J. Cent. South Univ. (2012) 19: 1334−1340
DOI: 10.1007/s11771-012-1146-7

Springer

Construction of unsupervised sentiment classifier on idioms resources

XIE Song-Xian(谢松县),WANG Ting(王挺)

Department of Computer Science and technology, School of Computer, National University of Defense Technology ChangSha, 410073, China

Abstract: Sentiment analysis is the computational study of how opinions, attitudes, emotions, and perspectives are expressed in language, and has been the important task of Natural Language Processing in recent years. Sentiment analysis is highly valuable for both research and practical applications. The focuses were put on the difficulties in the construction of sentiment classifiers which normally need tremendous labeled domain training data, and a novel unsupervised framework has been proposed to make use of the Chinese idiom resources to develop a general sentiment classifier. Furthermore, the domain adaption of general sentiment classifier was improved by taking the general classifier as the base of a self-training procedure to get a domain self-training sentiment classifier. To validate the effect of the unsupervised framework, several experiments were carried out on publicly available Chinese online reviews dataset. The experiments show that the proposed framework is effective and achieves encouraging results. Specifically the general classifier outperforms two baselines (a naïve 50% baseline and a cross-domain classifier), and the bootstrapping self-training classifier approximates the upper bound domain-specific classifier with lowest accuracy of 81.5%, but the performance is more stable and the framework needs no labeled training dataset.

Keywords: sentiment analysis; sentiment classification; bootstrapping; idioms; machine learning; unsupervised approach

1 Introduction

The amount of user-generated content (UGC) on the Internet has risen exponentially over the last decade with the emergence and advance of web 2.0 technology, and such content is now always at our fingertips. UGC, in particular, become an ever-growing source of opinions and sentiments which are spread worldwide through blogs, wikis, chats and diverse social networks such as Twitter and Facebook[1]. The distillation of subjective knowledge from such abundant sources is an important part of applications in fields such as commerce, tourism, politics and health, but the quantity and nature of user-generated contents make it a very difficult and challenging task.

As a live example of daily life, more and more review sites continue to grow in popularity as more and more people begin to refer the advice of fellow users regarding services and products before they make their dealing decision. However, with the explosion of such information, users are often forced through large quantities of and sometimes low quality reviews in order to find the useful information they really need. This has led to increasing research interest in the areas of opinion mining and sentiment analysis, with the goal of finding effective methods and techniques that can automatically analyze reviews and extract the subjective information to be summarized for the users[2].

Sentiment analysis is the computational study of how opinions, attitudes, emotions, and perspectives are expressed in language (especially in written text), so as to provide tools and techniques for extracting this kind of evaluative information from large datasets and summarizing it[3]. With the growing need of identifying opinions and sentiments automatically from text data on the web, sentiment analysis has received considerable attention recently, and been applied to Business Intelligence, Public Opinion Analysis, Election Prediction, etc.

Sentiment classification, which deals with determining sentiment orientation of target text, is one of sentiment analysis tasks[4]. The task can be viewed as a specific text categorization problem. Given an instance of opinionated text (may be document, sentence or words, since document sentiment classification is investigated, text stands for document), the goal is to classify it as positive or negative , or neutral in multi-class classification sometimes[5]. In fact, sentiment classification is a more challenging task than text classification. Firstly, methods and techniques developed in traditional text classification usually do not work well on this task, since they tend to take frequent-occurring words, which are also called keywords, as good indicators of the class a document belongs to. However, for sentiment classification of an opinionated document, words indicating sentiment are usually ambiguous and maybe infrequent. Secondly and most importantly, sentiment expressions critically depend on domains and contexts[6], and opinions are often hidden in a

large amount of domain dataset, so there are often no universal sentiment resources available for sentiment classification. However human-labeled resources for each domain are costly and difficult, for manual annotation is very expensive and time-consuming.

Since sentiment classification can be viewed as specific text categorization[4], many researchers have cast their eyes on all kinds of machine learning techniques. With more and more work on document-level sentimental polarity classification using machine learning methods, various classifiers and feature sets have been explored[4][7], which can be categorized into supervised, semi-supervised and unsupervised approaches.

Supervised approaches were firstly applied to sentiment classification by Pang, et al.[4] by comparing multiple supervised machine learning algorithms (Naive Bayes, maximum entropy, support vector machines) for the task of movie reviews sentiment classification. Afterwards various classifiers and features selection has manifested in many other researches[8–13]. Performance of supervised approaches is reasonably satisfying because of the requirement that test data should be similar to manually annotated training data in the same domain, and the high accuracy often is the upper bound for other approaches to compare with. But collecting annotated data in the new domain and retraining the classifier are unavoidable for adapting a supervised sentiment classifier to another domain. The dependency on domain annotated training dataset is one major shortcoming of all supervised approaches.

Semi-supervised approaches try to improve the performance of classification by fitting labeled and unlabeled datasets together[14] with various methods such as EM on Naïve Bayes, co-training, transductive SVMs, and co-regularization, etc.[15–18].But just as supervised methods, labeled training datasets are needed for semi-supervised approaches, which are mainly annotated by hand, whereas there are some automated means, because reliability of training data is the main consideration for learning methods. And most importantly, training datasets are critically domain-dependent whether annotated automatically or manually, so different training dataset is needed to be labeled for different domain, which means supervised and semi-supervised sentiment classification is very hard for the domain without any training dataset. Although in recent years many researchers have tried to solve the training dataset shortage problem by adaptive techniques (such as transfer learning)[19–21] to realize cross-domain or cross-language[20], [22–24] sentiment classification, most are inefficient in that adaptive learning is needed for the changing of target domain. Furthermore the low accuracy is another problem of adaptive methods because of disambiguation of sentiment word in different domain or languages.

At the same time, many unsupervised approaches are brought forth to tackle the problem of annotated training data shortage and domain dependence[25–32], which are mostly domain-independent rules based, and try to get some highly confident examples produced by the rule-classifier as the training data for bootstrapping learning of next stage. These rules rely on universal sentiment expressions in the language recognized based on linguistic expertise knowledge. Therefore there are two problems of these methods: firstly, linguistic expertise knowledge are needed for the manually produced or automatically learned rules; secondly, discovering all sentiment expressions in the text by a few rules is impossible because human intuition may not be always correct and comprehensive for the complication and variety of language[4], so limited or biased examples are returned as the low coverage of the rules, which will influence the performance of supervised pipeline classifier of the bootstrapping procedure.

The problem of domain dependence of sentiment classification was focused on in this work, and to overcome such constraint, an unsupervised framework based on general resources independent of domains was put forward. In the framework, without the need of laborious labeling training data, a general classifier was trained on the off-the-shelf resources which are highly opinionated but do not depend on any context and domain. The proposed general classifier could output highly confident instances as training data for the next stage bootstrapping self-training domain classifier.

2 Formulation of sentiment classification

Sentiment classification aims to automatically classify document as predefined sentiment polarity classes of binary (negative or positive) or multi-class (negative, positive or neutral), and binary classification has been studied for simplicity. Formally, Given document corpus $D = \{d_1, \dots, d_n\}$, and predefined sentiment category set $C = \{1, -1 | positve = 1, negative = -1\}$, the task of sentiment classification is to predict each d_i in D with a label c_i expressed in C it should belong to.

To be along with text categorization, each document can be represented as a vector of bag-of-words features $x = R^n$, where n is the size of a pre-specified vocabulary V. The weight of each entry in this vector usually is specified as binary, with weight equals to 1 for terms present in the vector and 0 for absent.

Given a training dataset $X = \{x_1, \dots, x_m\}$, we can build a binary classifier:

$$f: X \to Y,\ Y = \{-1,1\} \quad (1)$$

and employ it to predict label for an unseen instance x by computing $f(x)$, with each instance represent as a vector $x = (w_1, \dots w_v)$, in which w_i is the ith feature's weight.

3 Hypothesis of feature space division

Often there is an implicit hypothesis underlying previous sentiment classification researches, which considers all features appearing in a document vector representing the document's sentiment polarity equally with different binary weights, for example in the following English book review which obviously expresses positive sentiment:

Example: *The book is recommended and sent to me by one of my good friends, and he told me to put it beside my pillow so as to read it anytime available, I read it through without any letup with the same idea, and I feel the translation very accurate without any mark of translation, the language very lively and vividly and the story easily understood to make the reader personally on the scene, and there is much use for reference in the book. I am going to read it again. Recommend!*

When computationally classifying sentiment polarity of the review, all words extracted as features are considered potentially indicators of positive evaluation for the book equally. However with careful considerations, it could be found that words "recommend" and "accurate" are positive indicators of reviews almost across all domains, while "lively and vividly", "personally on the scene" and "use for reference" are more frequently used in the book reviews to express positive evaluation. With this intuition, an assumption has been proposed as:

Assumption 1: In the feature space of sentiment classification, bag-of-words features can be divided into two different parts:

- domain-independent part, i.e. general sentiment features, which are indicators of sentiment polarity across all domains and independent of any context of any domain;
- domain-dependent part, i.e. specific sentiment features, sentiment polarity of which depend on specific context of each domain.

Formally, feature vector representing a document of the sentiment classification task can be expressed as $x = (w_1, \dots w_l, w_{l+1}, \dots w_v)$, and based on assumption 1, feature vector x could be divided into two parts:

$$x = \begin{cases} x_g : general\ sentiment\ features \\ x_s : specific\ sentiment\ features \end{cases} \quad (2)$$

where $x_g = (w_1, \dots w_l)$ denotes the weights of general part of features, and $x_s = (w_{l+1}, .. w_v)$ denotes the weights of specific part of features.

Questions might arise about assumption 1:

Firstly, what's the meaning of the division of feature space, and how to identify each part of feature space?

Imagine such a scenario, when reading a review about a professional book, that one could still distinguish which polarity (recommend or not) the reviewer prefer even if he knows nothing about the domain knowledge what the book describes, as long as "good", "accurate", "recommend" appear in the review. Intuitively, this kind of phenomenon may be explained by the general part of the text which is used to express the holistic sentiment polarity of the author, and the polarity of general sentiment words are prone to be recognized by anyone independent of domain knowledge. Comparably, in sentiment classification, we put forward that, the sentiment polarity of a document could still be recognized with only the text segment of general part of feature space x_g. That is to say, theoretically, if general sentiment knowledge could be modeled, what sentiment polarity a review prefers for could be still classified based on such general sentiment models.

The second question is how to establish such kind of general sentiment model?

Many researchers have tried to establish all kinds of sentimental ontology lexicons to represent general knowledge about human's sentiment, such as Sentiwordnet[33] and General Inquiry[34] in English, Hownet and NTUSD(Chinese Network Sentiment Dictionary)[35] in Chinese, etc. However, they all failed in modeling the universal sentiment knowledge in that many entries of these lexical resources have multiple senses with different sense representing different sentiment polarity, and the exact sense unavoidably depends on the context of each domain. Actually such knowledge exists in many cases, in which one word or combination of a few words could identify its own exact sentiment polarity independent of domains, such as idioms and proverbs. So the way that models the universal sentiment knowledge could be transformed as training a classifier on such resources with sentimental polarity independent of context and domains.

Then another question is: how to find such kind of instances?

In fact, this question has motivated our research at the very beginning. There are many linguistic resources highly valuable for sentiment classification, of which idiom resources attract interests of this research. Idioms are common phenomena of many languages beside Chinese, such as "castles in the air", "a bed of thorns", "bring down the house" in English. The form of idioms is succinct and the meaning is penetrating, which is a quintessence part of the language. Generally speaking, the structure of idioms is fixed and can't be changed at will; idioms have semantic intactness, which is not generally the simple

summation of the literal meaning of each component; idioms chiefly use metaphor, exaggerator and comparison in the rhetoric to express its real meaning; and most importantly, the sentimental orientation of idioms is independent and unchangeable under any context. There are many off-the-shelf lexical idiom resources in all kinds of languages with entries take the example form as:

Example: *castles in the air: a derogatory term, indicate the illusive things or impractical fanciness metaphorically.*

In this example, the entry is composed of three parts: the idiom "castles in the air", the semantic orientation "a derogatory term" representing negative sentiment polarity and a short paraphrase with three general negative words ("illusive", "impractical" and "fanciness"). The entry provides us with an illustrational universally-labeled sentimental example with general sentiment features and a negative label. Most importantly, the sentiment polarity of such instance is independent of any domain just as the idiom it explains. Based on such observation, another assumption has been proposed as follows:

Assumption 2: The sentiment polarity label of the paraphrase is independent of domains as the idiom it describes.

With assumption 2 admitted, the labeled training dataset could be constructed on idiom resources with the paraphrases of idioms as train instances, the semantic orientation values as sentimental labels. With such kind of training dataset at hand, a domain-independent classifier could be trained to model the universal sentiment knowledge, with the paraphrases being represented as vectors of general sentiment features.

4 Framework of unsupervised sentiment classification

Although theoretically a general classifier could be trained on Chinese idiom resources to model the domain-independent sentiment knowledge, the coverage and efficiency of such model are limited by the quality and quantity of idiom resources. Besides, it is obvious that the paraphrase of idiom is usually short, so the training instance for the general classifier must be very sparse, which would degrade the effect of such classifier. For above reasons, a consistent self-training bootstrapping framework of machine learning has been chosen to upgrade the performance of the general classifier. The framework is illustrated in figure 1.

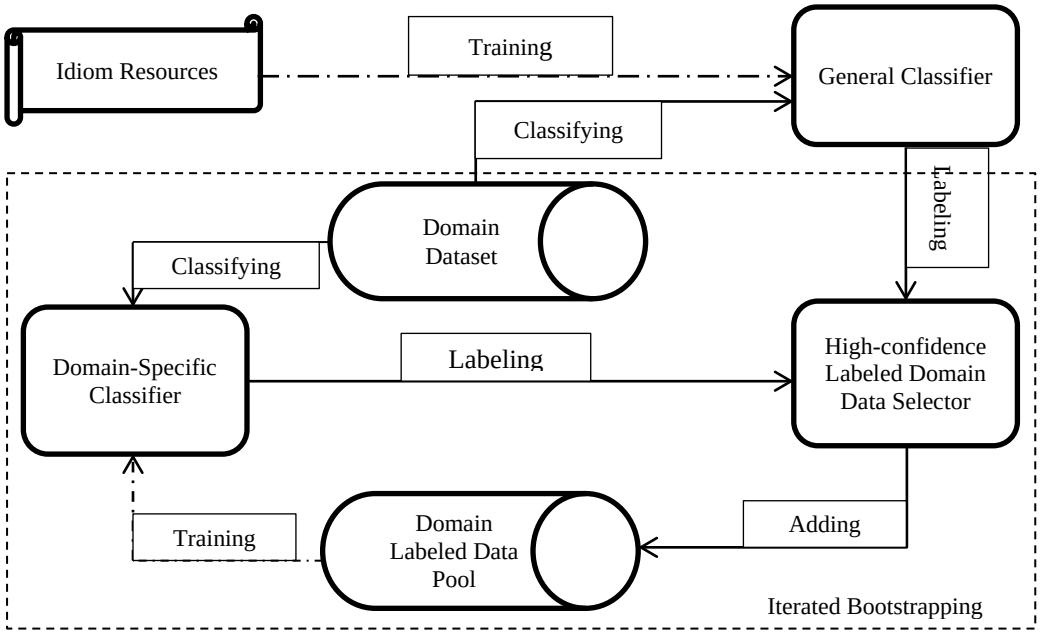

Fig. 1: Unsupervised sentiment classification Framework

As could be seen from the framework, a general domain-independent classifier is trained on the dataset extracted from idioms resources, which makes use of general sentiment part of features, and the general classifier is applied to each domain dataset to be classified so that the domain data are labeled initially. The labeled dataset are fed into the iterated convergent procedure of bootstrapping learning to train a domain-specific classifier which makes use of the domain specific part of features. At the same time the output of such unsupervised machine learning framework is result dataset of the sentiment classification. Above all, the whole framework could be segmented as two pipelining training and classifying phrases. The algorithms of two phrases are described in detail in the following section.

4.1 General classifying algorithm

The same methods as Pang et al. [4] were adopted, because they have applied Naive Bayes, Maximum Entropy (Maxent) and Support Vector Machine (SVM) classification techniques to identify the effectiveness of machine learning on sentiment classification of movie reviews, and they got satisfying result (accuracy 82.9%) using the simplest unigrams as features.

4.1.1 Naïve Bayesian classifier

Naïve Bayesian method is one of the most popular techniques for text classification. Given a set of training documents D, with each document considered as an ordered list of words, entry $w_{d_i,k}$ is used to denote the word in position k of document d_i, where each word is from the vocabulary $V = < w_1, \dots, w_{|v|} >$. The vocabulary is the set of all words considered as features for classification. A set of pre-defined classes, $C = \{c_1, c_2\}$ are used to label a document. To determine which class a document belongs to, it is needed to compute the posterior probability, $P(c_j|d_i)$, where c_j is a class label and d_i is a document. Based on the Bayesian probability and the multinomial model, the following equation is got:

$$P(c_j) = \frac{\sum_{i=1}^{|D|} P(c_j|d_i)}{|D|} \quad (3)$$

Based on the hypothesis that the probabilities of document words are independent given the class:

$$P(c_j|d_i) = \frac{P(c_j)\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_j)}{\sum_{r=1}^{|c|} P(c_r)\prod_{k=1}^{|d_i|} P(w_{d_k,k}|c_r)} \quad (4)$$

For the Naïve Bayes classifier, the class label with the highest probability is assigned as the class label of the document.

4.1.2 Maximum Entropy classifier

Maximum Entropy classifier has been widely applied in many Natural Language Processing tasks[36][37]. The classifier assigns the class label with the higher conditional probability given the document as follows:

$$P(c_j|d_i, \vec{\theta}) = \frac{1}{Z} exp(\vec{\theta}, \vec{f}(d_i, c_j)) \quad (5)$$

Where $\vec{\theta}$ is a vector of feature weights and $\vec{f}(d_i, c_j)$ is a feature function that maps pairs (d_i, c_j) to a nonnegative real-valued feature vector. Each feature has an associated parameter $\vec{\theta}_i$, called its weight; and Z is the corresponding normalization factor.

With a set of labeled documents D, Maximum Likelihood parameter estimation (training) procedure for such a model is trying so solve such a optimization problem as:

$$\vec{\theta}^* = argmax_{\vec{\theta}} \prod_{i=1}^{|D|} P(c_j|d_i, \vec{\theta}) \quad (6)$$

4.1.3 Support Vector Machines classifier

Support Vector Machines classifier, abbreviated as SVM, is a kind of discriminative method of machine learning techniques. Based on the structural risk minimization principle of the computational learning theory, SVM tries to find a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective discriminative points from the training dataset.

Multiple variants of SVM have been developed for different tasks[38], [39]. In this research, linear SVM has been adopted due to its popularity and sound performance in sentiment classification task. The optimization of SVM (dual form) is to minimize:

$$\vec{\alpha}^* = argmin\left\{-\sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j x_i x_j \langle \vec{x}_i, \vec{x}_j \rangle\right\} \quad (7)$$

Subject to: $\sum_{i=1}^{n}\alpha_i y_i = 0;\ 0 \le \alpha_i \le C$

4.2 Domain-specific classifying algorithm

As the general sentiment features are only one part of all features in the whole feature space, the domain-dependent part of features should be considered in order to capture the subtle clues embedded in the specific sentiment expressions in each domain, so that test data of each domain could be classified more accurately than just using the general part of features. Usually unsupervised approaches improve its performance by combining with supervised learning in target domain, and such ways were adopted to verify the rationality of the feature space division hypothesis. Specifically, initial label and the confidence of classification were identified for each instance of each domain after applying general classifier to the multi-domain dataset, so that a domain-specific supervised classifier could be trained on the labeled instances with high confidence under the unsupervised self-training framework. The main idea of this kind of bootstrapping method was to apply a domain-independent classifier to label each domain dataset, and the instances labeled by the general classifier with high confidence served as training data for a supervised machine learning classifier[40]. Generally speaking, the resulting domain-specific supervised classifier should be more effective in each domain than the general classifier as long as the general classifier is dependable enough. Otherwise, the performance would be inferior because too many false-labeled instances passed into iterative self-training process would bring down the domain-specific classifier. Table 1 illustrates the self-training algorithm.

5 Experiments

In this section, systematically evaluation of the proposed approaches designed for document-level sentiment classification is introduced.

Table1: self-training algorithm

Input	general classifier C_g domain test dataset U
Self-training procedure	(a) classify each domcument d_i in U with classifier C_g (b) initiate domain training dataset S b1) sort d_i according to classifying confidence b2) initiate S with s high-confidence documents b3)remove s documents from U (c) iterative self-training procedure: c1) train domain classifier C_d on dataset S c2) classify each document d_i in U with classifier C_d c3) if convergent, end the procedure c4) extend S c4-1) sort d_i according to classifying confidence c4-2) extend S with s high-confidence documents c4-3) remove s documents from U (d) get A by classifying U with the self-training classifier C_s
Output	automatically labeled domain dataset A self-training classifier C_s

5.1 Experiment description

5.1.1 Dataset

Idioms are common phenomena of Chinese, most of which embody the sentimental polarity of the users, but sentiment classification of Chinese hasn't make full use of such kind of rich language resources. So in our research of sentiment classification, the Chinese idioms became the target resources of constructing the training dataset to model the universal sentiment. Because of limited entries of most existent Chinese Idioms Dictionaries, and to make the case worse, some of entries do not contain sentiment polarity labels, the abundant idiom resources on the web labeled with collective intelligence became the main resources for constructing training dataset. After crawling from the online idiom dictionary of "China Education Network", an idiom dataset of 24,395 entries were achieved, with each entry labeled with positive, negative and neutral class. Since the research has focused on binary classification, the neutral entries were removed from the dataset. As a result, a final dataset of 8,160 instances labeled with positive and negative classes were used to train the general classifier. To evaluate the performance of general classifier and train the self-training classifier, three publicly available Chinese reviews corpus[1] of three domains (book, hotel and notebook PC) were adopted, each of which consisted of 4,000 reviews (2,000 positives and 2,000 negatives).

5.1.2 Packages and classifiers

Text written in Chinese are not well formatted in that words in a sentence are not separated by space as English. All the text in Chinese must be segmented before bag-of-words features being extracted. In the experiment, Chinese text of train and test dataset was segmented with an open source Chinese segmentation package named "Mmseg"[2]. As for classifiers, Naïve Bayes classifier and Maximum Entropy classifier of NLTK (Natural Language ToolKits) [3] package and Support Vector Machine classifier of Libsvm package[41] were used for classification. All the parameters and settings were optimized by cross-validation.

5.1.3 Feature selection

Feature selection is often used to pick out discriminating features from training dataset for efficient classification. Among various methods, the CHI statistic measures which calculates the association value between features and categories[42] was chosen, defined as:

$$CHI(t, c_i) = \frac{N \times (AD - BE)^2}{(A+E) \times (B+D) \times (A+B) \times (E+D)} \quad (8)$$

Where A is the number of times feature t and category c_i co-occur; B is the number of times t occurs without c_i; E is the number of times c_i occurs without t; D is the number of times neither c_i nor t occurs; N is the total number of documents. Features with high CHI values were selected and the last 5% low-value features were removed.

5.1.4 Baselines and upper bound

- Baselines

Two baselines were used to compare with the proposed method, the first one was naïve 50% baseline since the test corpus were all balanced with respect to the sentiment classes, the other one was the cross-domain classifier with the same algorithms and settings as the general classifier. The latter baseline was used to demonstrate the superiority of the

[1] http://www.searchforum.org.cn/tansongbo/corpus/

[2] http://code.google.com/p/pymmseg-cpp/

[3] http://nltk.org

proposed general classifier to cross-domain classifier in that not only could it be applied to any domain without any labeled training dataset but be more robust and dependable than using labeled dataset of other domains.

- Upper bound

As mentioned in introduction section, supervised machine learning methods in each domain are often setup as upper bound whose performance can be used to be challenged by semi-supervised and unsupervised methods. In the experiments, an upper bound was also setup by training supervised classifiers with the same algorithms and settings as general classifier except for the dataset settings. For each classifier, each domain dataset were split five-folded with one fifth for testing and others for training, and the performance was measured by averaging the results of five iterative computations on split dataset.

5.1.5 Performance measurement

There are various complicated measurements to evaluate the performance of computational algorithm, of which the simplest accuracy index was chosen to evaluate the performance of sentiment classifiers, because the comparison between measurements was not the important points of our research. Accuracy was defined as:

$$a = \frac{N_c}{N} \quad (8)$$

where N_c denotes number of correct predictions, and N denotes number of examples in test dataset.

5.2 Experiment Results

5.2.1 General classifier versus cross-domain classifier

Because of the imbalance of the idiom dataset with 5,611 negative instances and only 2,549 positive instances, over-sampling techniques were adopted which specifically aim to balance the class populations through replicating the minority class samples[43]. After over-sampling the positive data, a balanced training dataset of 11,222 instances were achieved. For the cross-domain classifier, three supervised classifiers were trained separately on full dataset of each domain and tested on the other two domains. The results are shown in table 2 in which NB denotes Naïve Bayes classifier, MX denotes Maximum Entropy classifier and SVM denotes Support Vector Machine classifier. From the table the following results can be observed.

Firstly, the accuracies of general classifier tested on three domains all surpass the naïve baseline (50 percent), with the least gap (11.640 percent) of Maximum Entropy classifier tested on the hotel corpus and the largest gap (30.7percent) of SVM classifier tested on the book corpus. The result proves that the general classifier is superior to random selection and may be better choice when there are no labeled dataset available for supervised or semi-supervised machine learning sentiment classification.

Table 2: Results for general vs cross-domain classifier

Domain		Book classifier	Hotel classifier	Notebook classifier	General classifier
book	***NB***		48.425	59.350	**69.995**
	MX		47.525	57.850	**63.675**
	SVM		32.500	61.225	**80.700**
hotel	***NB***	50.312		**76.344**	66.425
	MX	50.837		**76.341**	61.640
	SVM	43.685		59.665	**71.775**
notebook	***NB***	50.100	62.675		**65.100**
	MX	50.400	63.050		**70.500**
	SVM	50.075	63.250		**63.850**

Secondly, as for cross-domain classifiers, the performance of each classifier is diverse, from the worst hotel SVM classifier tested on book corpus (17.5 percent below 50%) to best notebook Naïve Bayes classifier tested on hotel corpus (26.344 percent above 50%). The result shows inequality of cross-domain classifiers and the reason will be discussed in the conclusion and discussion section. Besides, labeled dataset are needed for training cross-domain classifier despite of dataset out of domain.

Finally, for comparison between two kinds of classifier, the general classifier outperforms the cross-domain classifier with 7 highest accuracies versus 2 highest accuracies marked boldly in table 2. It is obvious enough that the average accuracy of general classifier is much higher than the cross-domain classifier, and more stable. In addition to the above observation, no laborious labeled data are needed for training the general classifier.

Above all, it is more robust and dependable for general classifier than cross-domain classifier.

5.2.2 Domain-specific versus self-training classifier

Three supervised domain-specific classifiers were trained in five-folded in cross-validation mode and the average accuracies were reported as the upper bound for unsupervised self-training classifier, and as illustrated in table 2, the general SVM sentiment classifier has showed the best performance of all three general classifiers, so according to algorithm 1 described in last section, the SVM classifier were used to activate the bootstrapping procedure by outputting high-confidence instances to a self-training classifier. Table 3 illustrates the result.

Table 3: Results for in-domain and self-training classifier

	NB	MX	SVM	Self-training
book	55.087	57.487	**93.375**	86.200
hotel	72.407	77.910	87.479	**87.521**
notebook	90.337	90.212	**90.700**	81.500

From table 3, it is provable of the result that the SVM classifier performs best of three domain-specific classifiers in accordance with

conclusion of other researchers[4]. The improvement of performance from the general classifier to the self-training classifier is undoubtedly outstanding after the bootstrapping training procedure in each domain, with 5.5 percent improvement in book domain, 15.146 percent improvement in hotel domain and 17.65 percent improvement in notebook domain. But only in hotel domain, does self-training classifier surpass all three domain-specific classifiers. In book and hotel domain the self-training classifier also surpass domain-specific Naïve Bayes and Maximum Entropy classifiers. As a whole, the performance of the self-training classifier based on general classifier approximates to the upper bound domain-specific classifier, which demonstrates the efficiency of the proposed framework.

6 Discussion and future work

In the proposed unsupervised framework, a self-training classifier which takes the high-confidence predictions of the general classifier as input is trained in each domain to exploit domain-dependent part of features, and the performance of which is much like the upper bound domain-specific classifier. Obviously the output quality of general classifier has a critical impact on the performance of the self-training classifier. Usually, the more accurate the prediction of the general classifier is, the better the resulting performance of self-training classifier is.

Therefore, further performance improvement of general classifiers is one of target future works, and there are some ways to do so. First of all, larger domain-independent language resources are needed to improve the coverage of features and reduce sparseness. Table 4 illustrates feature coverage among three domains and coverage between idioms dataset and each domain. It can be observed that the coverage of idiom features is relatively much less compared with three domains, so improvement of feature coverage would improve performance of general classifier. Sparseness is another factor that influence performance, according to the statistic, the paraphrase of each idiom is very short (19 characters of mean length) and the abandonment of words caused by segmenting error also causes sparseness.

Combining results of table 4 and table 2, it can be analyzed that, although some coverage among three domains is much larger than the idiom features, the cross-domain classification performance is worse than the general classifier. The reason lies in that the different distribution of common part of features in different domain will make cross-domain classifier unsuitable in another domain, while the distribution of common part between idiom features and other three domains features is independent of domains according to two assumptions discussed above.

Table 4: Statistic of feature coverage

	book	**hotel**	**notebook**	**idiom**
Book (20219)		0.413 (8365)	0.243 (4933)	0.358 (7251)
Hotel (16220)	0.515 (8365)		0.307 (4994)	0.329 (5345)
Notebook (7595)	0.649 (4933)	0.657 (4994)		0.408 (3100)

In addition, only unigrams have been used to model the sentiment polarity and used to train the polarity classifiers, while many researches in machine learning sentiment classification suggest that more advanced linguistic modeling is likely to improve the performance of sentiment classification because complicated human sentiment could not be fully modeled by simple linguistic features. So the next future work will focus on mining more useful linguistic features to improve performance of general classifier.

7 Conclusions

1) With the need of sentiment classification in various domains, domain dependency has become the bottleneck of machine learning techniques for the shortage of labeled domain dataset. The problem has been solved by a novel unsupervised learning framework on the off-the-shelf domain-independent resources.

2) A novel perspective of sentimental features is put forward with the assumption that feature space be divided into the general part and the specific part. Elicited by human's identification of sentiment polarity, which suggests the general features play important role in sentiment classification, a general classifier is proposed to make use of the general part features. The general classifiers can be obtained by training on linguistic resources independent of domains such as Chinese Idioms resources.

3) The performance of general classifier can be improved by adopting bootstrapping procedure in the target domain to make use of the domain-specific part features. Thus a self-training classifier is got by taking high-confidence predictions of the general classifier as initial input to an iterative training procedure. The performance of self-training classifier can approximate the upper bound supervised learning domain-specific classifier.

References

[1] MILLER M, SATHI C, WIESENTHAL D, LESKOVEC J, POTTS C. Sentiment Flow through hyperlink networks[C]. Proceedings of the Fifth international aaai conference on Weblogs and Social media, 2011: 550-553.

[2] PANG Bo, LEE L. Opinion mining and sentiment analysis[J]. Foundations and Trends in Information Retrieval, 2008, vol. 2, no. 1–2: 1–135.

[3] AGARWAL A, BHATTACHARYYA P. Sentiment Analysis: A New Approach for Effective Use of Linguistic Knowledge and Exploiting Similarities in a Set of Documents to be Classified[C]. Proceedings of the International Conference on Natural Language Processing (ICON), 2005:238-247 .

[4] PANG Bo, LEE L, VAITHYANATHAN S. Thumbs up? Sentiment Classification Using Machine Learning Techniques[C] Proceedings of *emnlp2002*, 2002: 79–86.

[5] B. Yang, “Semi-supervised Learning for Sentiment Classification,” pp. 1–8.

[6] S. Owsley, S. Sood, and K. J. Hammond, “Domain Specific Affective Classification of Documents,” in *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, 2006, pp. 181–183.

[7] V. Ng, S. Dasgupta, and S. Arifin, “Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews,” in *Proceedings of the COLING/ACL on Main conference poster sessions*, 2006, vol. pp, pp. 611–618.

[8] A. Stepinski and V. Mittal, “A fact/opinion classifier for news articles,” in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 807–808.

[9] E. Boiy and M.-F. Moens, “A machine learning approach to sentiment analysis in multilingual Web texts,” *Information Retrieval*, vol. 12, no. 5, pp. 526–558, 2008.

[10] O. T\"ackström and R. McDonald, “Discovering fine-grained sentiment with latent variable structured prediction models,” *Advances in Information Retrieval*, pp. 368–374, 2011.

[11] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, “Target-dependent Twitter Sentiment Classification,” in *Computational Linguistics*, 2011, pp. 151–160.

[12] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar, “Structured Models for Fine-to-Coarse Sentiment Analysis,” in *Proceedings of the Association for Computational Linguistics (ACL)*, 2007, pp. 432–439.

[13] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis,” in *Proceedings of the ACL (To appear)*, 2011.

[14] S. Somasundaran, G. Namata, L. Getoor, and J. Wiebe, “Opinion Graphs for Polarity and Discourse Classification,” in *Proceedings of the 2009 Workshop on Graphbased Methods for Natural Language Processing TextGraphs4*, 2009, pp. 66–74.

[15] O. Täckström, “Semi-supervised latent variable models for sentence-level sentiment analysis,” in *Computational Linguistics*, 2011, pp. 569–574.

[16] N. Yu and S. Kubler, “Semi-supervised Learning for Opinion Detection,” in *2010 IEEEWICACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp. 249–252.

[17] V. Sindhwani and P. Melville, “Document-Word Co-regularization for Semi-supervised Sentiment Analysis,” *2008 Eighth IEEE International Conference on Data Mining*, pp. 1025–1030, 2008.

[18] A. B. Goldberg and J. Zhu, “Seeing stars when there aren’t many stars: Graph-based semi-supervised learning for sentiment categorization,” in *TextGraphs: HLT/NAACL Workshop on Graph-based Algorithms for Natural Language Processing*, 2006.

[19] S. Tan, G. Wu, H. Tang, and X. Cheng, “A novel scheme for domain-transfer problem in the context of sentiment analysis,” *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management CIKM 07*, no. 2, p. 979, 2007.

[20] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, “Cross-domain sentiment classification via spectral feature alignment,” in *Proceedings of the 19th international conference on World wide web WWW 10*, 2010, p. 751.

[21] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, “Knowledge transformation for cross-domain sentiment classification,” in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval SIGIR 09*, 2009, pp. 716–717.

[22] B. Lu, C. Tan, C. Cardie, and B. K Tsou, “Joint Bilingual Sentiment Classification with Unlabeled Parallel Corpora,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 2011, pp. 320–330.

[23] R. Mihalcea, C. Banea, and J. Wiebe, “Learning multilingual subjective language via cross-lingual projections,” in *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, 2007, vol. 45, no. 1, p. 976.

[24] N. Yu and S. Kübler, “Filling the Gap: Semi-Supervised Learning for Opinion Detection Across Domains,” in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 2011, pp. 200–209.

[25] S. Brody and N. Elhadad, “An Unsupervised Aspect-Sentiment Model for Online Reviews,” *Human Language Technologies The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, no. June, pp. 804–812, 2010.

[26] T. Zagibalov and J. Carroll, “Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text,” *Proceedings of the Conference on Computational Linguistics COLING 2008*, no. August, pp. 1073–1080, 2008.

[27] P. D. Turney, “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, no. July, pp. 417–424.

[28] S. Tan, Y. Wang, and X. Cheng, “Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples,” *Proceedings of the 31st*

annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 08, p. 743, 2008.

[29] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *CIKM 09 Proceeding of the 18th ACM conference on Information and knowledge management*, 2009, pp. 375–384.

[30] A. Kennedy and D, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[31] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in *Proceedings of the 14th ACM international conference on Information and knowledge management CIKM 05*, 2005, p. 625.

[32] J. Liu and S. Seneff, "Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009, no. August, pp. 161–169.

[33] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC'10)*, 2010, vol. 25, pp. 2200–2204.

[34] P. J. Stone, *The General Inquirer: A Computer Approach to Content Analysis*. M.I.T. Press; First Edition edition (January 1, 1966), 1966, p. 651.

[35] L.-W. Ku, Y.-T. Liang, and H.-H. Chen, "Opinion Extraction, Summarization and Tracking in News and Blog Corpora," in *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, 2006, pp. 100–107.

[36] A. L. Berger, S. A. Della, V. J. Della, and V. J. D. Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, no. 1992, pp. 1–36, 1996.

[37] S. Li, H. He, W. R. Xu, and J. Guo, "Automatic Chinese Sentiment Word Extraction Based on Maximum Entropy," *Proceedings of 2009 International Conference on Wavelet Analysis and Pattern Recognition*, pp. 437–441 501, 2009.

[38] D. Niu, J. Wang, and J. Liu, "Knowledge mining collaborative DESVM correction method in short-term load forecasting," *Journal of Central South University of Technology*, vol. 18, no. 4, pp. 1211–1216, 2011.

[39] Y. Sun, V. Werner, and X. Zhang, "A robust feature extraction approach based on an auditory model for classification of speech and expressiveness," *Journal of Central South University*, vol. 19, no. 2, pp. 504–510, 2012.

[40] H. Li, P. Li, Y. Guo, and M. Wu, "Multi-label dimensionality reduction based on semi-supervised discriminant analysis," *Journal of Central South University of Technology*, vol. 17, no. 6, pp. 1310–1319, 2010.

[41] C.-C. Chang and C.-J. Lin, "{LIBSVM}: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[42] L. Galavotti, F. Sebastiani, and M. Simi, "Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization Research and Advanced Technology for Digital Libraries," vol. 1923, J. Borbinha and T. Baker, Eds. Springer Berlin / Heidelberg, 2000, pp. 59–68.

[43] A. Blum and S. Chawla, "Learning from Labeled and Unlabeled Data Using Graph Mincuts," 2001, pp. 19–26.