Information Sciences 181 (2011) 1138–1152

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Ensemble of feature sets and classification algorithms for sentiment classification

Rui Xia [a,*], Chengqing Zong [a], Shoushan Li [b]

[a] National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China
[b] Department of Computer Science and Technology, Soochow University, Suzhou 215006, China

ARTICLE INFO

Article history:
Received 28 December 2009
Received in revised form 9 October 2010
Accepted 18 November 2010

Keywords:
Sentiment classification
Text classification
Ensemble learning
Classifier combination
Comparative study

ABSTRACT

In this paper, we make a comparative study of the effectiveness of ensemble technique for sentiment classification. The ensemble framework is applied to sentiment classification tasks, with the aim of efficiently integrating different feature sets and classification algorithms to synthesize a more accurate classification procedure. First, two types of feature sets are designed for sentiment classification, namely the part-of-speech based feature sets and the word-relation based feature sets. Second, three well-known text classification algorithms, namely naïve Bayes, maximum entropy and support vector machines, are employed as base-classifiers for each of the feature sets. Third, three types of ensemble methods, namely the fixed combination, weighted combination and meta-classifier combination, are evaluated for three ensemble strategies. A wide range of comparative experiments are conducted on five widely-used datasets in sentiment classification. Finally, some in-depth discussion is presented and conclusions are drawn about the effectiveness of ensemble technique for sentiment classification.

1. Introduction

Text classification has been one of the key tools to automatically handle and organize text information for decades. In recent years, with more and more subjective information appearing on the internet, sentiment classification [21,29], as a special case of text classification for subjective texts, is becoming a hotspot in many research fields, including natural language processing (NLP), data mining (DM) and information retrieval (IR).

The dominant techniques in sentiment classification generally follow traditional topical text classification approaches, where a document is regarded as a bag of words (BOW), mapped into a feature vector, and then classified by machine learning techniques such as naïve Bayes (NB) [19], maximum entropy (ME) [27], or support vector machines (SVM) [14]. The effectiveness of machine learning techniques when applied to sentiment classification tasks is evaluated in the pioneering research by Pang et al. [30]. The experimental results on the movie-review dataset produced via NB, ME, and SVM are substantially better than those results obtained through human generated baselines. But their performance is not as remarkable as when they are used in topical text classification. The main reason may be that traditional BOW does not capture word order information, syntactic structures and semantic relationships between words, which are essential attributes for sentiment analysis. Therefore, various kinds of feature sets, such as part-of-speech (POS) based features [12], higher-order *n*-grams [5,15,30], word pairs and dependency relations [5,10,15,36], have been exploited to improve sentiment classification performance.

Previous work, however, mostly focuses on joint features while ignoring an efficient integration of different types of features to enhance the sentiment classification performance. On one hand, among different classification algorithms, which

* Corresponding author.
E-mail addresses: rxia@nlpr.ia.ac.cn (R. Xia), cqzong@nlpr.ia.ac.cn (C. Zong), shoushan.li@gmail.com (S. Li).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.11.023

one performs consistently better than the others remains a matter of some debate. On the other hand, different types of features have distinct distributions, and therefore would probably vary in performance between different machine learning algorithms. For example, it is reported in [30] that on the movie dataset, SVM performs the best, ME maintains an average, and NB tends to do the worst on unigram features; while the outcome is the reversed for bigrams. This is possibly due to the relevance between bigrams being lower than between unigrams. Moreover, the performance of classification algorithms is also domain-dependent. For instance, subsequent literature [4] shows that, using the same unigram features, NB performs better than SVM on datasets other than movie reviews.

We therefore intuitively seek to integrate different types of features and classification algorithms in an efficient way in order to overcome their individual drawbacks and benefit from each other's merits, and finally enhance the sentiment classification performance.

The ensemble technique, which combines the outputs of several base classification models to form an integrated output, has become an effective classification method for many domains [13,17]. In topical text classification, several researchers have achieved improvements in classification accuracy via the ensemble technique. In the early work [18], a combination of different classification algorithms (*k*-NN, Relevance feedback and Bayesian classifier) produces better results than any single type of classifier. Literature [6] makes a comparison of several ensemble methods for text categorization, which investigates six homogeneous ensemble methods (*k*-fold partitioning, bagging, boost, biased *k*-partitioning, biased *k*-fold partition, and biased clustering). In the field of sentiment classification, however, related works are very rare and no extensive evaluation has been carried out. Literature [38] proposes four ensemble algorithms (bagging, boosting, random subspace, and bagging random subspaces) using SVM as the base classifier and reports that ensemble of random subspaces can increase classification accuracy and the bagging subspaces model has the highest accuracies. In [20], different classifiers are generated through training with different sets of features, then component classifiers are selected and combined using several fixed combination rules. Experimental results show that all of the combination approaches can outperform individual classifiers and the sum rule achieves the best performance.

In this paper, we aim to make an intensive study of the effectiveness of ensemble techniques for sentiment classification tasks. Rather than an ensemble of different data re-sampling methods (e.g. bagging and boosting), we focus on ensemble of feature sets and classification algorithms. We design two schemes of feature sets that are particular to sentiment analysis: one is part-of-speech (POS) based and the other is word-relation (WR) based. For each scheme, we utilize NB, ME, and SVM as the base-classifiers to predict classification scores. In the ensemble stage, we apply three types of ensemble method (fixed combination, weighted combination, and meta-classifier combination) with three ensemble strategies (ensemble of feature sets, ensemble of classification algorithms, and ensemble of both feature sets and classification algorithms). A wide range of comparative experiments are conducted on five datasets widely used in sentiment classification. We seek answers based on empirical evidence to the following questions:

(1) What are the strengths and weaknesses of existing feature sets and classification algorithms when applied to the task of sentiment classification?
(2) Can the performance of a sentiment classification system benefit from the ensemble technique? To what extent can each of the three ensemble strategies improve the system performance?
(3) Among various combination methods, which one can be selected as the winner across all settings and datasets? Are there any guidelines to help choose the best from these methods?

The remainder of this paper is organized as follows. Sections 2 and 3 review traditional sentiment feature engineering and classification algorithms, respectively. In Section 4, we describe two schemes of feature sets and present the ensemble framework for sentiment classification. Experimental results are presented and analyzed in Section 5. In Section 6, we make in-depth discussion and answer the above three questions. Section 7 draws conclusions and outlines directions for future work.

2. Feature engineering

The text representation method dominating the literature is known as the BOW framework. In this framework, a document is considered as a bag of words and represented by a feature vector containing all the words appearing in the corpus. Although BOW is simple and quite efficient in text classification, a great deal of the information from the original document is discarded, word order is disrupted, and syntactic structures are broken. Therefore, sophisticated feature extraction methods with a deeper understanding of the documents are required for sentiment classification tasks. Instead of using a bag of words (unigrams), alternative ways to represent text, including POS based features, higher-order *n*-grams, and word dependency relations are presented in the literature.

2.1. Part-of-speech information

POS information is supposed to be a significant indicator of sentiment expression. The work on subjectivity detection [12] reveals a high correlation between the presence of adjectives and sentence subjectivity, yet this should not be taken to mean that other POS tags do not contribute to subjective expression. Indeed, in the study [30], the experimental results show that using only adjectives as features actually results in much worse performance than using the same number of most frequent unigrams. Other researchers point out that adjectives and adverbs are better than adjectives alone and certain verbs and nouns are also strong indicators of sentiment [1,35].

2.2. Word relation features

With the attempt to capture the word relation information behind the text, word relation (WR) features, such as higher-order *n*-grams and word dependency relations, have been widely employed in text representation [5,10,15,23,36,40].

Higher order *n*-grams are features that have gained ground in the field of NLP. Bigrams and trigrams are widely used as features in text classification and sentiment classification for their capacity to encode the word order information [5,30]. There have also been attempts at incorporating syntactic relations between words [5,10]. As a structured representation, a dependency parsing tree expresses the dependency relation between words in the sentence by child-parent relations of nodes. The dependency parsing tree of the sentence "*I definitely recommend this film*." is demonstrated in Fig. 1. A straightforward method for extracting dependency relations is simply using pairs of dependent words (for example, "*recommend film*") as features. It is believed that these features, to some extent, encode word order information and long-range dependency relations, and therefore are helpful to sentiment classification.

However, in most of the literature, the performance of individual WR features is poor, even inferior to the traditional unigrams. For example, it is reported that unigrams outperform bigrams [30], and individual dependency relation features are also shown inferior to unigrams [10]. For this reason, WR are commonly used as extra features, in addition to unigrams, to encode more word order and word relation information. Even so, the performance of joint features is still far from satisfactory [5,10,15].

2.3. Feature weighting

In topical text classification, term frequency – inverse document frequency (TF-IDF) weighting has gained great success, but in sentiment classification, Pang et al. [30] points out while a topic is more likely to be emphasized by frequent occurrences of certain keywords, overall sentiment may not usually be highlighted through repeated use of the same terms. In their experiment, better performance is obtained using presence rather than frequency, that is, binary-valued feature vectors in which the entries merely indicate whether a term occurs or not formed a more effective basis for review polarity classification. In the following literature, presence has been confirmed as the most effective feature weighting method and it is therefore most-frequently used in sentiment classification.

3. Classification algorithms

Machine learning techniques like naïve Bayes (NB), maximum entropy (ME), and support vector machines (SVM) have achieved great success in text categorization. For sentiment classification task, the feasibility of these classifiers is proved by Pang et al. [30]. In their experiment on the movie review dataset, the results of NB, ME, and SVM are substantially better than the results obtained through human generated baselines.

3.1. Näive Bayes

In the BOW framework, a document **x** is represented by $[w_1,\ldots,w_m]$ where w_k denotes the *k*th word appearing in the document. Naïve Bayes assumes that words are mutually independent. Under this assumption, the conditional probabilities can be simplified as

$$P(\mathbf{x}|y_j) = P([w_1,\ldots,w_m]|y_j) \approx \prod_{k=1}^{m} P(w_k|y_j). \tag{1}$$

The naïve Bayes decision can be described as

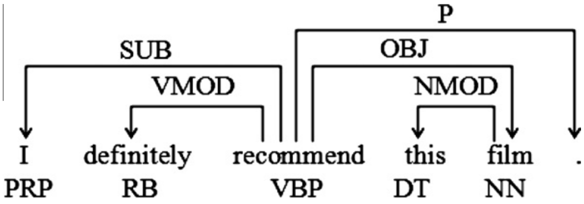

Fig. 1. A demonstration of dependency parsing tree.

$$\omega^* = \arg\max_{j=1,\ldots,c} \prod_{k=1}^{m} P(w_k|y_j)P(y_j). \tag{2}$$

The probabilities $P(w_k|y_j)$ and $P(y_j)$ can simply be estimated by maximum likelihood. Moreover, Laplace smoothing is necessary in order to prevent infrequently occurring words from being zero probabilities.

3.2. Maximum entropy

The maximum entropy (ME) classifier estimates the conditional distribution of the class label y given a document $\mathbf{x}$ using the form of an exponential family with one weight for each constraint:

$$P_\lambda(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\sum_i \lambda_i f_i(\mathbf{x}, y)\right\}, \tag{3}$$

where $Z(\mathbf{x})$ is a normalization factor, $f_i(\mathbf{x},y)$ is a feature function defined as

$$f_i(\mathbf{x}, y) = \begin{cases} 1, & \mathbf{x} = \mathbf{x}_i \text{ and } y = y_i, \\ 0, & \text{other}, \end{cases} \tag{4}$$

and λ_i is the weight coefficient.

The model with maximum entropy is the one in the parametric family $P_\lambda(y|\mathbf{x})$ that maximizes the likelihood. Numerical methods such as iterative scaling and quasi-Newton optimization are usually employed to solve the optimization problem.

3.3. Support vector machines

As a discriminative model, SVM uses $g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$ as the discriminant function, where $\mathbf{w}$ is the weights vector, b is the bias, and $\phi(\cdot)$ denotes nonlinear mapping from input space to high-dimensional feature space. The parameters $\mathbf{w}$ and b are learned automatically on the training dataset following the principle of maximized margin by

$$\begin{aligned} &\min \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i \\ &\text{s.t.} \quad \begin{cases} y_i g(\mathbf{x}_i) \geqslant 1 - \xi_i, \\ \xi_i \geqslant 0, \quad i = 1, \ldots, N, \end{cases} \end{aligned} \tag{5}$$

where ξ_i denotes the slack variables and C is the penalty coefficient. Instead of solving this problem directly, it is converted to an equivalent quadratic optimization problem by Lagrange multipliers.

The training sample $(\tilde{\mathbf{x}}_i, y_i)$ is called a support vector when satisfying the Lagrange multiplier $\alpha_i > 0$. By introducing kernel function, the discriminant function can be represented as

$$g(\mathbf{x}) = \sum_{i=1}^{\widetilde{N}} \alpha_i y_i K(\tilde{\mathbf{x}}_i, \mathbf{x}). \tag{6}$$

Due to the dimension of feature space is quite large in text classification tasks, the classification problem is always linearly separable [39], therefore linear kernel is commonly used.

4. The ensemble model

4.1. Model formulation

The pursuit of ensemble has been motivated by the intuition that an appropriate integration of different participants might leverage distinct strengths. In multiple classifier combination, the scores generated by contributing classifiers on component feature sets are taken as inputs to the combination function. Assuming that we combine D component models for a C-class classification task, the ensemble model can be formulated as

$$O_j(\mathbf{x}) = F\begin{pmatrix} o_{11}(\mathbf{x}_1), \ldots o_{1j}(\mathbf{x}_1), \ldots, o_{1C}(\mathbf{x}_1) \\ \cdots \\ o_{D1}(\mathbf{x}_D), \ldots, o_{Dj}(\mathbf{x}_D), \ldots, o_{DC}(\mathbf{x}_D) \end{pmatrix}, \tag{7}$$

where $o_{kj}(\mathbf{x}_k)$ is the predicted score of classification model k for class j and $F(\cdot)$ indicates the combining function. The ensemble components can be generated by different classification algorithms on different feature sets.

Table 1
The categorization of POS-based feature sets.

Component feature sets	Included POSs
POS-1	JJ, JJS, JJR, RB, RBR, RBS
POS-2	VB, VBZ, VBD, VBN, VBP, VBG, MD
POS-3	NN, NNS

Here the POS tags use the style of Penn Treebank.[a] JJ, RB, VB and NN, respectively, denote adjectives, adverbs, verbs and nouns. POS tags at different tenses are also included. For example, JJR denotes comparative adjective, JJS denotes superlative adjective, VBZ denotes the 3SG form of verb, NNS denotes the plural noun, etc.

[a] http://www.cis.upenn.edu/~treebank/.

4.2. Two schemes of component feature sets

We design two schemes of feature sets for sentiment classification. The first is called "POS-based feature sets" and the second is "WR-based feature set".

4.2.1. POS-based feature sets

According to the content of sentiment information, unigrams are divided into three groups in the first scheme: POS-1 comprises adjectives and adverbs, POS-2 includes verbs, and nouns are categorized into POS-3. The detailed categorization of POS-based feature sets is shown in Table 1. POS-1 is considered to be the most relevant feature subset to sentiment classification, POS-2 is in second place, and POS-3 ranks the third. The other types of POS tags are abandoned as they are supposed to carry a significant amount of noise rather than useful information.

4.2.2. WR-based feature sets

In WR-based feature sets, in addition to unigrams, bigrams and dependency parsing pairs are also employed as extra features. They represent a single word, word order information, and the long-distance word dependencies, respectively. Taking the sentence in Fig. 1 for example, the WR-based feature sets are illustrated in Table 2. Since the three kinds of features describe different relationships between words, we believe that they all contain some particular information to sentiment analysis.

4.3. Component classification algorithms and output normalization

NB, ME, and SVM are used as component classification models in the ensemble system. OpenPR-NB[1] is used as the naïve Bayes classifier in our experiments. OpenPR-NB is a C++ implementation of naive Bayes Classifier based on the multinomial event model [24] and Laplace smoothing. The tool of LIBSVM[2] is chosen as the SVM classifier in our experiments. The parameter of kernel function is set to be linear kernel and the penalty parameter is set to one. The Maximum Entropy Toolkit[3] is chosen as our ME classifier, where L-BFGS is chosen for optimization. We use presence feature weighting in all of the three classifiers.

When evaluating the degree of decision confidence of different classifiers, the outputs should be transformed to an uniform measure. Generally, the score level output (e.g. the posterior probability, or joint probability) contains richer information than the class labels or ranks, and is thus preferred [22]. In NB model, in order to prevent overflow in computing the products of conditional probabilities, the logarithmic form is usually taken:

$$\omega^* = \arg\max_{j=1,\ldots,C} \log(P(\mathbf{x}, y=j)) = \arg\max_{j=1,\ldots,C} \sum_{k=1}^{m} \log(P(w_k|y=j)P(y=j)). \tag{8}$$

Using o_j to denote the output belonging to class $j: o_j = \log p(\mathbf{x}|y=j)P(y=j)$, the posterior probabilities can be obtained by

$$P(y=j|\mathbf{x}) = \frac{P(\mathbf{x}, y=j)}{P(\mathbf{x})} = \frac{\exp\{o_j\}}{\sum_{\bar{j}=1}^{c}\exp\{o_{\bar{j}}\}} = \frac{1}{1+\sum_{\bar{j}\neq j}\exp\{o_{\bar{j}}-o_j\}}, \tag{9}$$

which has the same form as ME. Indeed, in the terminology of [26,37], NB and ME form a generative–discriminative pair. The outputs of SVM also need to be re-functioned to approximate the posterior probabilities. In our experiments, we apply the Platt's probabilistic outputs [31] which has been implemented in LIBSVM.

[1] http://www.openpr.org.cn
[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm
[3] http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

Table 2
A demonstration of WR-based feature sets.

WR type	WR-based feature sets
Unigrams	*i*, *definitely*, *recommend*, *this*, *film*
Bigrams	*i_definitely*, *definitely_recommend*, *recommend_this*, *this_film*, *film_.*
Dependencies	*i_recommend*, *definitely_recommend*, *this_film*, *film_recommend*, *._recommend*

Unigrams, Bigrams and Dependencies, respectively, denote unigrams, bigrams and dependency relation features.

4.4. Ensemble methods

The ensemble methods are usually categorized into two types, i.e., fixed rules and trained methods [22]. Fixed rules combine the individual outputs in a fixed manner, such as the sum rule and voting rule; whereas trained methods, including the weighted combination and meta-classifier, combine outputs via training on a validation dataset.

4.4.1. Fixed rules

The common rules for fixed combination include voting rule, sum rule, max rule, and product rule [17]. The voting rule counts the predictions of component classifiers and then assigns test sample $\mathbf{x}$ to class i with the most component predictions

$$O_j = \sum_{k=1}^{D} I\left(\arg\max_j (o_{kj}) = j\right), \tag{10}$$

where $I(\cdot)$ means the indicator function.

The sum rule combines component outputs by

$$O_j = \sum_{k=1}^{D} o_{kj}, \tag{11}$$

which is equivalent to the averaging of outputs over classifiers (average rule). It was reported in [17] that the sum rule outperforms other rules because of its resilience to estimation errors.

4.4.2. Meta-classifier

For combination using a meta-classifier, the outputs for all the class labels of component classifiers are viewed as new features for meta-learning. Among the various kinds of classification models, linear regression is most recommended.

We map the $F = D * C$ outputs of base classifiers into a feature vector and represent it by

$$\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_l, \ldots, \hat{x}_F] = [o_{11}, \ldots, o_{1C}, \ldots, o_{k1}, \ldots, o_{kj}, \ldots, o_{kC}, \ldots, o_{D1}, \ldots, o_{DC}], \tag{12}$$

and then the linear regression model can be formulated as

$$O_j = g(\hat{\mathbf{x}}) = \mathbf{w}_j^{\mathrm{T}} \hat{\mathbf{x}} + b_j = \sum_{l=1}^{F} w_{lj} \hat{x}_l + b_j, \tag{13}$$

where $\mathbf{w}_j$ is a F-dimensional weight vector for class j, and b_j is the bias.

The weights and biases can be adjusted by optimizing the cost function J of certain criteria: typically, least mean square (LMS) [3], cross entropy (CE) [11], and minimal classification error (MCE) [16].

Letting $\hat{\mathbf{x}}_i$ and y_i, respectively, denote the ith meta-sample and its class label, and the cost function of LMS and CE can be defined as

$$J_{lms} = \frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{C} \left[I(y_i = j) - f_j(\hat{\mathbf{x}}_i)\right]^2, \tag{14}$$

$$J_{ce} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} \left\{I(y_i = j) \log f_j(\hat{\mathbf{x}}_i) + I(y_i \neq j) \log\left(1 - f_j(\hat{\mathbf{x}}_i)\right)\right\}, \tag{15}$$

where a sigmoid function $\delta(\cdot)$ is applied to Eq. (13) to make the cost function differentiable:

$$f_j(\hat{\mathbf{x}}) = \delta\left(\mathbf{w}_j^{\mathrm{T}} \hat{\mathbf{x}} + b\right) = \frac{1}{1 + \exp\left\{-\left(\mathbf{w}_j^{\mathrm{T}} \hat{\mathbf{x}} + b\right)\right\}}. \tag{16}$$

The MCE criterion proposed in [16] is supposed to be more relevant to the classification error than LMS and CE. In their approach, the misclassification measure of a pattern is defined by

$$d_j(\hat{\mathbf{x}}_i) = -g_{y_i}(\hat{\mathbf{x}}_i) + \log\left[\frac{1}{C-1}\sum_{h\neq j}\exp\left(\eta g_h(\hat{\mathbf{x}}_i)\right)\right]^{1/\eta}. \tag{17}$$

When $\eta \to +\infty$, the misclassification measure is simplified as $d_j(\hat{\mathbf{x}}_i) = -g_{y_i}(\hat{\mathbf{x}}_i) + \max_{h\neq j} g_k(\hat{\mathbf{x}}_i)$ and the MCE cost function is then given by

$$J_{mce} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} I(y_i = j)\delta\left(d_j(\hat{\mathbf{x}}_i) + \alpha\right). \tag{18}$$

We also consider the single-layer perceptron model for meta-learning. Similar to the criteria for linear regression, the perceptron criterion function in multi-class case is given by

$$J_p = \frac{1}{N}\sum_{i=1}^{N}\left[\max_{j=1,\ldots,C} g_j(\hat{\mathbf{x}}_i) - g_{y_i}(\hat{\mathbf{x}}_i)\right]. \tag{19}$$

For all of the above criteria, we use stochastic gradient descent (SGD) for optimization. Compared to standard gradient descent, SGD is much faster and more efficient, especially for large datasets. SGD uses approximate gradients estimated from subsets of the training data and updates the parameters in an online manner:

$$w_{mn}(k+1) = w_{mn}(k) - \eta(k)\frac{\partial J}{\partial w_{mn}}. \tag{20}$$

When using SGD for optimization, the convergence of perceptron is very fast, but not especially robust; so we employ averaged perceptron – a variation of perceptron that averages weights of all iterations [8] – to improve the generalization performance. In the remainder of this paper, Averaged perceptron will be called AveP for short.

Besides linear regression, some other classification algorithms are also adopted in our experiments, such as linear SVM and logistic regression. Linear SVM also uses formula (13) as the discriminant function, but the weights are trained based on the principle that maximizes the margin between two classes. Logistic regression (Logit) directly estimates the posteriori probabilities and trains the weights on the principle of maximum likelihood.

4.4.3. Weighted combination

In meta-classifier combination, the final score for class j is related to not only the outputs for the corresponding class of base-classifiers, but also outputs for the other class labels (see formula (13)). But in weighted combination, each constituent classifier has exactly one weight for each class or for sharing for all classes. If we use formula (12) to present the feature vector, the weighted combination procedure could be expressed by

$$O_j = \sum_{k=1}^{D} w_k o_{kj} = h(\hat{\mathbf{x}}) = \sum_{k=1}^{D} w_k \hat{\mathbf{x}}_{k\times D+j}. \tag{21}$$

We attempt to find the best weights by minimizing the cost functions, which is similar to the training of linear regression. We still use SGD to optimize the perceptron, CE, LMS and MCE criteria. However, it should be noted that the derivations of gradients between the two models are slightly different.

5. Experimental study

In order to fully answer the questions raised in the introduction, we conduct a range of comparative experiments on five widely-used datasets. According to two schemes of component feature sets, the experiments are divided into two parts: POS-based and WR-based ensemble. In each of the experiments, the results of individual classifier and three ensemble strategies are reported, respectively.

5.1. Experimental settings

5.1.1. Datasets

We use five document-level datasets widely used in the field of sentiment classification. The Cornell movie-review corpora,[4] introduced in [28], consists of four collections of movie-review documents labeled with respect to their overall sentiment polarity (positive or negative) or subjective rating (e.g. two and a half stars) and sentences labeled with respect to their subjectivity status (subjective or objective) or polarity. We conduct our experiments on the document-level polarity dataset v2.0 that contains 1000 positive and 1000 negative processed reviews. The Multi-Domain Sentiment Dataset,[5] introduced in [2], contains

[4] http://www.cs.cornell.edu/people/pabo/movie-review-data/.

[5] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/.

Table 3
Accuracies (%) of individual classification algorithms on individual POS-based feature sets.

Dataset	POS-1			POS-2			POS-3			Joint POS		
	NB	ME	SVM	NB	ME	SVM	NB	ME	SVM	NB	ME	SVM
Movie	83.60	80.75	79.05	73.70	72.35	70.10	77.15	77.20	75.70	82.70	86.10	84.85
Book	75.10	69.75	71.85	70.30	67.50	67.15	65.10	64.00	64.65	76.70	78.10	75.60
DVD	76.15	72.75	73.00	68.95	66.85	68.20	68.80	67.80	66.65	78.85	78.95	77.60
Elec	78.55	72.65	75.45	73.80	67.65	69.90	69.15	67.05	66.35	81.75	80.35	78.30
Kitchen	80.10	74.70	78.45	73.85	69.35	70.40	70.05	67.50	66.45	82.40	82.50	82.10
Average	78.70	74.12	75.56	72.12	68.74	69.15	70.05	68.71	67.96	80.48	81.20	79.69

product reviews taken from Amazon.com from four product types (domains) – Book, DVD, Electronics and Kitchen. Each of these contains 1000 positive and 1000 negative reviews. All four domains are used in our experiments.

5.1.2. Pre-processing

To get the POS features and dependency relations, pre-processing steps such as word tokenization, POS tagging, and dependency parsing should be taken. We use the NLTK toolkit[6] for word tokenization. MXPOST[7] is chosen as the POS tagger in our experiments. MXPOST is a POS tagger based on maximum entropy which was introduced in [33]. To extract the dependency parsing tree, we chose the tool MSTParser[8] – a graph-based dependency parser described in [25]. Training is carried out on the WSJ part of the Penn Tree Corpus.

5.1.3. Implementation

Each dataset is evenly divided into 5 folds and all the experimental results are obtained with a 5-fold cross validation. In each loop of cross validation, documents from 4 folds are used as training data and the remaining fold is used as test data. Performance reported in all of the following tables is in terms of the average classification accuracy.

To produce the training samples for meta-learning, the stacking method [7] is employed. Taking the Movie dataset for example, in each loop of the 5-fold cross validation, the probabilistic outputs of the test fold are considered as test samples for meta-leaning; and an inner 4-fold leave-one-out procedure is applied to the training data, where samples in each of the four fold are trained on the remaining three folds (which can be considered as the validation set) to obtain the probabilistic outputs which serve as training samples for meta-learning.

5.2. Experiment I – POS-based ensemble

In this section, we report the experimental results of POS-based ensemble. As described in Section 4.2.1, features are divided into three subsets according to the types of POS tags. We first show the results of individual classifiers, and then report the results of three ensemble strategies, namely ensemble of feature sets, ensemble of classification algorithms and ensemble of both feature sets and classification algorithms, respectively.

5.2.1. Results of individual classifiers

In this part, we give the results of individual classifier. For comparison, we also give the result of joint features, denoted by Joint POS. In Joint POS tags, we gather features from the three subsets together. Each feature set is classified by three individual algorithms (NB, ME and SVM). The results are presented in Table 3.

Firstly, we focus on the comparison on different feature sets. POS-1 consistently performs the best among the three feature subsets. The performance of POS-2 and POS-3 drops by 5–10% in comparison with POS-1, yet it is still effective to some extent. This confirms that adjectives and adverbs are the most significant POSs tags related to sentiment; although verbs and nouns are not as significant as adjectives and adverbs, they still act as important features for sentiment classification.

Secondly, we observe the performance of different classifiers. For simplicity, we focus on the average results of five datasets. On individual feature sets (POS-1, POS-2 and POS-3), NB always yields the best performance, but it turns out to be worse on Joint POS. We conclude that NB seems to work better on feature sets with a smaller size, while, by contrast, ME and SVM appear to be more effective for high-dimensional feature sets. One possible explanation might be that the conditional independency assumption of NB hardly holds when the dimension is high while ME and SVM, as generative models, are better at representing the complexity of non-independent features.

5.2.2. Ensemble of feature sets

In this part, we investigate ensemble of feature sets. We choose SVM as the base-classifier and then combine the scores of component feature sets. In Table 4, the results of the ensemble methods described in Section 4.4 are all presented. Note that

[6] http://www.nltk.org/.
[7] http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html
[8] http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html

Table 4
Accuracies (%) of ensemble of POS-based feature sets using SVM as base-classifier.

Dataset	Individual feature set				Fixed combination		Weighted combination		Meta-classifier combination					
	POS-1	POS-2	POS-3	Joint POS	Vote	Sum	AveP	MCE	Linear regression			AveP	SVM	Logit
									LMS	CE	MCE			
Movie	79.05	70.10	75.70	84.85	82.15	85.15	84.90	**85.55**	84.60	84.95	84.60	85.00	85.40	85.15
Book	71.85	67.15	64.65	75.60	74.40	76.40	76.45	**76.75**	75.95	76.40	76.20	76.50	76.30	76.70
DVD	73.00	68.20	66.65	77.60	75.30	78.65	**79.15**	78.85	78.70	78.80	78.50	79.30	78.60	79.10
Elec	75.45	69.90	66.35	78.30	77.20	**81.35**	80.80	81.05	80.55	80.75	80.55	81.20	81.25	80.90
Kitchen	78.45	70.40	66.45	82.10	79.50	82.95	83.50	**83.40**	83.10	83.25	83.00	83.30	83.35	83.35
Average	76.70	71.80	69.21	80.01	77.71	80.90	80.96	**81.12**	80.58	80.83	80.57	81.06	80.98	81.04

in weighted combination, we find that the CE and LMS criteria do not guarantee the convergence, hence we only show the results of AveP and MCE. The results of individual feature sets are also listed for comparison.

We place an emphasis on comparison with Joint POS. Except for the voting rule, the performance of all ensemble methods are better than Joint POS. The poor performance of the voting rule is probably due to the reason that the voting rule does not benefit from all of the component feature sets. This indicates that using the same classification algorithm; ensemble of feature sets is generally more effective than joint features.

5.2.3. Ensemble of classification algorithms

In this part, we examine ensemble of classification algorithms. We perform experiments on the Joint POS feature set. The participant classification algorithms are NB, ME, and SVM. The accuracies of individual classification algorithms and ensemble methods are reported in Table 5.

As shown in the Table 5, almost all the ensemble methods get improvements over individual classifiers. Although on the Movie dataset, the ensemble methods do not show much superiority (ME works really well in this case) while the improvements on the other four datasets are comparatively higher. Let us review the conclusions in Section 5.2.1: The best individual classification algorithm is problem-dependent. Thus, an efficient ensemble may leverage distinct strengths and robustly enhance the system performance across datasets.

5.2.4. Ensemble of feature sets and classification algorithms

In this part, we consider ensemble of both feature sets and classification algorithms. Table 6 presents the detailed results of ensemble of both POS-based feature sets and classification algorithms. In order to show the comparative results more clearly, we give the average accuracies of three ensemble strategies in Table 7.

We first compare the three ensemble strategies (see Table 7). Strategy-3 performs consistently better than Strategy-1 across all methods and datasets. It is thus concluded that an ensemble of feature sets and classification algorithms together is more effective than only combining feature sets. Does Strategy-3 still significantly outperform Strategy-2? As we might expect, the answer is not so clear: Fixed combination is overall inferior to that in Strategy-2. One possible reason is that the voting rule does not benefit from an ensemble of feature sets (which we have analyzed in Section 5.2.2). Regarding weighted combination and meta-classifier combination, the situation has improved: Strategy-3 is slightly better overall than Strategy-2.

Secondly, we make some internal comparisons with concrete combination methods (see Table 6, as well as 4 and 5). The fixed rules give lower accuracies than the trained methods in all three ensemble strategies, where the sum rule is robustly better than the voting rule. In trained methods, the weighted combination and meta-classifier combination generally show similar performance. Although the model of weighted combination is simpler (with fewer weights), its performance is not worse than the latter. When comparing concrete methods, we find that no single one consistently performs the best. Generally speaking, the AveP-based and MCE-based weighted combination, and the CE-based meta-learning classifier give comparatively better results.

Table 5
Accuracies (%) of ensemble of classification algorithms on joint POS features.

Dataset	Individual classifiers			Fixed combination		Weighted combination		Meta-classifier combination					
	NB	ME	SVM	Vote	Sum	AveP	MCE	Linear regression			AveP	Linear SVM	Logit
								LMS	CE	MCE			
Movie	82.70	86.10	84.85	86.45	86.50	85.90	86.20	86.50	**86.6**	86.50	86.10	86.10	86.05
Book	76.70	78.10	75.60	78.75	78.95	78.75	79.30	79.40	79.20	79.10	78.95	78.15	**79.60**
DVD	78.85	78.95	77.60	79.90	80.40	80.65	80.55	80.10	**80.85**	80.15	80.55	78.90	80.70
Elec	81.75	80.35	78.30	80.80	81.05	**82.95**	82.35	82.20	82.65	81.90	**82.95**	81.50	82.55
Kitchen	82.40	82.50	82.10	83.35	83.50	84.50	**85.05**	84.20	85.00	84.05	84.45	82.75	84.80
Average	80.48	81.20	79.69	81.85	82.08	82.55	82.69	82.48	**82.86**	82.34	82.60	81.48	82.74

Table 6
Accuracies (%) of ensemble of POS-based feature sets and classification algorithms.

Dataset	Individual classifiers on joint POS			Fixed combination		Weighted combination		Meta-classifier combination					
	NB	ME	SVM	Vote	Sum	AveP	MCE	Linear regression			AveP	Linear SVM	Logit
								LMS	CE	MCE			
Movie	82.70	86.10	84.85	85.20	86.15	86.80	86.45	86.60	**86.85**	86.70	86.80	86.45	86.80
Book	76.70	78.10	75.60	77.30	77.65	**80.10**	78.85	78.70	79.40	78.95	79.50	79.40	79.15
DVD	78.85	78.95	77.60	79.35	80.60	80.40	80.60	80.85	**81.15**	80.50	80.80	80.35	81.05
Elec	81.75	80.35	78.30	81.05	81.95	**83.40**	83.05	82.70	83.20	82.70	83.35	82.90	82.95
Kitchen	82.40	82.50	82.10	82.25	82.90	84.90	83.85	84.35	**85.00**	84.35	84.85	84.80	84.70
Average	80.48	81.20	79.69	81.03	81.85	**83.12**	82.56	82.64	**83.12**	82.64	83.06	82.78	82.93

Table 7
Average accuracies (%) of three ensemble strategies.

Dataset	Strategy-1: ensemble of feature sets			Strategy-2: ensemble of classification algorithms			Strategy-3: ensemble of feature sets and classification algorithms		
	Fixed	Weighted	Meta	Fixed	Weighted	Meta	Fixed	Weighted	Meta
Movie	83.65	85.23	84.95	86.48	86.05	86.31	85.68	86.63	86.70
Book	75.40	76.60	76.34	78.85	79.03	79.07	77.48	79.48	79.18
DVD	76.98	79.00	78.83	80.15	80.60	80.21	79.98	80.50	80.78
Elec	79.28	80.93	80.87	80.93	82.65	82.29	81.50	83.23	82.97
Kitchen	81.23	83.45	83.23	83.43	84.78	84.21	82.59	84.38	84.68
Average	79.31	81.04	80.84	81.97	82.62	82.42	81.44	82.84	82.86

5.3. Experiment II – WR-based ensemble

In Section 5.2, we have presented the experimental results of POS-based Ensemble. In this section, we report the performance of WR-based ensemble. Component feature sets are categorized according to the pattern of word relations: Unigrams, Bigrams and Dependency relations. As before, the presentation of experimental results is still organized in four parts. We first report the results of individual classifiers and then give the results of each of three ensemble strategies.

5.3.1. Results of individual classifiers

The accuracies of individual classifiers as well as the results of joint WR features (including unigrams, bigrams, and dependencies) are reported in Table 8.

Three individual WR feature sets generally show comparable performance and the Joint WR features significantly outperform the individual feature sets. It should be noted that the performance of different classification algorithms seems inconsistent between different feature sets. Specifically, with unigram and joint WR feature, the performance of discriminative models (SVM and ME) seems equal to or even better than the generative model (NB); while on Bigrams and Dependencies, however, the performance of generative model (NB) is far superior. This is in accordance with the results reported by Pang et al. [30]. We speculate that bigrams and dependency relations cover some of the relevance corresponding to unigram pairs, which meets the requirement of feature independency assumption of NB.

5.3.2. Ensemble of feature sets

Table 9 shows the performance of an ensemble of WR-based feature sets using SVM as the base-classifier. We also give the results of the three individual feature sets and the joint feature set for comparison.

Table 8
Accuracies (%) of individual classification algorithms on individual WR-based feature sets.

Dataset	Unigrams			Bigrams			Dependencies			Joint WR features		
	NB	ME	SVM	NB	ME	SVM	NB	ME	SVM	NB	ME	SVM
Movie	82.95	86.30	84.75	84.95	81.95	83.35	83.90	80.70	81.95	85.80	85.40	86.45
Book	77.60	76.35	74.70	79.45	78.20	76.70	78.65	75.70	75.15	81.20	78.45	77.65
DVD	79.60	79.00	77.20	79.75	77.10	76.35	78.45	75.15	73.75	81.70	80.00	79.45
Elec	81.75	80.25	80.05	83.45	80.90	79.50	80.80	79.00	78.90	84.15	82.95	82.50
Kitchen	82.80	81.45	83.25	86.65	81.85	82.05	84.20	80.70	80.95	87.50	85.35	85.40
Average	80.94	80.67	79.99	82.85	80.00	79.59	81.20	78.25	78.14	84.07	82.43	82.29

Table 9
Accuracies (%) of ensemble of WR-based feature sets using SVM as base-classifier.

Dataset	Individual feature set				Fixed combination		Weighted combination		Meta-classifier combination					
	Uni	Bi	Dp	Joint	Vote	Sum	AveP	MCE	Linear regression			AveP	SVM	Logit
									LMS	CE	MCE			
Movie	84.75	83.35	81.95	86.45	85.90	86.15	87.20	86.65	86.55	87.00	86.65	87.00	87.15	**87.25**
Book	74.70	76.70	75.15	77.65	77.70	78.15	78.30	78.15	78.30	78.15	78.20	**78.35**	78.25	78.25
DVD	77.20	76.35	73.75	79.45	79.35	80.45	80.70	**81.00**	80.80	80.65	80.70	80.90	80.75	80.90
Elec	80.05	79.50	78.90	82.50	82.20	82.90	83.10	**83.35**	83.25	83.15	83.10	83.15	83.05	83.20
Kitchen	83.25	82.05	80.95	85.40	85.50	85.80	86.10	**86.75**	85.85	85.90	85.85	86.10	86.10	**86.75**
Average	79.99	79.59	78.14	82.29	82.13	82.69	83.08	83.18	82.95	82.97	82.90	83.10	83.06	**83.27**

Again, we focus on the comparison with Joint features. Most of the ensemble methods can outperform Joint features, but with limited improvements. The voting rule is even inferior to Joint features. We guess that using one fixed classification algorithm to combine different types of feature sets may not achieve the best ensemble results since the fixed classification algorithm does not guarantee compatibility with all the component feature sets. For instance, SVM is not as effective on Bigrams and Dependencies as it is on Unigrams.

5.3.3. Ensemble of classification models

Table 10 gives the results of ensemble of classification algorithms. The participant classification algorithms are NB, ME, and SVM, and each of the component classifiers is trained on the Joint WR features. The accuracies of individual classification algorithms are also reported for comparison.

As shown in the Table 10, the ensemble methods are more effective compared to individual classification algorithm. But based on the conclusion in Section 5.3.2, the ensemble result of one fixed classifier combining different component feature sets is better than one using Joint features. Therefore, we believe such ensemble is non-optimal. We will seek ensemble of both feature sets and classification algorithms in next section.

5.3.4. Ensemble of feature sets and classification algorithms

Table 11 presents the detailed performance of ensemble of WR-based feature sets and classification algorithms. Similarly, we give the average accuracies of three ensemble strategies in Table 12.

We first compare three ensemble strategies (Table 12) once again. Strategy-3 still performs significantly better than strategy-1. Slightly different from POS-based ensemble, this time the fixed combination in strategy-3 seems to work better and the improvements are significant on all five datasets. With regard to the trained methods, they are still more effective except

Table 10
Accuracies (%) of ensemble of classification algorithms on joint word-relation features.

Dataset	Individual classifiers			Fixed combination		Weighted combination		Meta-classifier combination					
	NB	ME	SVM	Vote	Sum	AveP	MCE	Linear regression			AveP	Linear SVM	Logit
								LMS	CE	MCE			
Movie	85.80	85.40	86.45	87.10	**87.40**	86.60	87.20	87.20	87.35	87.10	86.75	86.30	87.10
Book	81.20	78.45	77.65	79.60	80.95	82.20	82.80	82.55	82.75	**83.00**	82.35	81.20	82.75
DVD	81.70	80.00	79.45	80.95	81.15	82.30	82.25	82.10	82.70	81.70	82.40	81.70	**82.50**
Elec	84.15	82.95	82.50	84.25	84.55	85.30	**85.40**	85.10	85.00	85.05	85.15	84.10	**85.40**
Kitchen	87.50	85.35	85.40	86.35	86.90	87.55	87.45	87.60	**87.75**	87.15	**87.75**	87.50	87.70
Average	84.07	82.43	82.29	83.65	84.19	84.79	85.02	84.91	**85.11**	84.80	84.88	84.16	85.09

Table 11
Accuracies (%) of ensemble of WR-based feature sets and classification algorithms.

Dataset	Individual classifiers			Fixed combination		Weighted combination		Meta-classifier combination					
	NB	ME	SVM	Vote	Sum	AveP	MCE	Linear regression			AveP	Linear SVM	Logit
								LMS	CE	MCE			
Movie	85.80	85.40	86.45	87.15	87.00	87.70	87.85	87.80	**88.00**	87.45	87.35	87.00	87.65
Book	81.20	78.45	77.65	80.90	81.70	81.80	82.00	82.05	82.60	82.00	82.55	**82.65**	82.45
DVD	81.70	80.00	79.45	82.65	82.75	**83.80**	83.05	82.85	83.10	82.75	83.25	82.70	83.40
Elec	84.15	82.95	82.50	84.80	85.50	85.95	85.65	85.55	85.65	85.50	85.50	85.20	**86.00**
Kitchen	87.50	85.35	85.40	87.80	87.35	**88.65**	88.25	87.4	87.85	87.30	88.60	87.55	88.55
Average	84.07	82.43	82.29	84.66	84.86	85.58	85.36	85.13	85.44	85.00	85.45	85.02	**85.61**

Table 12
Average accuracies (%) of three ensemble strategies.

Dataset	Strategy-1: ensemble of WR-based feature sets			Strategy-2: ensemble of classification algorithms			Strategy-3: ensemble of feature sets and classification algorithms		
	Fixed	Weighted	Meta	Fixed	Weighted	Meta	Fixed	Weighted	Meta
Movie	86.03	86.93	86.93	87.25	86.90	86.97	87.08	87.78	87.54
Book	77.93	78.23	78.25	80.28	82.50	82.43	81.30	81.90	82.38
DVD	79.90	80.85	80.78	81.05	82.28	82.18	82.70	83.43	83.01
Elec	82.55	83.23	83.15	84.40	85.35	84.97	85.15	85.80	85.57
Kitchen	85.65	86.43	86.09	86.63	87.50	87.58	87.58	88.45	87.88
Average	82.41	83.13	83.04	83.92	84.91	84.83	84.76	85.47	85.28

for the Book dataset (a slight decline). Therefore, it is reasonable to draw the conclusion that combining both feature sets and classification algorithms together is the most attractive option.

When comparing different combination methods, our conclusions are similar to those about the POS-based ensemble. The trained methods still yield better accuracies overall than the fixed rules. With regard to fixed rules, the sum rule is usually better than the voting rule. In trained methods, the average performance of weighted combination and meta-classifier remain similar.

Finally, we compare different training methods in weighted combination and meta-learning classifier. Considering Tables 9–11, different ensemble methods generally show similar performance, such that none of them can be selected as the absolute winner. This is in accordance with the conclusions drawn from the POS-based ensemble. Among these methods, the AveP based weighted combination and the Logit Model perform comparatively better.

6. Discussion

Based on the experimental results, we make in-depth discussion about the three questions raised in the introduction. Questions (1)–(3) are answered in Sections 6.1,6.2,6.3, respectively. Moreover, the computational cost of our ensemble system is discussed in Section 6.4.

6.1. Comparisons with related work

The POS-based ensemble uses unigram words as features. The WR-based ensemble uses bigrams and dependency relations as extra features in addition to unigrams. Bigrams and dependency features encode the word relation information behind the text, which are helpful to sentiment classification. Therefore, the WR-based ensemble is more effective in its classification accuracy at the cost of higher computational complexity.

Related work can also be categorized into two groups according to whether features beyond unigrams are used or not. We take the performance of SVM with unigrams as Baseline 1, and the performance of SVM with Joint WR features as Baseline 2. Related work and our ensemble methods that (1) only use unigram features, and (2) also use WR features, are compared with Baseline 1 and Baseline 2, respectively. The performance of both baselines and our best ensemble results (AveP-based weighted combination) are reported in Table 13.

Riloff et al. [34] propose a hierarchy structure to extract subjective nouns based on POS information. Their method improves upon Baseline 1 by an average of 1.4%. Compared with their approach, our POS-based ensemble outperforms Baseline 1 by 3.31%. Moreover, bigram features are used in their approach while we only use unigrams. In the literature [20], unigrams with different POS tags are trained on SVM, and then base-classifiers are selected and combined using several fixed combination rules. The sum rule yields the best performance and outperforms Baseline 1 by 2.54% on the Movie dataset V1.0. In fact, we have carried out similar experiments in Section 5.2.2 (Table 4); the average performance of only combining feature sets is not that high, only outperforming Baseline 1 by average 0.91% (79.99% vs. 80.90%). We update their approach in

Table 13
Average accuracies (%) of baseline systems and ensemble systems.

Dataset	Baseline 1: unigrams	POS-based ensemble	Baseline 2: joint WR	WR-based ensemble
Movie	84.75	86.80	86.45	87.70
Book	74.70	**80.10**	77.65	81.80
DVD	77.20	80.40	79.45	**83.80**
Elec	80.05	**83.40**	82.50	85.95
Kitchen	83.25	84.90	85.40	**88.65**
Average	79.99	83.12	82.29	85.58

Baseline 1 and 2 are the performance of SVM classifier on Unigram features and Joint WR features, respectively; as for ensemble methods, we report the performance of AveP-based weighted combination.

two ways: (1) creating an ensemble which combines classification algorithms as well as feature sets; (2) using trained combination methods, and finally achieve results in a 3.13% better performance (79.99% vs. 83.12%).

In most of the related work utilizing WR features, they were used as extra features in addition to unigrams. For example, Gamon [10] extracts an additional set of linguistic features (dependency relations) from phrase structure trees. But the single linguistic feature set performs poorly when compared to traditional features. Use of joint features (including dependency relation features and n-gram features) yields significant improvements in performance when compared with using only word n-gram features. In the literature [15], Joshi and Penstein-Rosé try to extract more generalized dependency features (by backing off one word in a relation feature to its POS tag) as extra features in addition to simple unigrams. Adding these joint features improves performance by 2.7% on their dataset. In our own experiments, the joint WR features (Baseline 2) outperform simple unigrams (Baseline 1) by 2.3%, which is similar to their results. With regard to the ensemble method, performance is more attractive than that of Joint features. Seen from Table 13, the WR-based ensemble (AveP-based weighted combination) yields an average accuracy of 85.58%, improving upon Baseline 2 by 3.29%.

6.2. Regarding the effectiveness of ensemble for sentiment classification

We now answer the second question. We first analyze the three ensemble strategies and then discuss in depth why the proposed ensemble method is effective when applied to sentiment classification. Regarding the effectiveness of three ensemble strategies, conclusions are drawn as follows:

- The performance of ensemble of feature sets using a single classification algorithm is generally better than that on joint features, with the exception of the voting rule. The voting rule works poorly since it cannot easily benefit from all component sets.
- Ensemble of classification algorithms on the same feature set perform robustly better than any individual classifier. All the ensemble methods yield better results, where the trained rules are still superior to the fixed rules.
- Ensemble of both feature sets and classification algorithms are the most effective when compared to the above two strategies. With our designed feature sets, ensemble of both is optimal compared to ensemble of either.

Generally, an ensemble usually benefits more from leveraging the distinct strengths of imbalanced subsets. Therefore, it is crucial how to design the imbalanced subsets before applying an ensemble technique. Taking the WR-based feature sets as an example, we give some explanations why they are effective in sentiment classification.

Different types of features have distinct distributions, and therefore would probably vary in performance with different machine learning algorithms. The generative model is optimal only if the distribution is well estimated; otherwise the performance will lag significantly. For example, NB performs poorly if the feature independence assumption does not hold. On the contrary, a discriminative model such as SVM is better at representing the complexity of relevant features. We concluded in Section 5.3.1 that ME/SVM performs significantly better than NB with unigram features while the outcome is the reversed with bigram features. One possible explanation is that the relevance between unigrams is higher than between bigrams (because bigrams themselves cover some of the relevance corresponding to unigram pairs).

In traditional classification algorithms, such as a linear classifier, the weights are assigned to each individual feature. These weights embody the importance of each feature to classification but are less useful in reflecting the differences between feature sets. In this case, ensemble technique is quite useful for leveraging the trade-off between component classifiers with different feature sets. The weights assigned to each component are learned automatically via machine learning techniques and represent the relevance of corresponding components to sentiment classification.

6.3. Regarding the winner among different ensemble methods

In this part, we answer the third question. Regarding the three kinds of ensemble methods, our conclusion is drawn as follows:

- Among the three kinds of ensemble methods (fixed rules, weighted combination and meta-classifiers), the weighted combination is the most attractive.
- The fixed rules give lower accuracies overall than the trained methods. Nevertheless, the sum rule can still be regarded as a low-cost yet effective approach (no need to estimate component weights).
- In trained methods, weighted combination shows a slight superiority when compared to meta-classifiers. Considering that there are also fewer parameters to tune in a weighted combination model, we would further recommend it over other models.
- Regarding the evaluated training methods in a trained combination, they generally yield similar performance and none has been proven to consistently outperform the others. The AveP-based weight combination is preferred since its overall performance is better and its computational cost is less than most of the others (no need to apply sigmoid function when perform SGD optimization).

Table 14
Execution time of training base-classifiers and traditional classifiers on movie dataset.

Classification algorithm	Unigrams (s)	Bigrams (s)	Dependency features (s)	Joint features (s)
NB	15	435	498	1020
ME	19	305	350	797
SVM	67	408	452	994

6.4. Regarding the computational complexity of ensemble system

Generally, one possible weakness of ensemble system is its higher computational cost when compared to a single classier system. However, in our ensemble method, the base-classifiers are trained on component feature sets rather than the entire feature set. Therefore, in this way, the increase of computational cost is not as great as the traditional ensemble methods such as bagging and boosting.

Taking the WR-based ensemble for example, we report the average execution time of training each of three classification models on three component feature sets (nine base-classifiers) as well as the set of the joint features (a traditional classifier) in Table 14. We can see from Table 14 that the computational cost of training is almost proportional to the dimension of the feature vectors. That is, the sum of the training time with all separate component feature sets is roughly equal to (or even less than) the training time with a set of joint features. We also conclude that, the computation cost of ensemble of both feature sets and classification algorithms (Ensemble Strategy-3) is also less than ensemble of classification algorithms (Ensemble Strategy-2) with joint features. Therefore, ensemble Strategy-3 is superior to Strategy-2 from the prospects of both classification accuracy and computational efficiency.

Note that in the trained combination, the computational cost is further increased. This is especially true when generating training samples for meta-learning as a 4-fold leave-one-out procedure is applied. Fortunately, the increase is linear. If we only use one validation set instead of the 4-fold stacking procedure, the cost could be reduced. Besides, there is also a computational cost associated with training weights for meta-learning. But the cost of training weights is still low because the dimension of the feature vector used for meta-learning is quite small ($9 * 2 = 18$).

Overall, the ensemble model does increase the computational cost, but the increase is entirely acceptable. Furthermore, the training and testing of base-classifiers could be conducted in parallel. Therefore, the total execution time (especially the testing time) would not be significantly higher than traditional methods.

7. Conclusions and future work

The aim of this paper is to evaluate ensemble technique for sentiment classification. In order to make this an extensive study, we consider two schemes of feature sets, three types of ensemble techniques, and three ensemble strategies to conduct a range of comparative experiments on five widely-used datasets, with the emphasis on the evaluation of effects of three ensemble strategies and comparisons of different ensemble methods. Experimental results demonstrate that using ensemble technique is an effective way to combine different feature sets and classification algorithms for better classification performance.

In this paper, we take advantage of ensemble frameworks for integrating different feature sets and classification algorithms to boost the overall performance. We are also interested in pursuing hybrid generative/discriminative models that are suitable for sentiment classification. In the literature [26], NB and ME were formulized as a generative/discriminative pair. Following literature had done some work on the hybrid NB/ME model for the tasks of text classification [9,32], this may be a promising direction for future research. Furthermore, we note that syntactic relations are significant features for sentiment classification, but it brings the additional problem of high computation complexity. In fact, we have proposed an ensemble model to integrate generalized WR features and a fast feature selection method for the WR features [41] to address this problem. We believe feature selection for syntactic relations may also be an important issue worthy of study in future work.

References

[1] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, V.S. Subrahmanian, Sentiment analysis: adjectives and adverbs are better than adjectives alone, in: Proceedings of the International Conference on Weblogs and Social Media (ICWSM), 2007.

[2] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: Proceedings of the Association for Computational Linguistics (ACL), 2007.

[3] J. Bridle, Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, Neurocomputing: Algorithms, Architectures and Applications 227 (1990) 236.

[4] H. Cui, V. Mittal, M. Datar, Comparative experiments on sentiment classification for online product reviews, in: Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI), 2006.

[5] K. Dave, S. Lawrence, D.M. Pennock, Mining the peanut gallery: opinion extraction and semantic classification of product reviews, in: Proceedings of the International World Wide Web Conference (WWW), 2003, pp. 519–528.

[6] Y.-S. Dong, K.-S. Han, A comparison of several ensemble methods for text categorization, in: The 2004 IEEE International Conference on Services Computing (SCC), 2004, pp. 419–422.

[7] S. Džeroski, B. Ženko, Is combining classifiers with stacking better than selecting the best one?, Machine Learning 54 (2004) 255–273

[8] Y. Freund, R. Schapire, Large margin classification using the perceptron algorithm, Machine Learning 37 (1999) 277–296.

[9] A. Fujino, N. Ueda, K. Saito, A hybrid generative/discriminative approach to text classification with additional information, Information Processing and Management 43 (2007) 379–392.

[10] M. Gamon, Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2004.

[11] S. Hashem, Optimal linear combinations of neural networks, Neural Networks 10 (1997) 599–614.

[12] V. Hatzivassiloglou, J. Wiebe, Effects of adjective orientation and gradability on sentence subjectivity, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2000, pp. 299–305.

[13] T. Ho, J. Hull, S. Srihari, Decision combination in multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 66–75.

[14] T. Joachims, C. Nedellec, C. Rouveirol, Text categorization with support vector machines: learning with many relevant, in: Proceedings of the European Conference on Machine Learning, Springer, 1998, pp. 137–142.

[15] M. Joshi, C. Penstein-Rosé, Generalizing dependency features for opinion mining, in: Proceedings of the 47th ACL and the 4th IJCNLP Conference, Association for Computational Linguistics, 2009, pp. 313–316.

[16] B. Juang, S. Katagiri, Discriminative learning for minimum error classification [patternrecognition], IEEE Transactions on Signal Processing 40 (1992) 3043–3054.

[17] J. Kittler, Combining classifiers: a theoretical framework, Pattern Analysis and Applications 1 (1998) 18–27.

[18] L. Larkey, W. Croft, Combining classifiers in text categorization, in: Proceeding of ACM SIGIR Conference, ACM, New York, NY, USA, 1996, pp. 289–297.

[19] D. Lewis, Naive (Bayes) at forty: the independence assumption in information retrieval, Lecture Note in Computer Science 1398 (1998) 4–18.

[20] S. Li, C. Zong, X. Wang, Sentiment classification through combining classifiers with multiple feature sets, in: Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 07), 2007, pp. 135–140.

[21] B. Liu, Sentiment analysis: a multifaceted problem, IEEE Intelligent System 25 (2010) 76–80.

[22] C. Liu, Classifier combination based on confidence transformation, Pattern Recognition 38 (2005) 11–28.

[23] W. Liu, X. Quan, Feng Min, Q. Bite, A short text modeling method combining semantic and statistical information, Information Sciences 180 (2010) 4031–4041.

[24] A. McCallum, K. Nigam, A comparison of event models for naive Bayes text classification, in: Proceedings of the AAAI Workshop on Learning for Text Categorization, Citeseer, 1998.

[25] R. McDonald, F. Pereira, K. Ribarov, J. Hajic, Non-projective dependency parsing using spanning tree algorithms, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), 2005, pp. 523–530.

[26] A. Ng, M. Jordan, On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes, Advances in Neural Information Processing Systems 2 (2002) 841–848.

[27] K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, in: Proceedings of IJCAI Workshop on Machine Learning for Information Filtering, 1999, pp. 61–67.

[28] B. Pang, L. Lee, A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts, in: Proceedings of the Association for Computational Linguistics (ACL), 2004, pp. 271–278.

[29] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (2008) 1–135.

[30] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.

[31] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Advances in Large Margin Classifiers (1999).

[32] R. Raina, Y. Shen, A. Ng, A. McCallum, Classification with hybrid generative/discriminative models, Advances in Neural Information Processing Systems 16 (2004) 545–552.

[33] A. Ratnaparkhi, A maximum entropy model for part-of-speech tagging, in: Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP), 1996, pp. 133–142.

[34] E. Riloff, S. Patwardhan, J. Wiebe, Feature subsumption for opinion analysis, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2006, pp. 440–448.

[35] E. Riloff, J. Wiebe, T. Wilson, Learning subjective nouns using extraction pattern bootstrapping, in: Proceedings of the Conference on Natural Language Learning (CoNLL), 2003, pp. 25–32.

[36] V. Subrahmanian, D. Reforgiato, AVA: adjective–verb–adverb combinations for sentiment analysis, IEEE Intelligent Systems 23 (2008) 43–50.

[37] C. Sutton, A. McCallum, An introduction to conditional random fields for relational learning, in: Introduction to Statistical Relational Learning, 2007.

[38] M. Whitehead, L. Yaeger, Sentiment mining using ensemble classification models, in: International Conference on Systems, Computing Sciences and Software Engineering (SCSS 08), Springer, 2008.

[39] Y. Yang, X. Liu, A re-examination of text categorization methods, in: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), ACM, New York, NY, USA, 1999, pp. 42–49.

[40] H.-T. Zheng, B.-Y. Kang, H.-G. Kim, Exploiting noun phrases and semantic relationships for text document clustering, Information Sciences 179 (2009) 2249–2262.

[41] R. Xia, C. Zong, Exploring the use of word relation features for sentiment classification, in: Proceedings of the 23rd International Conference on Computational Linguistics (COLING), 2010, pp. 1336–1344.