based sentiment lexicon could be updated as follows

$$\begin{aligned} PMI^{(k')}(w_i, c_j) &= \log \frac{p^{(k')}(w_i, c_j)}{p^{(k')}(w_i) p^{(k')}(c_j)} \\ &= \log \frac{(N^{(t)} + N^{(k)})(N^{(t)}(w_i, c_j) + N^{(k)}(w_i, c_j))}{(N^{(t)}(w_i) + N^{(k)}(w_i))(N^{(t)}(c_j) + N^{(k)}(c_j))}, \end{aligned} \tag{18}$$

$$SO^{(k')}(w_i) = PMI^{(k')}(w_i, +) - PMI^{(k')}(w_i, -). \tag{19}$$

By combining the vocabulary and the SO score, we get an updated sentiment lexicon $L^{(k')}$ for the merged task.

(4) Statistical Learning Hypotheses Updating. Suppose $\theta^{(t)}$ and $\theta^{(k)}$ are the parameters of the same classification algorithm obtained in two similar tasks $D^{(t)}$ and $D^{(k)}$. We provide two kinds of classification knowledge updating rule.

- *Online learning rule*. If the classification algorithm supports online learning (e.g., stochastic gradient descent), the new parameters $\theta^{(k')}$ could be obtained by using $\theta^{(k)}$ as the initial parameter, and conducting online learning on the new appearing task $D^{(t)}$. The two classification algorithms used in this work (logistic regression and naïve Bayes) both support online learning.
- *Weighted combination rule*. If the classification algorithm does not directly support online learning (e.g., SVM), we define the following weighted combination rule

$$\theta^{(k')} = \lambda \theta^{(t)} + (1 - \lambda) \theta^{(k)}, \tag{20}$$

where λ is a tradeoff parameter weighting the importance of two tasks. $\lambda = \frac{N^{(t)}}{N^{(t)} + N^{(k)}}$ could be a default value.

The process of new-task learning and knowledge updating are continuously maintained as the new task appears, till the end of the dataset partitions.

4.3 Lifelong Bagging for Future-Task Prediction

Suppose we have already learned the knowledge on several past tasks. In this section, we will introduce how to use the previous knowledge for future-task prediction. We first propose a method similar as BAGGING (bootstrap aggregating) in ensemble learning, to aggregate these knowledge for future-task prediction.

But it should be noted that we do not conduct bootstrapping from an isolate set of labeled data. Since the scale of our dataset is very large, we do not need to re-sample the training data into different partitions, instead, we directly use the dataset partitions defined in Section 3.2 for aggregating.

Given a fixed number of representative past tasks $\{D^{(1)}, \ldots, D^{(t)}\}$ as well as the knowledge learned from them, we could obtain the prediction score $s^{(k)}(x)$ provided by the classifier $h^{(k)}$ in each past task, for each testing example x, where $k = 1, \ldots, t$ is the task index. $s^{(k)}$ is either posterior probability $p^{(k)}(y|x)$, or the log-odds $\theta^{(k)} \cdot x$, depending on the types of outputs in different classification algorithms.

The average score given by all past tasks will be used as the ensemble prediction score

$$f_b(x) = \frac{1}{t}\sum_{k=1}^{t} s^{(k)}(x). \tag{21}$$

Of course, we could also use the voting rule to aggregate different past-learning classifiers, the same as BAGGING.

We can evaluate the lifelong learning model by using a testing dataset for future-task prediction. In the experiments, we train the lifelong sentiment learning model on the large-scale Twitter/Weibo dataset, and test the model on several benchmark sentiment classification datasets.

5 From Lifelong Bagging to Lifelong Stacking

5.1 Motivation of Meta-Learning

In Section 4, we have introduced the Lifelong Bagging model, where the past classifiers are aggregated in an averaging or voting manner, for future-task prediction.

As we have mentioned in the Introduction, the contents of the social media is continuously increasing and the topics are also constantly changing. It is known that sentiment analysis is closely tied to the topics and domains of texts. Although Lifelong Bagging integrates the knowledge gained from different past tasks and thus can improve the generalization ability, the manner of averaging or voting is not adaptive toward different domains.

In ensemble learning, to tackle the shortcomings of BAGGING that uses the average combination or majority voting rule, a STACKING method is developed with a meta-learning layer to re-learn the outputs of the base classifier. The basic idea of meta-learning is to try to find the best prediction mode among the base classifiers.

5.2 The Lifelong Stacking Model

Inspired by that, we furthermore propose a stacking-based lifelong learning method, called Lifelong Stacking, with the aim to improve domain adaptability by using a small set of labeled data in the target domain, to leverage the knowledge from different past tasks.

First, the base-learning layer denotes the learning process of single-task classifiers based on the distantly supervised dataset partitions. For simplicity, we still use the logistic regression classifier as an example to introduce. The method can be easily generalized to the other classification algorithms.

Second, suppose the past classifiers $\{h^{(1)}, \ldots, h^{(t)}\}$ are already learned on the past tasks $\{D^{(1)}, \ldots, D^{(t)}\}$ respectively. In meta-learning, we use each of the classifier $h^{(k)}$ to give the prediction on the target-domain labeled examples. By taking the predictions as features in association with the class labels, we construct a meta-learning training set. The predictive patterns on the target domain thus can be studied by training a meta-learning classifier, so as to adjust the weights of the base classifiers to better suit to the target domain.

Finally, toward a new testing example x, the output of Lifelong Stacking will be a weighted ensemble of predictions from all past tasks

$$f_s(x) = \sum_{k=1}^{t} \beta^{(k)} g_j^{(k)}(x), \tag{22}$$

where $g^{(k)}(x)$ denotes the prediction score provided by $h^{(k)}$ in class j. The component weight $\beta^{(k)}$ is the determined in meta-learning.

The domain adaptive property of Lifelong Stacking can be interpreted as follows [37], [38]. Suppose in each past task we used a linear classifiers $g^{(k)}(x) = \alpha^{(k)} \cdot x$. The stacking learning algorithm can be written as

$$f_s(x) = \sum_{k=1}^{t} \beta^{(k)} \alpha^{(k)} \cdot x, \tag{23}$$

where the primal weight vector $\alpha^{(k)}$ with respect to the kth past task, is converted to $\beta^{(k)}\alpha^{(k)}$. The weights $\beta^{(k)}$ are learned by a meta-learning procedure based on a small labeled set in the future task, so that the past task close to the future task will receive a higher weight in ensemble learning; on the contrary the past task far from the future task will receive a lower weight. It is also worth noting that the learning process on the past tasks is maintained in a continuous learning manner. The knowledge of similar past tasks are combined, and that of distinct past tasks are separated. Thus, the composition of base classifiers is dynamic. In such a way, the base learners can better meet the diversity requirement of ensemble learning. The combination/separation of past tasks are controlled by the similarity measure (such as K-L divergence) that ensures a good complementary condition in Lifelong Bagging and Stacking.

6 Experiments

6.1 Evaluation Datasets

In this section, we report the experimental results of our lifelong sentiment learning approach. The lifelong sentiment classifier is trained on two large-scale social media datasets (one English Twitter dataset and one Chinese Weibo dataset) which were introduced in Section 3, and evaluated on the following nine testing datasets.

(1) For English, we use the well-known SEMEVAL 2013 to 2016 message-level Twitter sentiment classification datasets;

TABLE 3
The Statistics of the Evaluation Datasets

Language	Dataset	Training set size		Testing set size	
		Pos	Neg	Pos	Neg
English	SEMEVAL 2013	1000	1000	1572	601
	SEMEVAL 2014	1000	1000	982	202
	SEMEVAL 2015	1000	1000	1038	365
	SEMEVAL 2016	1000	1000	7059	3231
	ELECTRONICS	800	800	200	200
	KITCHEN	800	800	200	200
Chinese	NLPCC 2013	1000	1000	631	444
	NLPCC 2014	1000	1000	1250	1250
	HOTEL	1000	1000	509	638

(2) For Chinese, we use the Weibo datasets published by the NLPCC 2013 and 2014 Chinese microblog sentiment classification competition. Both datasets are widely used in Chinese social media sentiment analysis;
(3) To test the generalization ability of our approach, we furthermore choose three product review datasets (two domains from the multi-domain sentiment datasets and one domain from the ChnSenticorpus), our training algorithm is totally based on the social media texts.

Table 3 summarizes the statistics of nine evaluation datasets. In Lifelong Bagging, we only use the testing set of these nine datasets for evaluation. In Lifelong Stacking, a small part of the training set is also used as the labeled data in meta-learning. Furthermore, since this work only focuses on the sentiment polarity classification, we remove the neutral class in those datasets if the original datasets are designed for positive-negative-neutral sentiment classification.

6.2 Experimental Settings

(1) Settings of Dataset Partitions. For the English Twitter training dataset, we set the partition size (i.e., the number of microblogs in each task) as 500,000; For the Chinese Weibo training dataset, we set the partition size as 200,000. We conduct lifelong learning over all past tasks sequentially, store the domain knowledge in each task and merge the knowledge when two tasks are similar. Finally, we keep 10 to 15 representative past tasks, and aggregate the knowledge in these tasks for prediction on the testing dataset.

(2) Settings of Learning Algorithms. As mentioned above, the proposed lifelong sentiment learning model is a general framework that could support different single-task learning algorithms. In our work, we conduct experiments by using three single-task sentiment classification methods, including PMI-SO lexicon, logistic regression, naïve Bayes. We will report the performances of the Lifelong Bagging model, of using four different learning algorithms. Before that, we first give a brief description about the settings used in each algorithm.

- *PMI-SO*. In Section 4 we have described how to learn a sentiment lexicon using PMI and SO. The PMI-SO lexicon could be used for a rule-based sentiment classification [9]. PMI-SO lexicon differs from the machine learning methods in that it does not train a classifier and its output is a sentiment score rather than the prediction probability. In each past task, we learn a sentiment lexicon and store it as the lexicon knowledge in that task. As the number of past tasks increases, we update the knowledge and finally establish a number of past PMI-SO lexicons. In the evaluation stage, we get the SO scores of testing examples according to the PMI-SO lexicon, and finally determine the sentiment category according to Equation (6).
- *Logistic Regression*. For logistic regression, we use the Liblinear toolkit[1] with the training parameter "-s 7 -c 1", i.e., using the dual optimization method with L_2 regularization. The weight of the regular term is set as 1. Features are composed of traditional unigrams and bigrams.
- *Naïve Bayes*. For naïve Bayes, we use the XIA-NB toolkit[2], with a setting of multinomial event model and Laplacian smoothing. Similar as logistic regression, both unigrams and bigrams are considered as terms in classification.

6.3 Results of Distantly Supervised Lifelong Bagging

We report the experimental results of Lifelong Bagging in terms of classification performance, computational efficiency and parameter sensitivity study, respectively.

6.3.1 Comparison of Classification Performance

The F_1 score is used as the evaluation metric which is calculated as

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{24}$$

where P and R denote the averaged precision and recall score of each category. A paired t-test is used to check the significance of differences in two compared systems.

Table 4 displays the results of Lifelong Bagging using different single-task algorithms in each past-task learning. For comparison, we also report the results of the same classification algorithms using the entire training data for single-task learning. We denote it as Entire Learning.

First, we compare different single-task classification algorithms in Entire Learning. PMI-SO Lexicon shows the lowest performance, followed by naïve Bayes, and logistic regression performs better. The same conclusion can be drawn by comparing different algorithms in Lifelong Bagging.

Second, we compare Entire Learning and Lifelong Bagging. It can be seen that Lifelong Bagging shows significantly and robustly better performance than Entire Learning. For instance, on the NLPCC 2013 dataset, the F_1 score of Lifelong Bagging is 1.54, 0.1 and 3.09 percent higher than Entire Learning in three kinds of classification algorithms respectively. By aggregating the knowledge gained from different past tasks, Lifelong Bagging significantly outperforms traditional way of using entire training data in one task.

1. https://www.csie.ntu.edu.tw/~cjlin/liblinear
2. https://github.com/rxiacn/xia-nb

TABLE 4
The Classification Performance (Lifelong Bagging versus Entire Learning)

	Entire Learning			Lifelong Bagging		
	PMI-SO Lexicon	Naïve Bayes	Logistic Regression	PMI-SO Lexicon	Naïve Bayes	Logistic Regression
SEMEVAL 2013	74.31	75.13	75.14	74.86	75.41	76.03
SEMEVAL 2014	78.57	79.99	79.91	79.47	80.07	81.02
SEMEVAL 2015	72.08	72.89	72.76	73.59	73.64	74.88
SEMEVAL 2016	70.26	70.34	70.85	70.74	70.73	72.53
ELECTRONICS	72.04	73.07	78.13	74.82	74.89	76.87
KITCHEN	69.16	72.19	77.19	72.21	72.95	79.06
NLPCC 2013	76.49	77.45	76.63	78.03	77.55	79.72
NLPCC 2014	68.14	70.65	69.38	69.17	70.89	72.47
HOTEL	76.92	81.01	79.16	77.18	81.09	81.32

6.3.2 Sensitivity of Dataset Partition Parameters

In lifelong learning, the number of past tasks and the size of each past task are two important parameters. This section discusses their influence on the system performance.

(1) *Changing the Number of Past Tasks*. Figs. 2 and 3 display the relation between sentiment classification performance and the number of past tasks. We still take the results of using logistic regression for example. Similar conclusions can be drawn when using the other classification algorithms. We fix the size of each dataset partition as 300,000, and display the classification performance of the following three classifiers when the number of past tasks increases:

- *Current Learning classifier* which is trained on the latest past task (current task);
- *Entire Learning classifier* which is trained on the joint dataset of all past tasks;
- *Lifelong Learning classifier* which is trained on all past tasks sequentially.

First, the performance of Current Learning classifier is the lowest and inconsistent in different past tasks.

Second, the Entire Learning classifier is better than Current Learning classifier. But the conclusion that "the more training data the better performance" seems not hold in large-scale social media sentiment analysis. It shows that a single-task learning algorithm may not fully capture the knowledge in the big text data. It is known that sentiment classification is a domain-sensitive task. In statistical machine learning, it is typically assumed that the labeled training data and the test data should be drawn from the same underlying distribution. Under this assumption, we think the more training data will result in better machine learning performance. But in the scenario of large-scale social media, the text data is continuously increasing and the hot topics/domains are constantly changing. The test domain might differ from the source domains in social media sentiment analysis. In this case, more training data, especially when the training data are not in-domain, will not be helpful to improve the sentiment classification performance.

Third, the Lifelong Learning classifier performs consistently the best. As the number of past tasks increases, its performance increases significantly in the early stage and becomes more stable finally.

(2) *Changing the Size of Each Past Task*. We then discuss the lifelong learning performance when choosing different sizes of dataset partition. For simplicity, we fix the number of past tasks at 10 and compare the performance of different partition size in Fig. 4.

We can see that when the partition size becomes larger, the performance of Lifelong Bagging increases accordingly. But the improvement becomes less as the size increases gradually. It shows that the size of each past task cannot be

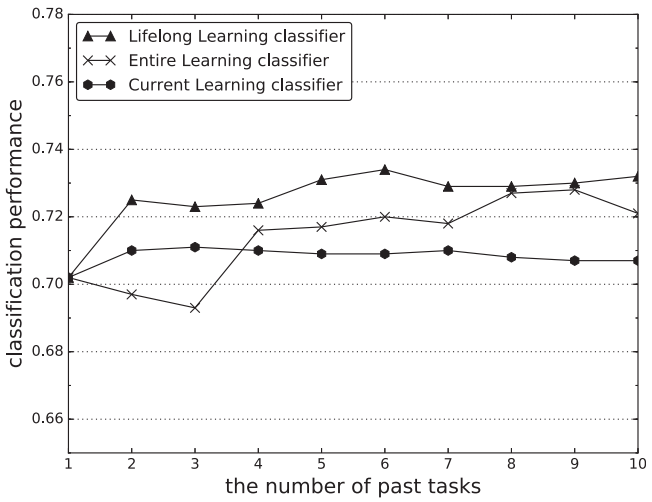


Fig. 2. The relation between sentiment classification performance and number of past tasks on SEMEVAL 2015.

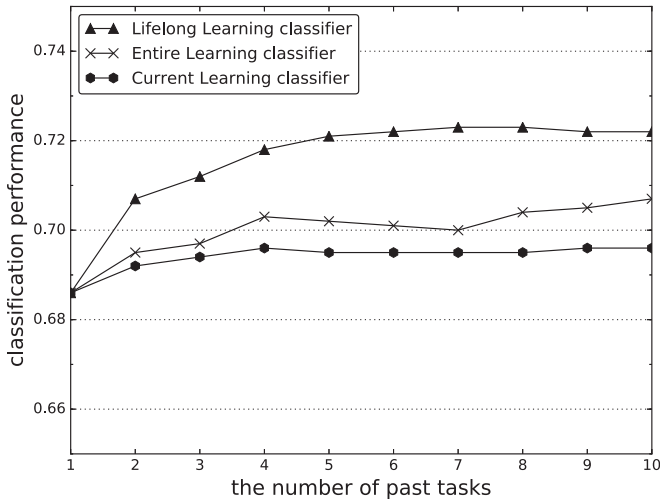


Fig. 3. The relation between sentiment classification performance and number of past tasks on SEMEVAL 2016.

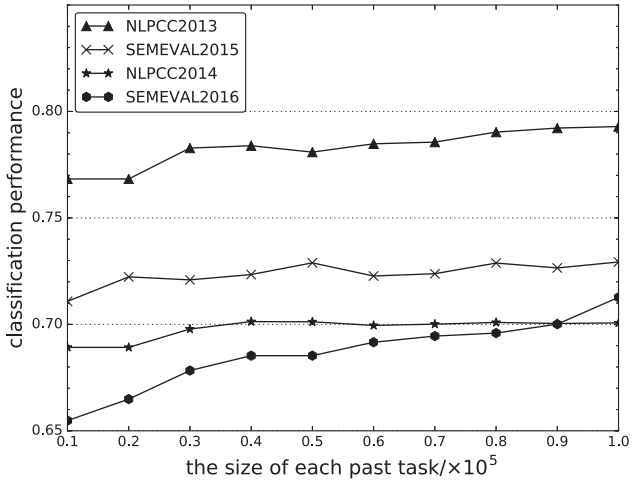


Fig. 4. The relation between sentiment classification performance and the size of each past task.

too small, probably due to that the model is trained by distant supervision on the noisy dataset, and therefore the single-task learning quality is not guaranteed if the size of each past task is too small.

6.3.3 Comparison of Computational Efficiency

Fig. 5 displays the training time, when the amount of training data gradually increases. The x-axis denotes the number of dataset partitions, and y-axis denotes the seconds used in training. The experiment is conducted on a server with a 3.0 GHz CPU and 64 GB RAM.

Seen from the figure, the Entire Learning system has a quadratic growth of training time, as the number of past tasks increases. That is because that it has to add the new data into the previous ones and re-train a new classifier on the total training set. The quadratic growth in computational load make it difficult to apply traditional Entire Learning to large-scale sentiment analysis.

In comparison, Lifelong Bagging's training time has a roughly linear increase, as it only needs to learn on the new appearing task, and the knowledge updating computation is also efficient. Both aspects lead to much lower computational cost especially when the scale of training dataset is large. It

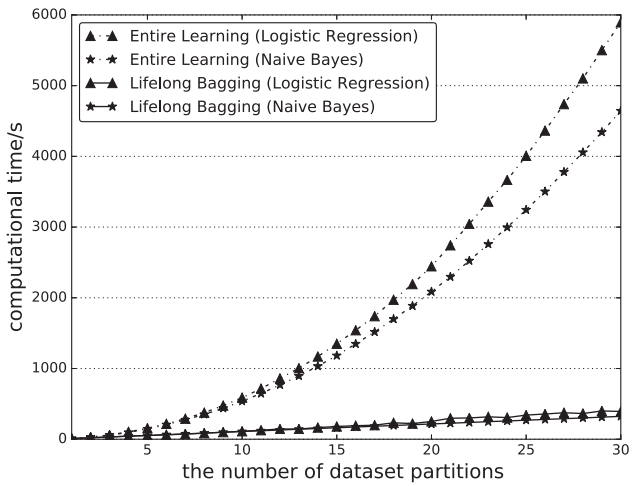


Fig. 5. The training time (Lifelong Bagging versus entire learning).

TABLE 5
The Classification Performance (Lifelong Stacking versus Lifelong Bagging versus In-Domain Training)

	Lifelong Bagging	In-domain Training	Lifelong Stacking
SEMEVAL 2013	76.03	72.89	80.30
SEMEVAL 2014	81.02	73.41	82.36
SEMEVAL 2015	74.88	68.97	76.02
SEMEVAL 2016	72.53	71.05	77.25
ELECTRONICS	76.87	80.28	83.11
KITCHEN	79.06	78.01	86.75
NLPCC 2013	79.72	79.72	83.02
NLPCC 2014	72.47	71.44	74.52
HOTEL	81.32	87.74	88.17

shows that our lifelong sentiment learning approach has a good scalability in big-data social media sentiment analysis.

6.4 Results of Distantly Supervised Lifelong Stacking

In this section, we report the experimental results of Lifelong Stacking. As an extension to Lifelong Bagging, Lifelong Stacking is motivated to establish a more robust and adaptive model, with the help of a small labeled set in the target domain.

The experimental settings of Lifelong Stacking are similar as that in Lifelong Bagging, except for an extra meta-learning process. In the following, we first report its performance and discuss the influence of the size of the target-domain labeled data in meta-learning.

6.4.1 Comparison of System Performance

Lifelong Stacking uses two kinds of labeled information:

- a large-scale distantly supervised dataset;
- a small manually labeled dataset in the target domain.

Table 5 reports the results of three methods:

- *Lifelong Bagging* (which only uses the distantly supervised data);
- *In-domain Training* (which only uses the target-domain labeled data);
- *Lifelong Stacking* (which uses both distantly supervised and target-domain labeled data).

In Lifelong Stacking we still only report the result of using logistic regression as the single-task learning algorithm. Conclusions are similar when using the other learning algorithms.

First, it can be seen that although learned from distantly supervised data, Lifelong Bagging can still outperform the in-domain system learned on a small set of manually labeled data.

Second, by making the use of two kinds of supervision, Lifelong Stacking can robustly achieve the best performance than Lifelong Bagging and In-domain Learning.

6.4.2 Discussion on the Size of Target-Domain Labeled Set

Finally, we observe the influence of the size of target-domain manually labeled set on Lifelong Stacking. We set

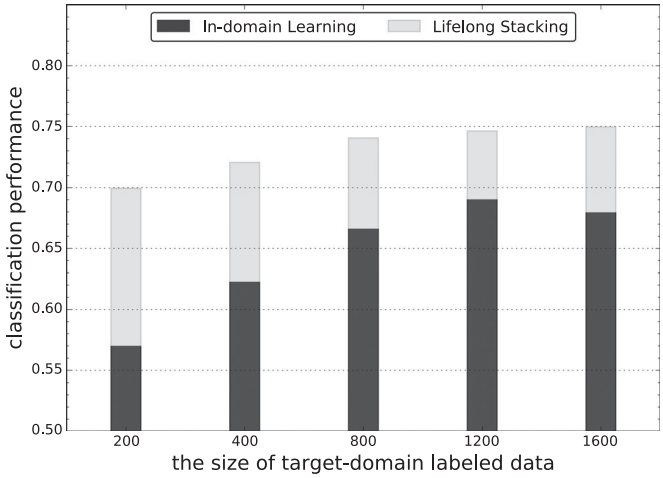


Fig. 6. The relation between Lifelong Stacking's performance and the size of target-domain labeled data on SEMEVAL 2016.

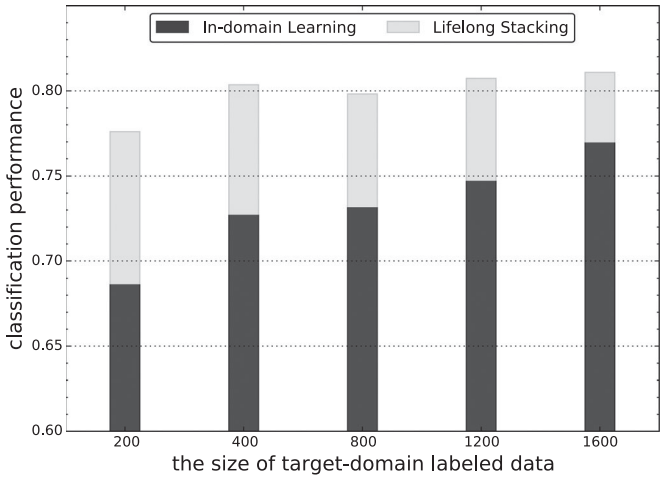


Fig. 7. The relation between Lifelong Stacking's performance and the size of target-domain labeled data on NLPCC 2013.

its size as 200, 400, 800, 1,200 and 1,600 respectively. Figs. 6 and 7 report the performance of In-domain Learning and Lifelong Stacking, on the testing datasets of SEMEVAL 2016 and NLPCC 2013, respectively.

We can observe that when the size of target-domain labeled set increases, the performance of In-domain Learning improves significantly, but such improvements are less significant in Lifelong Stacking. Also, the extra improvement of Lifelong Stacking against In-domain learning becomes less. It suggests Lifelong Stacking has the following good property: although the improvement of Lifelong Stacking is at the cost of a set of manually labeled data, it has little dependence on the size of manually labeled data. That is, Lifelong Stacking can improve the system performance, even with a relatively small set of manually labeled data in the target domain.

7 Conclusions

Traditional learning algorithms based on single tasks can often exhibit good performance in small-scale sentiment analysis. However, in the environment of large-scale social media, it would be very difficult to construct a universal sentiment analysis model by using single-task learning algorithms. In this work, we propose a distantly supervised lifelong learning approach, to address such challenges in large-scale social media sentiment analysis.

On one hand, our approach has the characteristic of continuous sentiment learning from the social media texts under distant supervision. It can store the knowledge learned from past tasks, and continuously update the knowledge as new tasks appear. On the other hand, it inherits the advantages of BAGGING and STACKING in ensemble learning, by integrating the knowledge learned in the past tasks for future-task prediction. Furthermore, our approach provides a general framework that can adapt any single-task sentiment learning algorithms to the scene of lifelong sentiment learning.

We evaluate our approach by training the model on two large-scale distantly supervised social media datasets in languages of both English and Chinese, and testing it on nine benchmark sentiment analysis datasets. The experimental results prove our approach's feasibility and efficiency in large-scale social media sentiment analysis.

Acknowledgments

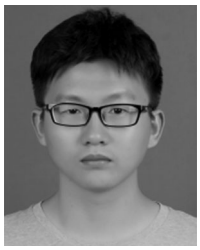
The work was supported by the Natural Science Foundation of China (No. 61672288), and the Natural Science Foundation of Jiangsu Province for Excellent Young Scholars (No. BK20160085).

References

[1] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Rep., Stanford University, 2009.
[2] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures Artif. Intell. Mach. Learn.*, vol. 10, no. 3, pp. 1–145, 2016.
[3] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research [review article]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 48–57, May 2014.
[4] R. Xia, F. Xu, J. Yu, Y. Qi, and E. Cambria, "Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis," *Inf. Process. Manag.*, vol. 52, no. 1, pp. 36–45, 2016.
[5] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar./Apr. 2016.
[6] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*. Berlin, Germany: Springer, 2017.
[7] E. Kouloumpis, T. Wilson, and J. D. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *Proc. Int. Conf. Weblogs Social Media*, 2011, pp. 538–541.
[8] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," in *Proc. Workshop Languages Social Media*, 2011, pp. 30–38.
[9] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets," in *Proc. 7th Int. Workshop Semantic Eval.*, 2013, pp. 321–327.
[10] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics: Human Language Technol.* vol. 1, 2011, pp. 151–160.
[11] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of Twitter," in *Proc. Int. Semantic Web Conf.*, 2012, pp. 508–524.
[12] X. Hu, L. Tang, J. Tang, and H. Liu, "Exploiting social relations for sentiment analysis in microblogging," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 537–546.
[13] S. Mukherjee, A. Malu, B. AR, and P. Bhattacharyya, "Twisent: A multistage system for analyzing sentiment in Twitter," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manag.*, 2012, pp. 2531–2534.
[14] F. Liu, F. Weng, and X. Jiang, "A broad-coverage normalization system for social media language," in *Proc. 50th Annu. Meet. Assoc. Comput. Linguistics: Long Papers*, 2012, pp. 1035–1044.

[15] V. M. K. Peddinti and P. Chintalapoodi, "Domain adaptation in sentiment analysis of Twitter," in *Proc. 5th AAAI Conf. Analyzing Microtext*, 2011, pp. 44–49.
[16] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. 7th Conf. Int. Language Resources Eval.*, 2010, pp. 1320–1326.
[17] J. Zhao, L. Dong, J. Wu, and K. Xu, "Moodlens: An emoticon-based sentiment analysis system for Chinese tweets," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1528–1531.
[18] K.-L. Liu, W.-J. Li, and M. Guo, "Emoticon smoothed language models for Twitter sentiment analysis," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1678–1684.
[19] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale Twitter corpus," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manag.*, 2012, pp. 1980–1984.
[20] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1555–1565.
[21] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A deep learning system for Twitter sentiment classification," in *Proc. 8th Int. Workshop Semantic Eval.*, 2014, pp. 208–212.
[22] D.-T. Vo and Y. Zhang, "Target-dependent Twitter sentiment classification with rich automatic features," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 1347–1353.
[23] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 959–962.
[24] S. Thrun, "Is learning the N-th thing any easier than learning the first?" in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1996, pp. 640–646.
[25] D. L. Silver, Q. Yang, and L. Li, "Lifelong machine learning systems: Beyond learning algorithms," in *Proc. AAAI Spring Symp.: Lifelong Mach. Learn.*, 2013, pp. 49–55.
[26] Y. Gu, J. Liu, Y. Chen, and X. Jiang, "Constraint online sequential extreme learning machine for lifelong indoor localization system," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 732–738.
[27] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 283–291.
[28] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[29] R. Caruana, "Multitask learning," in *Proc. Learning Learn Conf.*, 1998, pp. 95–133.
[30] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1306–1313.
[31] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 759–766.
[32] L. Bottou, "Online learning and stochastic approximations," *On-Line Learn. Neural Netw.*, Cambridge University Press: New York, NY, USA, pp. 9–42.
[33] T. Mitchell, et al., "Never-ending learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2302–2310.
[34] Z. Chen, N. Ma, and B. Liu, "Lifelong learning for sentiment classification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2015, pp. 750–756.
[35] D. T. Vo and Y. Zhang, "Don't count, predict! an automatic approach to learning sentiment lexicons for short text," in *Proc. 54th Annu. Meet. Assoc. Comput. Linguistics*, 2016, pp. 219–224.
[36] K. Gimpel, et al., "Part-of-speech tagging for Twitter: Annotation, features, and experiments," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics: Short Papers*, 2011, pp. 42–47.
[37] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Inf. Sci.*, vol. 181, no. 6, pp. 1138–1152, 2011.
[38] R. Xia, C. Zong, X. Hu, and E. Cambria, "Feature ensemble plus sample selection: Domain adaptation for sentiment classification," *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 10–18, May/Jun. 2013.

Rui Xia received the PhD degree from the Institute of Automation, Chinese Academy of Sciences, in 2011. He is currently a professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He directs the Text Mining Group at Nanjing University of Science and Technology (NUSTM). His research interests include natural language processing, machine learning, and data mining.



Jie Jiang received the BSc degree from Nanjing University of Science and Technology, in 2014. He is currently working toward the master's degree in computer science and engineering at Nanjing University of Science and Technology, China. His research interests include natural language processing and text mining.

Huihui He received the BSc degree from Nanjing University of Science and Technology, in 2016. She is currently working toward the master's degree in computer science and engineering at Nanjing University of Science and Technology, China. Her research interests include natural language processing and text mining.