J. Shanghai Jiao Tong Univ. (Sci.), 2017, 22(3): 291-302
DOI: 10.1007/s12204-017-1835-3

Semi-Supervised Learning in Large Scale Text Categorization

XU Zewen[1,2] (许泽文), *LI Jianqiang*[1,2,3,4*] (李建强), *LIU Bo*[1] (刘 博)
BI Jing[1] (毕 敬), *LI Rong*[1] (李 蓉), *MAO Rui*[3,4] (毛 睿)
(1. School of Software Engineering, Beijing University of Technology, Beijing 100124, China; 2. Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, Beijing 100124, China; 3. Guangdong Key Laboratory of Popular High Performance Computers, Shenzhen University, Shenzhen 518060, Guangdong, China; 4. Shenzhen Key Laboratory of Service Computing and Applications, Shenzhen University, Shenzhen 518060, Guangdong, China)

Abstract: The rapid development of the Internet brings a variety of original information including text information, audio information, etc. However, it is difficult to find the most useful knowledge rapidly and accurately because of its huge number. Automatic text classification technology based on machine learning can classify a large number of natural language documents into the corresponding subject categories according to its correct semantics. It is helpful to grasp the text information directly. By learning from a set of hand-labeled documents, we obtain the traditional supervised classifier for text categorization (TC). However, labeling all data by human is labor intensive and time consuming. To solve this problem, some scholars proposed a semi-supervised learning method to train classifier, but it is unfeasible for various kinds and great number of Web data since it still needs a part of hand-labeled data. In 2012, Li et al. invented a fully automatic categorization approach for text (FACT) based on supervised learning, where no manual labeling efforts are required. But automatically labeling all data can bring noise into experiment and cause the fact that the result cannot meet the accuracy requirement. We put forward a new idea that part of data with high accuracy can be automatically tagged based on the semantic of category name, then a semi-supervised way is taken to train classifier with both labeled and unlabeled data, and ultimately a precise classification of massive text data can be achieved. The empirical experiments show that the method outperforms the supervised support vector machine (SVM) in terms of both F1 performance and classification accuracy in most cases. It proves the effectiveness of the semi-supervised algorithm in automatic TC.

Key words: text data mining, semi-supervised, automatic tagging, classifier
CLC number: TP 391.1, TP 311 **Document code:** A

0 Introduction

In recent years, a great deal of textual information appears on the World Wide Web and institutional document repositories. However, it is hard for people to acquire the information they really need. Text mining is used to quickly get useful knowledge from large text database and is becoming more and more popular. It is mainly based on text categorization (TC) technology which is used to automatically classify text documents into a set of given categories.

Text classification technology has aroused scholars' great interest for its widely using in natural language processing field[1]. Miyato et al.[2] extended adversarial training and virtual adversarial training to the text domain by applying perturbations to the word embedded in a recurrent neural network rather than to the original input itself. They found that adversarial training and virtual adversarial training have good regularization performance in sequence models on text classification tasks. By pretreating the short text set and selecting the significant features, Yin et al.[3] used semi-supervised learning and support vector machine (SVM) to mine the useful message from the short text. Using a new semi-supervised framework with convolutional neural networks (CNNs) for TC, our method proposed in this paper learns embedding of small text regions from unlabeled data for integration into a supervised CNN[4]. Johnson and Zhang[5] explored a more sophisticated region embedding method using long short-term memory (LSTM). LSTM can embed text regions of variable (and possibly large) sizes, whereas the region

Received date: 2016-07-25
Foundation item: the National Key Technology Research and Development Program of China (No. 2015BAH13F01), and the Beijing Natural Science Foundation (No. 4152007)
∗**E-mail:** lijianqiang@bjut.edu.cn

size needs to be fixed in a CNN. They were seeking effective and efficient use of LSTM for this purpose in the supervised and semi-supervised settings.

TC technology plays an important role in classifying the large electronic documents efficiently in multiple sources[6]. It is mainly based on machine learning which can determine the category via using example data or past experience.

Machine learning contains three learning styles: supervised learning, semi-supervised learning and unsupervised learning. Supervised learning and semi-supervised learning are mainly used for classification, while unsupervised learning is mainly used for clustering.

1 Machine Learning

1.1 Supervised and Semi-Supervised Learning Approaches Based on Manual Labeling

Supervised learning is one of the most common methods in automatic TC[6]. It uses the name of the species as a symbolic label, and then these labeled documents are used as training data to supervise the classifier learning[6]. Figure 1 shows the schematic flowchart of a supervised learning way. The knowledge obtained by this learning almost comes from the labeled data. Since a huge number of data need to be labeled by human themselves, it would spend a lot of time and energy, especially when the number of category names is very much.

To avoid manual labeling, some people proposed semi-supervised learning approach[6], as shown in Fig. 2. Both labeled data and unlabeled data are used to acquire valuable knowledge. However, even though only a small set of labeled data of each category is used for training classifier, it is still difficult for people to tag various kinds of Web data.

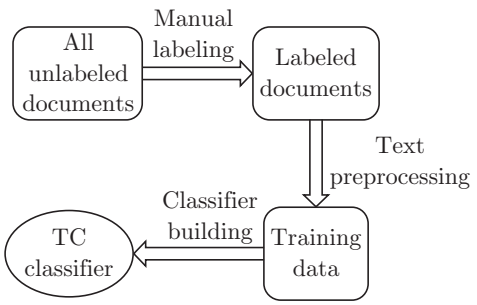

Fig. 1 Supervised learning approach

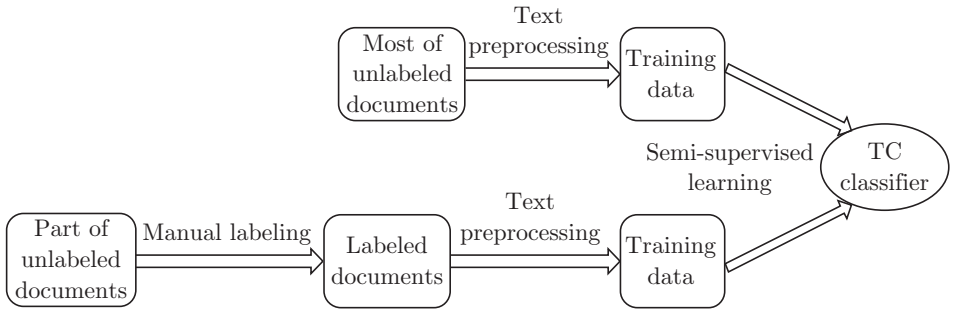

Fig. 2 Semi-supervised learning approach

1.2 Supervised Learning Based on Automatic Labeling

So, a feasible approach investigated for the large-scale TC problem has been proposed by Li et al[1]. The approach called fully automatic categorization approach for text (FACT), as shown in Fig. 3, combines the semantic analysis of the category name and the statistical analysis of the unlabeled document set. It can realize fully automatic TC by automatically annotating textual data, based on external semantic resources like WordNet and HowNet. Thanks to automatic document labeling, all labeled documents can be acquired without manually labeling data any more. Then, those labeled data are used to supervise the classifier leaning. However, automatically tagging all data certainly introduces noise information, and causes the fact that the result cannot meet the accuracy requirement.

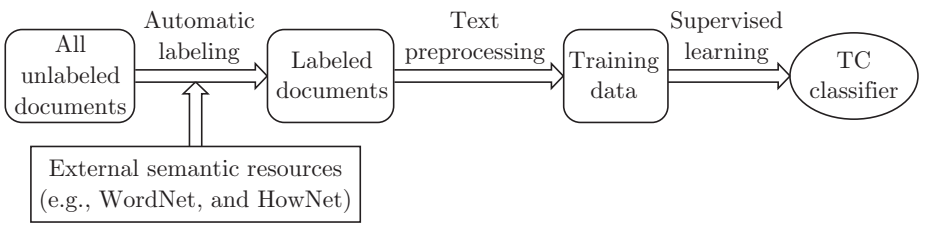

Fig. 3 FACT

1.3 Semi-Supervised Learning Based on Automatic Labeling

In this paper, we put forward a new method to solve the problem. Firstly, part data are tagged automatically to guarantee the accuracy of tag via filtering. Next, the classifier is trained by the way of semi-supervised learning using a large number of unlabeled data and a part of labeled data. Then, we can obtain a text classifier with high precision. Our approach can not only ensure the accuracy of text classification, but also avoid manual data labeling. Eventually we can achieve the high precision of text classification under huge amounts of Web data.

Our method is based on the assumption that the category name is specified explicitly in human-understandable words. This assumption is usually correct in many real applications via good human-computer interfaces.

Figure 4 shows the schematic flowchart of our approach. Firstly, a certain number of text documents are selected and those documents are automatically tagged based on the external semantic resources. Then, we can obtain a small part of labeled training data by selecting those data which are labeled with high accuracy. Secondly, we combine the labeled data and most of the unlabeled training data as the training data. Thirdly, we use the training data to train the classifier by the way of semi-supervised learning. Finally, we obtain a text classifier that can classify huge amounts of data such as Web data with high accuracy.

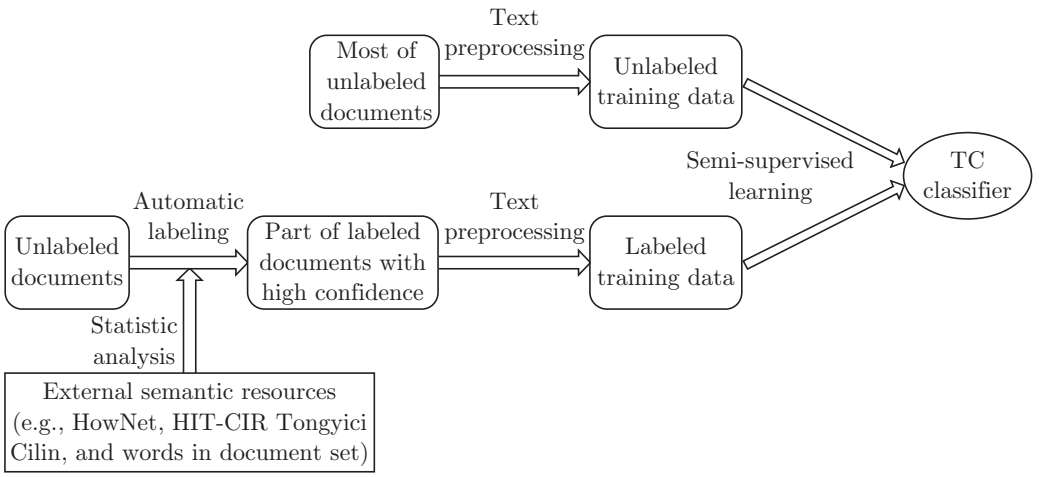

Fig. 4 Semi-supervised TC

The lexical databases (WordNet and HowNet) provide explicit and formal semantics on human-understandable words[1]. We use such lexical databases as external semantic resources to obtain the knowledge of the category name and generate a set of features for the corresponding category. Those features are employed as extended feature for each category. At the same time, a set of features are extracted from the documents as corresponding feature vector. Then, we calculate the similarity between each category name and each text document and rank the documents by similarity. Finally, we classify the documents into corresponding category by the calculated similarity. To guarantee the accuracy of the experimental result, we only select the data labeled with high similarity as our labeled training data.

The core module of our method is automatic tagging of part text documents with high precision, through which we can get part of correct labeled training data. It is important for semi-supervising of classifier learning. Generally speaking, this idea derives from two observations.

Firstly, given a category name, the prerequisite for the human experts (or experienced users) to manually label the documents as training data is that they know the intensions or meanings of the concepts implied by the category name[1]. They can assign the category label on the documents they read by the background knowledge in their mind. The background knowledge about the category name in their mind can serve as a bridge to link the category label with the content of text documents together. Similarly, with the availability of the semantic resources like WordNet and HowNet, we can adopt an intuitive way to simulate the humans' document labeling ability. These external semantic resources play the same role as the background knowledge in the experts' brains.

Secondly, the above method implementing the functionality of automatic document labeling is based on calculating the similarity between the category name and the document content in fact. Usually, those documents that have high similarity with one category name are correctly assigned to the category. The higher the similarity is, the more correct the labeling result is. So,

to ensure that the training data are labeled correctly, we only choose those labeled documents that have high similarity with one category name and low similarity with other category name as our labeled training data.

This paper mainly presents our experimental study on two aspects: ① automatically labeling documents accurately through category name understanding; ② data filtrating via similarity between the category name and the document content which is used to obtain part labeled training data and finally implement semi-supervised TC.

The contribution of this paper can be summarized as two points. ① A new method is proposed to obtain part of correct labeled training data automatically. This new method is based on category name understanding and data filtrating. ② We implement a precise classification of huge amounts of data (Web data) by semi-supervising the classifier learning with both labeled training data and unlabeled training data. The empirical experiments of TC on Chinese datasets are reported. It shows that the best performance of our method can outperform the supervised SVM in terms of both F1 performance and classification accuracy in most cases, which demonstrates the effectiveness of our proposed method.

2 Related Work

Generally speaking, there have three main technologies used for implementing automatic TC, i.e., supervised, semi-supervised, and unsupervised approaches. Since the unsupervised approach mainly involves clustering knowledge, we only discuss the first two algorithms in the following sections.

As plenty of original tagged documents are transformed into a set of labeled training data, supervised learning methods can use a deductive process to build the TC classifier based on the characteristics of the hidden knowledge in training data. In the field of text mining, there are several popular supervised methods that can be utilized to construct a classifier, such as k-nearest neighbor (KNN) algorithm, linear regression, logistic regression, decision tree, SVM, Naïve Bayes (NB) and neural networks.

Since in supervised methods, a huge number of data need to be labeled by people themselves, it is a waste of time and energy. In order to overcome this bottleneck, semi-supervised methods such as expectation-maximization (EM) algorithm, and transductive support vector machine (TSVM) for TC are more and more attractive[7]. They can make use of the valuable information hidden in the plenty of unlabeled data. Then, a text classifier with high precision can be obtained even with insufficient labeled training data[6].

Even though only small size labeled data are needed for each category, it is still difficult for semi-supervised methods to be applied in Web-scaled TC tasks because of a large number of categories[1]. FACT approach was proposed to realize full automatic TC based on the lexical databases such as WordNet and HowNet[1]. It mainly uses the embedded functionality of automatic document labeling to label all documents and then uses the labeled training data to supervise the classifier learning.

Since automatically tagging all data introduces noise, the classification result can hardly meet the accuracy requirement. On the basis of the predecessors, we put forward a semi-supervised method that full automatic TC can be realized by the method based on automatic document labeling. Our method is different in two aspects.

(1) In terms of automatic document labeling, Li et al.[1] only utilized one external semantic resource (WordNet or HowNet) to automatically generate a set of representative words as the extended features of the category. Then, the extended features were used to initially label a set of documents. Our method utilizes two components to generate the extended feature for each category name. The first component is comprised of those words that have high similarity with the category name in original documents. The second component is comprised of those words generated from multiple external semantic resources such as HowNet APIs, HIT-CIR Tongyici Cilin (extended) and Baidu search engine. After deleting the uncorrelated words, combining two components and deleting those repeating words, we finally obtain the extended feature for each category name.

(2) In terms of semi-supervised learning, the labeled training data are obtained by automatic document labeling and with high quality at the same time. We not only reduce the burden of manual labeling, but also finally construct a TC classifier with high classification accuracy. Those labeled training data with high quality are obtained based on filtering of the labeled documents.

Our method is related to the study on applying different semantic resources to enhance TC performance. Siolas and D'Alché-Buc[8] used synonym and hypernym to define a new metric between two documents, through which the semantic knowledge of WordNet is introduced into text representation and text classifiers. Basili et al.[9] applied the conceptual density function to the WordNet hierarchy to define a document similarity metric. And then the semantic kernel was constructed to train SVM classifiers. Through identifying the most related encyclopedia articles for every document, Gabrilovich and Markovitch[10] utilized the concepts that correspond to these articles to produce new features so as to improve the document representation for TC. Wang and Domeniconi[11] proposed a method to embed background knowledge obtained from Wikipedia

into a semantic kernel. And then it was used to enrich the document representation for TC.

Our method is also related to the research on applying semi-supervised algorithms in TC. Semi-supervised learning[12] deals with the methods that attempt to automatically exploit unlabeled data, where the unlabeled data are usually different from the test data. Transductive learning deals with the methods that also attempt to automatically exploit unlabeled data but assume that the unlabeled data are exactly the test data[7].

The research described in this paper is different with the above work because we combine automatic document labeling based external semantic resources with semi-supervised algorithms to build a text classifier. Since our method focuses on exploiting the original documents and the external semantic resources to construct extended features from the category name, we can select part of labeled documents with high quality and then combine plenty of unlabeled documents to automatically construct the required training data. Finally, we can use the training data to build the classifier and implement semi-supervised TC.

3 Semi-Supervised TC

In this section, we use HowNet, HIT-CIR Tongyici Cilin (extended) and Baidu search engine as the example semantic resources and use WordSimilarity.exe as the word similarity computing tool. Then, we illustrate our semi-supervised learning approach for Chinese-TC.

At the beginning, there is a document set D where all documents are sorted into a set of categories. In our experiments, we totally have two categories. For the experimental requirement, the document set D is equally divided into two parts, i.e., D_1 and D_2. Documents in D_1 are treated as unlabeled training data (the initial label of those documents will be deleted after dividing) that will be used to train the classifier after being processed. Documents in D_2 are treated as labeled test data that will be used to evaluate the effectiveness of binary TC classifier after being processed.

Considering a set of categories C (each category $c_j \in C$) and a set of documents D_1 (each document $d_i \in D_1$), our method consists of five steps to implement classifying each d_i into the right c_j: ① category name extension; ② document labeling; ③ filtering of the labeled documents; ④ training data and test data construction; ⑤ classifier building.

3.1 Category Name Extension

Given a set of categories C, each category $c_j \in C$ has a certain name. A category name contains one or multiple words, where each word corresponds to a set of relevant words (synonyms and hyponyms). In other words, a category name covers one or more concepts. So, our goal in this step is to find the relevant concepts of the word appearing in the category name and use those concepts as the extended feature to represent the category.

Since Chinese category names are usually simple, part-of-speech (POS) tagging, word segmentation and stop-words removing are ignored in category name processing while these steps play an important role in the document preprocessing. Also, there is no polysemy in these category names, so we do not need an algorithm to eliminate ambiguous meanings of the words in category names.

There are totally two components to generate our extended features for each category.

(1) For all document d_i in unlabeled document set D_1, after POS tagging, word segmentation and stop-words removing, we can construct the bag-of-words model of the document set D_1 and then retain a word list which contains all words in document set D_1 except stop-words. Those words that have high similarity with the category name calculated by WordSimilarity.exe are selected as the first component to constitute the extended features of each category. The other words will be deleted.

(2) To get the second component, we first collect the synonyms, hyponyms and another relevant words using HowNet, HIT-CIR Tongyici Cilin (extended) and Baidu search engine, and then rank them by their semantic similarities with corresponding category name's synset[1]. To guarantee the quality of the extended features, we delete those words that have little similarity with category name's synset especially if this category name has plenty of hyponyms and relevant words. After deleting those uncorrelated words, we obtain the second component.

Combining the two components and deleting those repeating words, we can finally obtain the extended features of the category name.

3.2 Document Labeling

Once the extended feature for each category is constructed, then it is used for document labeling. The implementation is based on the similarity between each document d_i and the extended feature f_j of each category c_j.

For all documents in the unlabeled document set D_1, after POS tagging, word segmentation, stop-words removing and feature selection, we can get a feature directory DC that contains all feature words in the document set D_1, and each word in DC is labeled with a serial number. Each document d_i is represented as a vector by using the vector space model (VSM), i.e., vector $\boldsymbol{v}(d_i) = (\varpi_{1i}, \varpi_{2i}, \cdots, \varpi_{ti})$, where each $\varpi_{ji}\,(1 \leqslant j \leqslant t)$ corresponds to a single word and t is the maximum length of the vector $\boldsymbol{v}(d_i)$. If a word occurs in d_i, its weight value in the vector is non-zero. Given document set D, the term frequency inverse document frequency (TF-IDF) model is adopted to compute these values.

Similarly, for each f_j, those words in f_j that do not appear in the DC will be deleted, and then the remaining words will be ranked by the order of DC. Finally, we can build vector $\boldsymbol{v}(f_j) = (\varpi_{1j}, \varpi_{2j}, \cdots, \varpi_{tj})$ for each f_j, where the TF-IDF weight is computed by treating each f_j as a document and all the extended features as the document set[1].

After the two vectors, i.e., $\boldsymbol{v}(d_i)$ and $\boldsymbol{v}(f_j)$, are obtained, we use the cosine similarity metric to calculate the similarity between d_i and f_j. The resulting score $s(d_i, c_j)$ is obtained to indicate semantic distance between d_i and c_j.

Cosine Similarity This metric is frequently applied to determining similarity between two documents. Since there are many words that are in common between two documents, it is useless to use the other methods (namely the Euclidean distance and the Pearson's correlation coefficient discussed earlier) to calculate the semantic distance between two documents. As a result, the likelihood that two documents do not share the majority is very high (the same as the Tanimoto coefficient) and does not create a satisfactory metric for determining similarities. In this similarity metric, the attributes (or words, in the case of the documents) are used as a vector to find the normalized dot product of two documents $\boldsymbol{x}$ and $\boldsymbol{y}$. By determining the cosine similarity, the user tries to effectively find the cosine of the angle θ between the two objects. For cosine similarities resulting in a value of 0, the documents do not share any attributes (or words) because the angle between the objects is 90°. Cosine similarity is expressed as

$$\text{similarity}(\boldsymbol{x}, \boldsymbol{y}) = \cos\theta = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| * \|\boldsymbol{y}\|}.$$

For each $c_j \in C$ and $1 \leqslant k \leqslant |D|$, assuming s_j is the maximum $s(d_k, c_j)$, we define the confidence that document d_i is classified into category c_j as $T(c_j|d_i) = s(d_i, c_j)/s_j$. There are many cases that, for one document d_i, there are multiple confidence scores $T(c|d_i) > 0$ regarding different categories in C. We use the following formula to normalize them, and then we can get the probability $P(c_j|d_i)$ to reflect the extent that document d_i falls within category c_j:

$$P(c_j|d_i) = \frac{T(c_j|d_i)}{\sum_{c \in C} T(c|d_i)}. \tag{1}$$

In our next experiment, we set two categories, i.e., c_1 and c_2 for the document set D_1. For each document d_i, we can calculate the corresponding $P(c_j|d_i)$:

$$P(c_1|d_i) = \frac{T(c_1|d_i)}{\sum_{c \in C} T(c|d_i)}, \tag{2}$$

$$P(c_2|d_i) = \frac{T(c_2|d_i)}{\sum_{c \in C} T(c|d_i)}. \tag{3}$$

Then, all documents can be tagged with two probability values: $P(c_1|d_i)$ and $P(c_2|d_i)$.

3.3 Filtering of the Labeled Documents

Though the probability values that document d_i belongs to category c_j can be calculated, we cannot guarantee the tagging correctness because the initial document labeling might be biased by the category name and the extended features. Because our semi-supervised learning method only needs a few labeled data, what we need to do is just to select those labeled documents that are indeed labeled rightly.

After getting those initial labeled documents which are tagged with two probability values $P(c_1|d_i)$ and $P(c_2|d_i)$, our method ranks them by a descending order of the probability values for each category. For example, given category c_1 and document set D_1, our method only considers the probability value $P(c_1|d_i)$ of each document d_i, and then ranks all documents by the value of $P(c_1|d_i)$ in a descending order which we set it as $O(c_1)$. Similarly, we rank all documents by the value of $P(c_2|d_i)$ in a descending order which we set it as $O(c_2)$ for category c_2.

We set those labeled documents that are sorted into c_1 as positive samples and those labeled documents that are sorted into c_2 as negative samples. Firstly, we select top $m\%$ documents of the order $O(c_1)$ to get the positive samples, where the parameter m usually monotonously descends with the document number and it is set to be 10 in our method to guarantee the labeling quality; secondly, we rank these selected documents by the value of $P(c_2|d_i)$ in an ascending order; thirdly, we select the top $n\%$ documents as our final positive samples, where the parameter n usually is small to guarantee the labeling quality.

We can get the negative samples with the same steps.

3.4 Training Data and Test Data Construction

Until now, we have obtained the labeled documents (positive samples and negative samples) for each category.

Except for those two labeled samples, the rest of documents are all treated as unlabeled documents. They will be used to train the semi-supervised classifier together with the labeled samples. Our goal in this segment is to process all documents (both of labeled documents and unlabeled documents) and finally get the right format data that can be used to train the classifier directly.

Word Segmentation and POS Tagging Chinese word segmentation is a process that a sequence of Chinese characters can be divided into many separate words. POS tagging means that we indicate the correct part of speech (noun, verb, adjective and so on)

for each word in the word segmentation result. In our paper, we use ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System) to process the documents for word segmentation and POS tagging.

Stop-Words Removing Stop-words generally refers to those words that have no clear meanings and appear frequently in the documents. Usually, we filter out the stop-words in the natural language processing to improve the classification efficiency. The stop-words is placed on the stop-words list which usually comprises the resulting preposition, conjunction, auxiliary words, interpunction, pronoun, exclamation, modal words, onomatopoeic words, and so on.

Feature Selection Feature selection means a process that the words which are easy to distinguish in multiple categories are selected to construct the feature vector after word segmentation. It not only reduces the amount of calculation, but also maintains the classification accuracy. Since the words we select them as feature vector should be easy to distinguish in multiple categories, we use chi-square instead of TF-IDF which is usually used to distinguish in multiple documents to measure the importance of the feature words. We calculate the chi-square values of all words and rank them by a descending order. The top h words will be selected as the feature vector, where h is the length of the feature vector.

Term Weighting After feature selection, we select TF-IDF as the term weighting scheme to calculate the weight value of all feature words in each document d_i. Every document is represented as a numeric vector, i.e., $\boldsymbol{v}(d_i) = (n_{1i}, n_{2i}, \cdots, n_{ti})$, where each $n_{ji}\,(1 \leqslant j \leqslant t)$ corresponds to the TF-IDF value of each word in the document d_i.

Ranking and Scaling Since each TF-IDF value in the vector $\boldsymbol{v}(d_i)$ corresponds to a feature word in the document d_i, we rank those TF-IDF values by the order of the feature vector and then normalize the values to the value range $[0, 1]$ using the scaling formula $(x_{\mathrm{v}} - x_{\min})/(x_{\max} - x_{\min})$, where x_{v} corresponds to the TF-IDF value of each word in the vector $\boldsymbol{v}(d_i)$, $x_{\min}$ corresponds to the minimum TF-IDF value in the vector $\boldsymbol{v}(d_i)$, and $x_{\max}$ corresponds to the maximum TF-IDF value in the vector $\boldsymbol{v}(d_i)$.

Finally, we can obtain the training data after conducting the above five steps.

Documents in another document set D_2 are treated as test document, and we can finally obtain the test data using the above steps which are the same as the training data after being constructed.

3.5 Classifier Constructing

As the training and test data being constructed, our semi-supervised binary TC classifier will be built by two methods, i.e., TSVM and deterministic annealing (DA) algorithms.

SVM is specially designed for binary classification problems at the beginning. Since the dimensions of the text sample are usually huge enough and the performance of the text classification based on SVM is irrelevant to the dimensions, the SVM is very suitable for solving the problem of text classification. At the same time, there is no local minimum value problem by using SVM and the generalization ability of SVM is stronger. Of course, it is also because the kernel function is introduced. To make the experiment more persuasive, we introduce the supervised classifier SVM as a comparison.

3.5.1 *SVM*

SVM algorithm originated from the research of data classification problem is a way of implementing the statistic learning theory. Based on the Vapnik-Chervonenkis (VC) dimension and the structure risk minimization principle of statistical learning theory, SVM is designed for the case that the training sample set size is limited. Given a set of positive and negative datasets, its main idea is to separate those data by building a hyperplane classifier. The classifier not only correctly classifies the data but also guarantees the margin between two classes of data to be maximal.

SVM mainly has two advantages in TC task[7]. ① Since the theoretically derived default selection of parameters setting has been proved with the best effectiveness, there is no need to spend efforts in parameter tuning. ② Feature selection can be ignored for it is fairly robust to over-fitting and can scale up to high dimensionalities.

3.5.2 *TSVM*

The goal of traditional SVM is to ensure that the classifier has small error rate in the training set as far as possible by using the distribution of the training set to approximate the distribution of the test set. But in fact, what we really want is to make that the classifier has small error rate as far as possible in the test set; it is just the problem solved by TSVM[7]. TSVM aims at finding a low-density area of data to maximize margin over both labeled data and unlabeled data[1].

TSVM almost inherits all properties of SVMs. Joachims[7] proved that TSVM can substantially improve the performance of SVMs for TC by utilizing a mass of unlabeled data.

3.5.3 *Semi-Supervised SVM Based on DA*

Semi-supervised SVM based on DA algorithm is developed by Sindhwani and Keerthi[13] in 2006. DA is an established tool for combinatorial optimization which processes the problem from information theoretic principles[13]. The discrete variables in the optimization problem are associated to continuous probability variables, and a non-negative temperature parameter T is utilized to track the global optimum.

Since the TSVM loss function over the unlabeled samples can be non-convex, this makes the TSVM optimization procedure susceptible to local minimum issues

and then causes a loss in its performance in many situations. Sindhwani and Keerthi[14] presented a new method based on DA that can potentially overcome this bottleneck while also be good at large scale applications. This method generates a family of objective functions whose non-convexity is controlled by an annealing parameter, and the global minimizer is parametrically tracked in this family.

The DA method is reasonably slower but always gives the best accuracy. For one thing, the semi-supervised solutions never lag behind purely supervised solutions in terms of performance[14].

4 Experiments and Results

In this section, we demonstrate the effectiveness of the semi-supervised TC and introduce the data set, the evaluation metrics and the experimental results.

4.1 Experimental Settings

Two widely used datasets, i.e, Neteasy-classification-corpus and SogouC-UTF8 datasets, are adopted in our experiments.

Neteasy-classification-corpus is often used as Chinese text classification dataset. It contains a total of 24 000 documents which are divided into 6 categories: Auto, Culture, Economics, Medicine, Military and Sports. Each categories contains 4 000 documents. Our experiments choose four categories: Auto, Culture, Economics and Medicine. Then we select 1 600 documents from both Auto and Culture respectively as Dataset 1, and select 2 000 documents from both Economics and Medicine respectively as Dataset 2. Dataset 1 and Dataset 2 contain 7 200 documents in total. We both choose 50% documents of each category as the training set and remaining 50% documents as the test set.

SogouC-UTF8 dataset was collected by Sogou Labs in 2012. It includes 10 categories: Auto, Finance and Economics, Health, Sports, Tour, Education, Recruitment, Culture, War and Information Technology (IT). Each categories contains 8 000 documents. Our experiments also choose four categories: Tour, War, IT and Sports. We select 1 600 documents from both Tour and War respectively as Dataset 1, and select 2 000 documents from both IT and Sports respectively as Dataset 2. Dataset 1 and Dataset 2 contain 7 200 documents in total. Similarly, we both choose 50% documents of each category as the training set and remaining 50% documents as the test set.

4.2 Experimental Evaluation Criteria

Evaluation criteria about TC mainly focus on two aspects of the classifier. One is the accuracy of the classification result, and the other is the speed of the processing procedure in classification.

Generally speaking, there are some standard criteria that are commonly used in TC to measure the effectiveness of binary TC classifier, such as precision/recall rate, break-even point, F-measure, and precision/error rate. In the multiple classification problems, different class has different F1 measure. In order to evaluate the whole classification system, the results of multiple binary tasks are averaged as a single performance value. Two averaging functions are adopted, i.e., micro-averaging and macro-averaging.

We just use three standard criteria to measure the effectiveness of binary TC classifier: precision/recall rate, F-measure and precision/error rate. Precision represents the proportion that the documents classified into class c_j truly belong to c_j. Recall denotes the proportion that the documents belonging to class c_j are classified into c_j. In order to make a tradeoff between high precision and recall, F1 measure which takes into account of both of them is a widely adopted measure for binary TC tasks. Precision reflects the rate of the documents that are classified correctly in total dataset. Obviously, error rate stands for the proportion of documents that are classified mistakenly.

4.3 Experiments of Semi-Supervised TC

Category Name Extension We use HowNet, HIT-CIR Tongyici Cilin (extended) and Baidu search engine as the example semantic resources and use WordSimilarity.exe as word similarity computing tool to get the extended features of category name. HowNet and HIT-CIR Tongyici Cilin (extended) both do not provide explicit definition about synset. In semantic resources, words are grouped into sets of synonyms called synset. For HowNet, we can obtain the synset of each category name by finding all the words with identical definitions. For HIT-CIR Tongyici Cilin (extended), we can search the category name directly, and then get the synset. Finally, we put the two synsets together as the synsets of each category name.

Note that since these Chinese category names are simple, POS tagging, word segmentation and stopwords removing are ignored. So, there is no need to eliminate the ambiguity of category names understanding due to the fact that there is no polysemy in these category names.

To obtain the vector of each category, on the one hand, we collect the related words after word segmentation of the original document set and set those words as EF1. On the other hand, we collect the hyponyms and relevant words using HowNet, HIT-CIR Tongyici Cilin (extended) and Baidu search engine, and then rank them by their semantic similarities with corresponding category name's synset[1]. We delete those words that have little similarity with category name's synset if this category name has too many hyponyms and relevant words, and set this words set as EF2. We combine both of EF1 and EF2 as the final extended feature of the category name.

Document Labeling First of all, we obtain both vector $\boldsymbol{v}(d_i)$ and $\boldsymbol{v}(f_j)$ by VSM, and then we calculate

the cosine similarity between each $\boldsymbol{v}(d_i)$ and each $\boldsymbol{v}(f_j)$. Finally, all documents are classified into a category and ranked by the similarity value.

The top percentage of the labeled documents, denoted as p_{l}, is selected to evaluate the quality of document labeling. We draw the curves of the quality of document labeling by changing the percentage p_{l} from 1% to 50%. The horizontal axis indicates the top percentage of the initially labeled documents to be evaluated, and the vertical axis is the accuracy value. Figure 5 indicates that the quality of the document labeling for both two datasets decreases monotonously with p_{l}, and before $p_{\mathrm{l}} < 10\%$, the accuracy of document labeling achieves more than 95%.

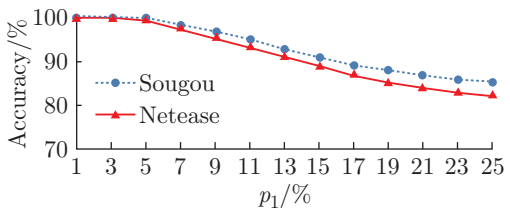

Fig. 5 Evaluation of document labeling on two Chinese datasets

Filtering of the Labeled Documents Our binary classifier only needs a few labeled training data. After calculating two probability values $P(c_1|d_i)$ and $P(c_2|d_i)$ for each document d_i, we select both positive samples and negative samples using the method we have introduced before. Finally, we select the top 25 labeled documents of each category as our labeled training data in semi-supervised TC experiments.

Training Data and Test Data Construction ICTCLAS is used to process the documents for word segmentation and POS tagging. The stop-word list comprises preposition, conjunction, auxiliary words, interpunction, pronoun, exclamation, modal words, onomatopoeic words, and so on. After removing the stop-words, we select 5 000 feature words in each document set as feature vector. Then, we select TF-IDF as the term weighting scheme. Finally, we obtain the training data and test data after ranking and scaling.

Classifier Constructing In our experiments, we select SVM, TSVM and DA as TC algorithms. SVM serves as the representation of supervised learning approach. TSVM and DA serve as the representation of semi-supervised learning approach. For all of them, the given training and test data from each of the four document sets are used for building and evaluating the classifiers.

For SVM, all training documents and test documents are labeled at the beginning, and we can obtain the labeled training data and labeled test data directly by conducting five steps: word segmentation and POS tagging, stop-words removing, feature selection, term weighting, and ranking and scaling .

For TSVM and DA, only test documents are labeled at the beginning. After category name extension, document labeling and filtering of the labeled documents, we can obtain part of correctly labeled training documents, and these parts of correctly labeled training documents together with the rest of unlabeled documents are treated as final training documents for semi-supervised learning. Then, we can obtain the labeled training data and labeled test data by conducting five steps: word segmentation and POS tagging, stop-words removing, feature selection, term weighting, and ranking and scaling.

Table 1 illustrates the TC results. We observe that, except for Sogou Dataset 1, semi-supervised algorithms TSVM and DA play better than the supervised algorithm SVM in terms of both F1 performance and classification accuracy in other three datasets. This fact proves the effectiveness of the semi-supervised algorithms in TC.

Table 1 Performance comparison on four Chinese datasets

Dataset	Algorithm	Wrong classification number	Evaluation criteria/%			Accuracy/%
			Precision	Recall	F1 measure	
Netease Dataset 1	SVM	211	79.4	99.4	88.3	86.8
	TSVM	59	96	96.6	96.3	96.3
	DA	54	96.2	97.1	96.7	96.6
Netease Dataset 2	SVM	154	87.1	99.6	92.9	92.3
	TSVM	84	94.7	97	95.8	95.8
	DA	67	96.0	97.4	96.7	96.7
Sogou Dataset 1	SVM	85	97.6	91.6	94.5	94.7
	TSVM	118	90.8	94.5	92.6	92.6
	DA	86	95.5	93.6	94.5	94.6
Sogou Dataset 2	SVM	137	88.1	99.8	93.6	93.2
	TSVM	70	94.5	98.8	96.6	96.5
	DA	73	94.1	98.9	96.4	96.4

In view of our advantage in category name extension and filtering of the labeled documents, we can select those labeled documents with high quality. Then, we use these labeled documents together with plenty of unlabeled documents to train our semi-supervised classifiers. Finally, the classifiers we obtain have high performance in TC. On the contrary, the performance of supervised algorithm such as SVM depends heavily on the quality of automatic tagging. However, there is no doubt that labeling all documents will introduce noise information.

The ratio of training data to test data, denoted as R, is $1:1$ in previous experiments. Here, R is set from $5:5$ to $9:1$. Then, we conduct several experiments on Netease dataset. We select 1 600 documents from both Economics and Medicine respectively as Dataset 3, and Dataset 3 contains 7 200 documents in total. Figure 6 shows the results. The vertical axis is the value of classification accuracy.

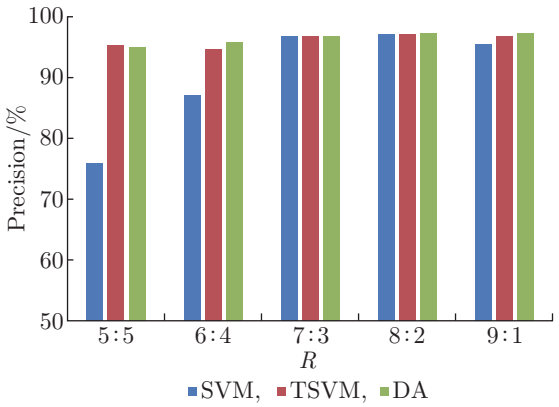

Fig. 6 Precision of TC based on different ratios of training data to test data

From the categorization results on Dataset 3, we can see that, when R is $5:5$ or $6:4$, the precision increases with the increase of the ratio at the beginning, and then reaches the peak value. After $R = 7:3$, the precision increases keep stable on the whole except that the precision of SVM decreases a little. This is because the noise information is introduced with the increase of the training data number. It proves again that our semi-supervised classifiers perform better in terms of robust aspect.

4.4 Results and Discussion

According to the analysis results of the data, the experimental results can show that the semi-supervised classifier based on automatic tagging is superior to the supervised classifier based on automatic tagging. The reason is because all training data must be automatically tagged if we want to get a supervised classifier. However, automatic tagging cannot ensure that all training data are tagged accurately. In fact, quite a number of training data can be classified into a wrong category. Finally, inaccuracy training data result in a supervised classifier with low accuracy.

As we know, automatic tagging can save a lot of time cost and labor cost. However, the accuracy of the automatic tagging is based on the similarity between the category name and the text content[15]. It is very likely to classify the document into a wrong category when the content of the document is not easy to distinguish. As a result, the supervised classifier based on the automatic labeling is hard to meet the accuracy requirements.

For semi-supervised learning, we can obtain a high accuracy classifier if we can guarantee that a few representative data are tagged correctly. This has been realized by automatic tagging in our experiments and we finally get a semi-supervised classifier with high accuracy. Therefore, our semi-supervised classifier is superior to the supervised classifier. If automatic tagging can guarantee that all data are tagged correctly, both semi-supervised classifier and supervised classifier may achieve a high classification accuracy. Obviously, it is very difficult for us at the moment.

Our results confirm two aspects. ① In TC, we can acquire a small number of labeled data with high quality by automatic tagging. ② Using few labeled data and a large number of unlabeled data as training data, we can obtain a semi-supervised text classifier with high categorization precision. Through the semi-supervised text classifier, we can classify plenty of unlabeled documents into a set of specific categories accurately and efficiently.

To build a supervised text classifier, we need a lot of labeled training documents. However, labeling all data by human is labor intensive and time consuming. Some scholars proposed automatic labeling documents. However, automatically labeling all data will bring noise into training data. Our algorithm proposes a new method to automatically tag part of labeled data with high quality, and then combines the labeled data and a lot of unlabeled data to build a semi-supervised classifier. Since the data we label have high quality, based on the plenty of unlabeled data, our semi-supervised algorithm can classify the unlabeled documents with high accuracy.

Our algorithm can solve the problem that automatically tagging all training documents will introduce noise information. At the same time, our semi-supervised classifier has very good robustness and can classify text documents with high accuracy.

Our experiments are based on the predecessors' research. Some scholars proposed that supervised learning is the most common method in automatic TC[7,16]. To avoid manual labeling, Sebastiani[6] and Shang et al.[17] proposed a semi-supervised learning approach. However, this study is still based on manual labeling training data. FACT is based on automatically tagging

all data and uses all labeled data to build a supervised text classifier. Our experiments are also based on the idea of automatic document labeling to improve the classification accuracy, but the problem of noise introducing is solved.

In classification building step, we find that the supervised algorithm SVM plays better than the semi-supervised algorithms TSVM and DA in terms of F1 performance and classification accuracy in Sogou Dataset 1. This can explain that supervised algorithm such as SVM can achieve high classification accuracy as long as the labeled training data have high quality[18-19]. So, if the original document set is small and compact, and all documents are easy to distinguish, supervised classifier based on automatic document labeling can be a good selection[20-21].

There are some defects in our experiments. To solve the problem of the experimental time, we do not automatically label all documents for the supervised classifier SVM and directly use all already labeled documents as its training data. Experiment in the future should automatically label all documents for the supervised classifier SVM. Since the limitation of computer configuration, we cannot use too many documents as training data to build the classifier. Therefore, the experimental results have some limitations.

Some questions still need to be studied in the future. For example, when document set is huge enough, feature selection may affect training time and classifier classification accuracy[22]. So, how to set the number of feature words plays an important role.

Semi-supervised classifier based on automatic document labeling can classify text documents with high efficiency in most cases[23-24], especially in the case of a lot of unlabeled documents.

5 Conclusion

Applying TC in large-scale document set plays an important role in the information age for it can help people to find useful acknowledge quickly. However, labeling all documents to construct training documents for classifier can be hard and will introduce noise information. Besides, it is difficult for text classifier to classify a huge number of documents into right categories with high accuracy. This paper provides a new method for TC to address these problems. It is based on using the labeled data with high quality and plenty of unlabeled data to construct the supervised classifier. We use the semantics of the category names hidden in original document set and many semantic resources to label the documents and build the training data. Since we make use of the labeled data and plenty of unlabeled data, our text classifier based on semi-supervised learning way can achieve a high classification accuracy. The experiments in this paper have demonstrated the effectiveness of our proposed approach. Its performance is better especially when the given category name can represent the topics of the documents clearly, and the content of documents in one category is easy to distinguish with the content of documents in another category.

Our method to obtain the extended features of the category name is very useful and it is proved by experiments. We first extract the words related with category name from the original documents and some lexical databases, then refine those words again and finally use them as the extended features of the category name. This makes the extended feature very representative, and it enables us to label the documents with high accuracy. In addition, we apply this approach to other language document labeling.

In future work, we apply our methods to multi-category text classification based on a bigger document set and use other machine learning algorithms like Naïve Bayes to implement a discriminative classifier.

References

[1] LI J Q, ZHAO Y, LIU B. Exploiting semantic resources for large scale text categorization [J]. *Journal of Intelligent Information Systems*, 2012, **39**(3): 763-788.

[2] MIYATO T, DAI A M, GOODFELLOW I. Virtual adversarial training for semi-supervised text classification [EB/OL]. (2016-07-22). https://arxiv.org/abs/1605.07725v1.

[3] YIN C Y, XIANG J, ZHANG H, et al. A new SVM method for short text classification based on semi-supervised learning [C]//*2015 4th International Conference on Advanced Information Technology and Sensor Application*. Dubai, UAE: IEEE, 2015: 100-103.

[4] JOHNSON R, ZHANG T. Semi-supervised convolutional neural networks for text categorization via region embedding [J]. *Advances in Neural Information Processing Systems*, 2015, **28**: 919-927.

[5] JOHNSON R, ZHANG T. Supervised and semi-supervised text categorization using LSTM for region embeddings [C]//*Proceedings of the 33rd International Conference on Machine Learning*. New York, USA: JMLR W&CP, 2016: 1-9.

[6] SEBASTIANI F. Machine learning in automated text categorization [J]. *ACM Computing Surveys*, 2002, **34**(1): 1-47.

[7] JOACHIMS T. Transductive inference for text classification using support vector machines [C]//*Proceedings of the 16th International Conference on Machine Learning*. Bled, Slovenia: [s.n.], 1999: 200-209.

[8] SIOLAS G, D'ALCHÉ-BUC F. Support vector machines based on a semantic kernel for text categorization [C]//*Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neuralnetworks*. Washington, USA: IEEE, 2000: 205-209.

[9] BASILI R, CAMMISA M, MOSCHITTI A. Effective use of Wordnet semantics via kernel-based learning [C]//*Proceedings of the 9th Conference on*

Computational Natural Language Learning. Ann Arbor, USA: Association for Computational Linguistics, 2005: 1-8.

[10] GABRILOVICH E, MARKOVITCH S. Feature generation for text categorization using world knowledge [C]//*International Joint Conference on Artificial Intelligence*. [s.l.]: Morgan Kaufmann Publishers Inc, 2005: 1048-1053.

[11] WANG P, DOMENICONI C. Building semantic kernels for text classification using wikipedia [C]//*ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, USA: ACM, 2008: 713-721.

[12] CHAPELLE O, SCHÖLKOPF B, ZIEN A. Semi-supervised learning [M]. London, England: MIT Press, 2006.

[13] SINDHWANI V, KEERTHI S S. Large scale semi-supervised linear SVMs [C]//*International ACM SIGIR Conference on Research and Development in Information Retrieval*. Washington, USA: ACM, 2006: 477-484.

[14] SINDHWANI V, KEERTHI S S. Newton methods for fast solution of semi-supervised linear SVMs [EB/OL]. (2016-07-22). http: //citeseerx.ist.psu.edu/viewdoc/download.

[15] LI C H, YANG J C, PARK S C. Text categorization algorithms using semantic approaches, corpus-based thesaurus and WordNet [J]. *Expert Systems with Applications*, 2012, **39**: 765-772.

[16] FOX-ROBERTS P, ROSTEN E. Unbiased generative semi-supervised learning [J]. *Journal of Machine Learning Research*, 2014, **15**: 367-443.

[17] SHANG F H, JIAO L C, LIU Y Y, et al. Semi-supervised learning with nuclear norm regularization [J]. *Pattern Recognization*, 2013, **46**(8): 2323-2336.

[18] WANG J, JEBARA T, CHANG S F. Semi-supervised learning using greedy max-cut [J]. *Journal of Machine Learning Research*, 2013, **14**: 729-758.

[19] CHENG S, SHI Y H, QIN Q D. Particle swarm optimization based semi-supervised learning on chinese text categorization [C]//*Proceedings of the 2012 IEEE Congress on Evolutionary Computation*. Brisbane, Australia: IEEE, 2012: 1-8.

[20] LENG Y, XU X Y, QI G H. Combining active learning and semi-supervised learning to construct SVM classifier [J]. *Knowledge-Based Systems*, 2013, **44**(1): 121-131.

[21] LI J Q, LIU C C, LIU B, et al. Diversity-aware retrieval of medical records [J]. *Compuer in Industries*, 2015, **69**(1): 81-91.

[22] YANG J M, LIU Y N, ZHU X D, et al. A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization [J]. *Information Processing and Management*, 2012, **48**(4): 741-754.

[23] BREVE F, ZHAO L, QUILES M, et al. Particle competition and cooperation in networks for semi-supervised learning [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2011, **24**(9): 1686-1698.

[24] LI J Q, WANG F. Semi-supervised learning via mean field methods [J]. *Neurocomputing*, 2016, **177**: 385-393.