Combining Lexicon and Learning based Approaches for Concept-Level Sentiment Analysis

Andrius Mudinas
DCSIS
Birkbeck, University of London
London WC1E 7HX, UK
andrius@dcs.bbk.ac.uk

Dell Zhang
DCSIS
Birkbeck, University of London
London WC1E 7HX, UK
dell.z@ieee.org

Mark Levene
DCSIS
Birkbeck, University of London
London WC1E 7HX, UK
mark@dcs.bbk.ac.uk

ABSTRACT

In this paper, we present the anatomy of *pSenti* — a concept-level sentiment analysis system that seamlessly integrates into opinion mining lexicon-based and learning-based approaches. Compared with pure lexicon-based systems, it achieves significantly higher accuracy in sentiment polarity classification as well as sentiment strength detection. Compared with pure learning-based systems, it offers more structured and readable results with aspect-oriented explanation and justification, while being less sensitive to the writing style of text. Our extensive experiments on two real-world datasets (CNET software reviews and IMDB movie reviews) confirm the superiority of the proposed hybrid approach over state-of-the-art systems like *SentiStrength*.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; I.2.6 [**Artificial Intelligence**]: Learning; I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

General Terms

Algorithms, Experimentation, Performance

Keywords

Opinion Mining, Sentiment Analysis, Natural Language Processing, Supervised Learning.

1. INTRODUCTION

Everyday a large number of opinion related documents are put on the Internet – people post product reviews, express their political views, and share their feelings. The ability to extract sentiments from such sources can provide invaluable information about people's views on various topics.

Many of today's sentiment analysis systems are based on so-called lexicon design, having domain-specific sentiment lexicons as their main sentiment information source [6, 20, 21]. Such an approach is usually implemented in two separate steps: lexicon detection/extension and sentiment strength measurement. On the other hand sentiment detection can be treated as a simple classification problem and achieve very high accuracy by employing various machine learning algorithms, such as Naïve Bayes or Support Vector Machine (SVM). Yet simple classification provides limited information about sentiment topic or rationale.

In this paper, we present the anatomy of *pSenti* — a concept-level sentiment analysis system that seamlessly integrates into opinion mining lexicon-based and learning-based approaches. The main idea is to generate the feature vectors for supervised machine learning in the same fashion as is seen in lexicon-based sentiment analysis systems. Compared with pure lexicon-based systems, it achieves significantly higher accuracy in sentiment polarity classification as well as sentiment strength detection. Compared with pure learning-based systems, it offers more structured and readable results with aspect-oriented explanation and justification, while being less sensitive to the writing style of text. The ability to perform cross-style sentiment analysis is very meaningful, as it implies that we can train the system using formal professional reviews as training examples and then apply the system to sentiment analysis on informal customer reviews from data sources such as blogs or twitter. Our extensive experiments on two real-world datasets (CNET software reviews and IMDB movie reviews) have confirmed the superiority of the proposed hybrid approach over state-of-the-art systems like *SentiStrength* [20, 21].

The rest of this paper is organised as follows. In Section 2, we review the related work. In Section 3, we present our *pSenti* system based on the hybrid approach in details. In Section 4, we show the experimental results on two real-world datasets. In Section 5, we make conclusions.

2. RELATED WORK

In recent years, opinion mining, aka sentiment analysis, attracted a lot of interest and has been studied by many researchers. In their early work, Hatzivassiloglou and McKeown [7] reported that it is possible to identify sentiment words (adjectives) and their polarity in sentences with a high accuracy of 82%. Following this finding, various sentiment analysis algorithms have been proposed. For example, Turney [22] introduced one of the first algorithms for document level sentiment analysis, which achieved an average accuracy of 74% for product reviews; but on movie reviews the performance was much worse – only 66%. In his design, rather than focusing on isolated adjectives, Turney proposed to de-

WISDOM'12, August 12, 2012, Beijing, China.

tect sentiments based on selected phrases, which are chosen via a number of Part-Of-Speech (POS) patterns. Generally speaking, POS information is frequently exploited in sentiment analysis systems. In particular, POS tagging helps with the word sense disambiguation problem and provides the ability to better understand the surrounding context. For another example, Cambria et al. proposed Sentic Computing which explores the usage of Common Sense Computing to significantly enhance computers' emotional intelligence, i.e., their capability of perceiving and expressing emotions [3–5].

As it stands, the design of sentiment analysis systems could be divided into two schools — a lexicon-based approach [6,20,21] and a learning-based approach [2,11,12,14, 22]. Pang et al. [15] have evaluated and compared several different supervised machine learning algorithms for classifying the sentiments of movie reviews. The learning algorithms they used include Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector Machine (SVM), with SVM slightly outperforming other learning algorithms. In their early work [15], they achieved 82.9% accuracy with a relatively simple design using SVM trained on bag-of-words (unigram) features; this was further increased in their later work [14] to 87.2%. This performance increase was achieved by employing graph min-cut based subjectivity detection before the classification step, thus removing objective text from the final sentiment classification. However, as other researchers have found [17], such a simple design, solely based on supervised machine learning, suffers from style, domain, or even time dependencies. Furthermore, it only provides an overall sentiment score for each review without any further explanation or justification. People often express multiple opinions in their reviews, therefore by just detecting that a given review is positive or negative we cannot obtain much knowledge about which specific aspects (e.g., product features) people liked or disliked and to what degree. To address these concerns, researchers have proposed various sentence-level opinion mining techniques. For example, Hu and Liu [9] proposed a two-step method for sentence-level sentiment analysis, which was later improved by Popescu and Etzioni [16]. Using the two-step method, sentiment analysis could be understood as two separate tasks — aspect identification and sentiment strength measurement for each aspect. The step of aspect identification has very important practical value, as the aspects establish the areas in which the sentiment was expressed. Extracting aspects from a "high-quality" text is usually a quite straightforward procedure. As Hu and Liu [9] have found, that could be done by simply selecting frequent nouns and noun phrases. However, customer reviews are usually short, informal, and sometimes even ungrammatical (e.g., consisting of incomplete sentences), which makes this task very challenging. To overcome this problem some researchers proposed to use labelled sequential rules (LSR) [10], where such a rule is essentially a special kind of sequential pattern. Until recently most sentiment detection algorithms were based either purely based on lexicon or learning. In the rare case of mixed systems, lexicon or learning was employed only for some minor sub-task — like extending the sentiment lexicon [23] or identification of subjective text blocks [14]. In recent years, some attempts were made to incorporate lexicon knowledge into machine learning classifiers [1, 8, 19], especially with increasing popularity of various generative probabilistic models based on Latent Dirichlet Allocation (LDA) [8, 11, 12]. However, their sentiment analysis performances were often worse than the simple bag-of-words SVM approach [15].

A recent hot topic in opinion mining is domain dependency. As Owsleys et al. [13] have found, to achieve good sentiment analysis results you have to build a domain-specific lexicon that is related to both the entities and their sentiment expressions. Particularly, in different domains, the same word could have completely opposite meanings or very different sentiment strengths. To build domain-specific lexicons, researchers have proposed various techniques. One common approach is to start from a small initial sentiment lexicon and gradually expand it during the processing of reviews. Some researchers have successfully utilised WordNet for the construction of sentiment lexcions [9], where WordNet provides initial seed information for the sentiment lexicon which will then be expanded using a sentiment corpus [6]. However, our experience is that WordNet is not a very reliable source to build sentiment lexicons, since it introduces too much noise. Furthermore, it is worthy to mention that their method does not adjust the sentiment value for each sentiment word in the lexicon — it merely expands the lexicon with previously unknown sentiment words. Another common approach is bootstrapping. For example, Riloff and Wiebe [18] employed a classifier to extract subjective patterns from text which could be used to build a sentiment lexicon.

3. APPROACH

Our concept-level sentiment analysis system, *pSenti*, is developed by combining lexicon-based and learning-based approaches. As shown in Figure 1, the supervised machine learning component is not just responsible for small tasks such as adjusting sentiment values or finding more sentiment words, but is actually responsible for evaluating all the ingredients of the sentiment system, including semantic rules used to derive the final output. The final component in *pSenti* measures and reports the overall sentiment of a given opinionated text, such as a customer review, as a real-valued score between -1 and $+1$, which can then be easily transformed into a positive/negative classification or into a scale of 1-5 stars.

The main advantage of our hybrid approach using a lexicon/learning symbiosis, is to attain the best of both worlds — stability as well as readability from a carefully designed lexicon, and the high accuracy from a powerful supervised learning algorithm. Due to the built-in sentiment lexicon and linguistic rules, *pSenti* can detect and measure sentiments at the concept level, providing structured and readable aspect-oriented outputs, as illustrated by Figure 2. Furthermore, *pSenti* is less sensitive to changes in topic domain or writing style. The system can even be extended after it has already been trained, by introducing new linguistic rules or expanding the sentiment lexicon at any time, so as to further improve the system's performance.

3.1 Preprocessing

At the first step, we use the Stanford CoreNLP[1] toolkit to carry out POS and entity tagging. Prior to feeding a piece of text into the Stanford parser we perform some simplification

[1] `http://nlp.stanford.edu/software/corenlp.shtml`

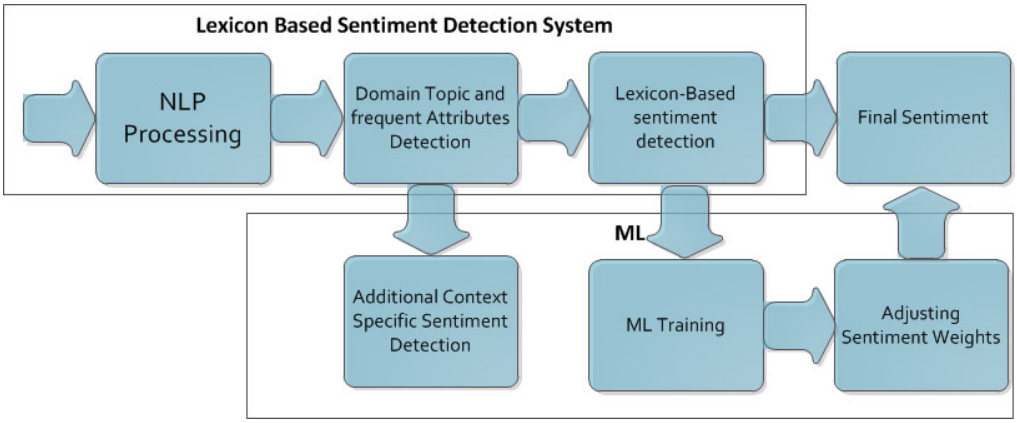


Figure 1: The system architecture of *pSenti*.

Customer Review => { (Aspect$_1$: View$_1$), (Aspect$_2$: View$_2$), ..., (Aspect$_k$: View$_k$) }, e.g., A user comment on Google Chrome => { (Appearance: +0.8), (Plugins: +0.6), ..., (Speed: +0.9) }.

Figure 2: An example of *pSenti*'s aspect-oriented output.

of the text. Specifically, we replace known idioms and emoticons with text masks. For example, if our dataset shows that the emoticon ":-)" has a positive sentiment strength +1, it will be replaced by the system-defined pseudo-word `_Good_One_`; and similarly, ":|, which has a negative sentiment strength −1, will be replaced with the system-defined pseudo-word `_Bad_One_`. The assumption here is that various emoticons express similar sentiment strength, which have already been measured and differentiated, so it would be redundant to generate a separate feature for each of them that will be later used by the supervised machine learning algorithm. Such heuristic rules also apply to idioms. Thus "crocodile tears", known to have sentiment strength −3, should be replaced by `_Bad_Three_`. The range of sentiment values is is from -1 to +1 for emoticons and from -3 to +3 for for idioms. Currently *pSenti* knows 116 emoticons and 40 idioms.

3.2 Aspect and View Extraction

People very often express multiple views (sometime even of opposite polarity) about different aspects of the same item in a single review, such as a a software product or a movie. Therefore, it is very important for a practical sentiment analysis system to extract the discussed aspects and the corresponding views from each document having sentiments.

The current implementation of *pSenti* uses a simple aspect and view extraction algorithm as follows:

- **Find candidate aspects and views**. We generate a list of candidate aspects by including nouns and noun phrases identified by the POS tagger as well as organisations identified by the entity tagger, but excluding all stop words, other types of entities (names and locations) and known sentiment words. We generate a list of candidate views by including adjectives and known sentiment words which occur near an aspect (in one sentence), but excluding all stop words and all types of entities.
- **Clean-up**. We further remove all candidate aspects or views that occur less than 5 times, and the aspects which have already been detected as top views.
- **Cluster similar aspects**. We cluster similar aspects into aspect groups using their lexical similarity.
- **Generate final aspects and views**. The final list includes only the top 100 grouped aspects, the top 100 views, plus the top 10 views for each selected aspect.

Another motivation for *pSenti* to emphasize aspect/view extraction, is that domain-specific aspect words will be excluded from the machine learning step in order to reduce the dependence on the current topic domain, writing style, or time period. For example, in many of the browser category customer reviews, we can clearly observe very negative sentiments towards "Internet Explorer" and "Microsoft", so if we include these words in the machine learning step they would be given high negative values. In the pure learning-based approach (using SVM as the learning algorithm) "Microsoft" would be in the top list with a strong negative weight of −1.36, and "Firefox" would have a strong positive weight of +1.07. However, it is clear that these words do not really carry any stable or robust sentiment value, and it is purely a coincidence that at the time of sentiment analysis Microsoft IE6 was having a lot of negative publicity. After a couple of years, we might find that the sentiment polarity and strength for these aspect words have become totally different from their current values.

In addition, aspect/view extraction allows us to find frequently occurring adjectives (views) which can be used to expand the sentiment lexicon, and also enables us to perform context-aware sentiment value estimation for such adjectives in the given aspect. For example, the same word "large" could have very different sentiment implications in different contexts: the sentiment for a "large monitor" is usually positive, while the sentiment for a "large phone" is probably negative.

3.3 Lexicon-based Sentiment Detection

Our system uses a sentiment lexicon constructed using public resources for initial sentiment detection. Currently the sentiment lexicon consists of 7048 sentiment words including words with wildcards. Their sentiment values are marked in the range from -3 to $+3$. All initial sentiment values and lexicon constants (negation, modifiers) are based upon the authors' judgement and experimentation results during the development stage. However, our experimental results show that the initial values have only minor influence on the final *pSenti* performance, as the machine learning stage is able to detect human bias and adjust those heuristic values. On the basis of the sentiment lexicon, we further apply the following heuristic linguistic rules to detect sentiments from text.

- **Negation**. We included both traditional negation words such as "not" and "don't" as well as pattern-based negations such as "stop *vb*-ing" and "quit *vb*-ing". Our system also employs an algorithm in which negation could be applied to more distant sentiments. If a negation word could not be attached to a sentiment or another known adjective it is treated as a negative sentiment word with weight -1.5, and will generate the feature `_Not_` for the machine learning algorithm. As part of the processing, we perform various sentence repairs using heuristic rules for more reliable negation detection. For example, the system detects negation words in phrases like "not just ..." and "not only ... but also", and exclude them as sentiment negations. In addition, it splits words with the "non-" prefix, e.g., the word "non-violent" will be separated into two words "not violent" in advance.

- **Modifier**. Comparative adjectives and adverbs (e.g., "more", "less"), intensifying adverbs (e.g., "very", "absolutely"), diminishing adverbs (e.g., "little", "somewhat"), and some other words can increase or decrease the sentiment value of their associated sentiment value by several fold. Currently we have 75 such modifiers with their impact value in the range from 0.4x to 2.5x.

3.4 Learning-based Sentiment Evaluation

The supervised machine learning algorithm used in our system is the linear SVM implementation in LibSVM[2] with L2 objective function for optimisation and grid-search for parameter tuning. We chose linear SVM as it has been shown to outperform other popular learning algorithms for sentiment analysis in previous studies [15].

3.4.1 Feature Extraction

In their work Pang and Lee [14] found that detecting and excluding objective text from reviews could significantly improve sentiment detection performance. However, in the current *pSenti* implementation, we use only very basic subjectivity detection: the feature vectors are generated only for reviews in which the lexicon-based algorithm was able to detect sentiment presence. In the future we are going to develop and apply a more advanced subjectivity detection algorithm.

The feature vector for each aspect consists of the following elements:

- **Sentiment words**. The weight of such a feature is the sum of the sentiment value in the given review. For example, if we have a review with the word "good" appearing twice, which has sentiment value $+2$, we would add the feature `_Good_` with a weight of $+4$. In addition, in the case of sentiment value modification, the feature and value generation is slightly more complicated. If the sentiment source has been inverted we generate a new feature with a `_Not_` prefix and inverted sentiment value. In case of the word "good", the feature `_Not_Good_` would have the value -2. A similar situation arises with intensifiers or diminishers, e.g. for the bigram "extremely good", where we have a sentiment word "good" appearing next to one of the strongest intensifiers with x2.5 impact value, we would generate the `_More_Good_` feature with $+5$ value. In the case of "sometimes good" we would generate `_Less_Good_` with the value $+1.33$

- **Other adjectives**. For those adjectives which are not in our sentiment lexicon, we just use their occurring frequencies as their initial values, and let their true sentiment values be estimated by the learning algorithm. For example, if the word "large" appears twice we would have the feature `_Large_` with value $+2.0$. In this case, a negation, intensifier, or diminisher does not modify the feature's weight but only triggers the generation of a new feature.

- **Lexicon based sentiment score**. This feature can cater for sentiment values of the sentiment words that were previously unseen in the training examples but exist in the test examples.

As an example of feature vector generation for an aspect with the lexicon-based sentiment calculated as $+0.5$, and 3 occurrences of the adjective "long" (with sentiment value $+1.0$), two occurrences of the sentiment word "good" (with sentiment value $+2$), one occurrence of the sentiment word "bad" (with sentiment value -2), we would generate the feature vector as $[+0.5, +3.0, +4.0, -2.0]$.

3.4.2 Sentiment Measurement

After the SVM model is trained, we can reuse the calculated feature weights to adjust the final sentiment calculation. To illustrate the calculation process we will now analyse a hypothetical scenario, in which, to simplify all calculations, we will not normalise the values; see Table 1. Let us assume that according to the training data our system has calculated the feature weights, and now will process a review with having the features `_Good_`, `_Good_One`, `_Excellent_`, `_Not_`, and `_Large_`.

- **Lexicon based Strength Calculation**. In this step, each feature's weight (sentiment strength) is just a product of its sentiment value and its occurring frequency. We ignore the features which do not carry any sentiment information. For example, the feature `_Large_` is excluded from this calculation. The feature `_Not_` is a special case, and its sentiment is calculated using one of the sentiment calculation rules, i.e., if negation is not attached to a sentiment word it carries the default -1.5 sentiment strength. In this way, the overall lexicon-based (sentiment) score of this example review is calculated as $+0.32$.

[2] `http://www.csie.ntu.edu.tw/cjlin/libsvm/`

- **Learning based Weight Adjustment**. In this step we adjust previously calculated sentiment values by their SVM coefficients. The feature `_Large_` has the negative SVM weight -0.1 which is multiplied by its occurring frequency. As the feature `_Excellent_` is previously unseen in training dataset we will assign it the standard 0.5 unknown feature coefficient. In addition, we also include the previously calculated overall lexicon-based (sentiment) score, and the SVM hyperplane bias $+0.42$. Thus the final sentiment value we get after the adjustment is $+0.22$.

3.5 Final Overall Sentiment Scoring

Although most of our experimental results are reported in terms of semtiment polarity classification into positive and negative classes (see Section 4), the actual output of *pSenti* is a real-valued sentiment score in the range of $[-1, +1]$, which is first calculated using the following equation

$$S_{senti} = \frac{1}{2} \log_2 \frac{pos}{neg} \ , \tag{1}$$

where *pos* and *neg* are overall positive and negative sentiment scores respectively. The overall sentiment score is then upper-bounded by $+1$ or lower-bounded by -1 when the value is out of range. If neither positive nor negative sentiment has been detected, our algorithm treats such text as objective text and assigns it the sentiment value 0. The sentiment value can be easily transformed into a five-star scale using the simple formula

$$S_{stars} = 2 \cdot S_{senti} + 3 \ . \tag{2}$$

4. EXPERIMENTS

4.1 Datasets

To empirically evaluate the *pSenti* system, we conducted experiments on two real-world datasets.

- The first dataset — Software Reviews[3] — consists of software reviews collected in 2011 by the first author from CNET's software download website. The dataset includes five software product categories: Browser, Antivirus, Video, Action Games, and Utilities. Most software reviews are written by customers (normal users), but there are some which are written by professionals (CNET editors).
- The second dataset — Movie Reviews[4] — consists of movie reviews collected by Pang and Lee [14] from the IMDB website. It is a well-known standard benchmark dataset for sentiment analysis.

The datasets have been pre-processed to remove duplicates, spam, and inconsistencies. The detailed characteristics of those datasets are shown in Table 2.

For sentiment polarity classification tasks where only five-star ratings are available, we obtain the ground-truth class labels by considering 1-star or 2-star reviews as negative, 4-star or 5-star reviews as positive, and discarding 3-star neutral reviews.

[3] `http://www.dcs.bbk.ac.uk/~andrius/psenti/`
[4] `http://www.cs.cornell.edu/People/pabo/movie-review-data/`

4.2 Systems

The *pSenti* system, based on the proposed hybrid approach, is compared with the following baselines:

- *LexiconOnly*: The pure lexcion-based approach using the same sentiment lexicon as *pSenti*;
- *LearningOnly*: The pure learning-based approach using the same learning algorithm (linear SVM) as *pSenti*, with bag-of-words features;
- *SentiStrength*[5]: a state-of-the-art sentiment analysis system free for academic research [20, 21].

In all cases, the final sentiment polarity of each review would be determined by the sign of its sentiment score calculated using equation (1); and the final sentiment strength of each review would be given by equation (2). For *LexiconOnly* and *SentiStrength* experiments, we have used the default configurations without any training or sentiment value adjustments.

All the following experimental results are reported using 10-fold cross validation.

4.3 Results

4.3.1 Standard Setting

Sentiment Polarity Classification

Table 3 shows the experimental results of sentiment polarity classification (into positive and negative classes) in the standard (single-style) setting, where the performance is measured by classification accuracy.

As we can see, our *pSenti* system based on the proposed hybrid approach achieved very good performance on all datasets: the accuracy of *pSenti* is slightly lower than that of *LearningOnly*, but significantly higher than that of *LexiconOnly*; and obviously *pSenti* works much better than *SentiStrength*.

The performance of *pSenti* on customer software reviews is not as good as on editor software reviews. This is understandable because customer software reviews are usually much noisier than professional prepared editor software reviews. Some customers give software ratings that are inconsistent with their reviews: they may write a quite positive review but assign it only a 1-star or 2-star rating. Moreover, it is not uncommon to find reviews in which customers express opposite sentiment towards competing products. For example, in our software product reviews dataset there is a 5-star review with the sentence, "Glad to dump (Internet) Explorer forever!", where the customer clearly expresses negative sentiment towards "(Internet) Explorer", but the review has a 5-star rating because it is posted for the "Firefox" browser.

The performance of *pSenti* is the lowest on all datasets. After manually inspecting its results, we think that *pSenti*'s accuracy has been severely affected by a large number of reviews for which it failed to detect any sentiment or assigned neutral sentiment scores.

Sentiment Strength Detection

Table 4 shows the experimental results of sentiment strength detection (i.e., predicting the five-star ratings) in the standard (single-style) setting, where the performance is measured by Root Mean Squared Error (RMSE). For

[5] `http://sentistrength.wlv.ac.uk/`

Table 1: An example of sentiment strength adjustment using SVM coefficients.

Feature	Sentiment Value	Frequency	Feature Weight	SVM Coefficient	Learning based Score
`_Good_`	+2.00	+2.00	+4.00	+1.50	+6.00
`_Bad_Three_`	−3.00	+1.00	−3.00	+1.70	−5.10
`_Excellent_`	+3.00	+1.00	+3.00	+0.50	+1.50
`_Not_`	−1.50	+1.00	−1.50	−0.50	−0.50
"large"	+1.00	+2.00	+2.00	−0.10	−0.20
Lexicon based Score	—	—	+0.32	+2.00	+0.64
SVM Bias Term	—	—	+1.00	+0.10	+0.10
Total	—	—	—	—	+2.44

Table 2: The experimental datasets.

Dataset		Labels of Reviews	Number of Reviews	Avg Size of Reviews
Software Reviews	Miscellaneous (Editor)	Pos/Neg	1660	1056.82
	Browser (Editor)	Pos/Neg	360	1091.61
	Browser (Customer)	Pos/Neg	2000	158.07
	Antivirus (Customer)	Pos/Neg	2000	165.06
	Video (Customer)	Pos/Neg	2000	152.43
	Action Games (Customer)	Pos/Neg	2000	136.21
	Utilities 1 (Customer)	Pos/Neg	2000	155.80
	Utilities 2 (Customer)	1-5 Stars	1850	295.19
Movie Reviews	Movies 1	Pos/Neg	2000	3892.96
	Movies 2	1-5 Stars	5000	2257.44

Table 3: The sentiment polarity classification performance (accuracy) in the standard (single-style) setting.

Dataset		*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Software Reviews	Miscellaneous (Editor)	89.64%	79.40%	90.78%	64.93%
	Browser (Editor)	86.94%	76.94%	91.39%	62.77%
	Browser (Customer)	79.60%	74.50%	80.54%	52.25%
	Antivirus (Customer)	78.55%	70.60%	82.91%	47.85%
	Video (Customer)	83.55%	75.95%	85.83%	52.80%
	Action Games (Customer)	78.75%	71.55%	82.92%	58.25%
	Utilities 1 (Customer)	78.80%	73.70%	82.03%	50.50%
Movie Reviews	Movies 1	82.30%	66.00%	86.85%	60.70%

LearningOnly experiments, the one-vs-one ensemble method has been used to achieve 5-class classification.

As we can see, our *pSenti* system based on the proposed hybrid approach worked quite well and again it significantly outperformed *SentiStrength* on all datasets.

4.3.2 Cross-Style Setting

This part of experiments illustrate one of the main advantages of our proposed hybrid approach over the pure learning-based approach — very little style dependency.

To evaluate the cross-style sentiment analysis performance, we took one set of software reviews for training, and then test the system performance on another set of software reviews which are written in a different style, i.e., to train on editor software reviews and test on customer software reviews, or vice versa.

As Table 5 shows, the performance of *LearningOnly* dropped greatly in comparison with the standard setting (e.g., from 80.54% to 68.55% for customer reviews of browsers), but the performance of *pSenti* was not affected much (e.g., from 79.60% to 77.10% for customer reviews of browsers). Consequently *pSenti* turned out to be significantly better than *LearningOnly* in terms of adapting to new review styles. The performance of *SentiStrength* was still far lower than all the other systems.

The above experimental results indicate that the pure

Table 4: The sentiment strength detection performance (RMSE) in the standard (single-style) setting.

Dataset		*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Software Reviews	Utilities 2 (Customer)	1.56	1.50	1.45	1.77
Movie Reviews	Movies 2	0.87	0.98	0.60	1.13

learning-based approach tends to overfit the style of training reviews, while the hybrid approach can inherit the cross-style stability of the lexicon-based approach and adapt to test reviews more easily.

From the practical point of view, such a cross-style ability of *pSenti* will greatly help sentiment analysis in less-structured social media sources like twitter, which usually do not have high-quality training data. It would be much easier to find professionally prepared reviews with reliable sentiment labelling, e.g., from the critics's columns in a newspaper, and then transfer the constructed model to informal reviews.

4.4 Discussion

The performance of sentiment analysis partially depends on the sentiment separability of reviews: if there is a clear separation between the positive and negative sentiment value distributions, the pure lexicon-based approach would work well; otherwise machine learning would substantially boost the performance. As shown in Figure 3, the sentiment separability in movie reviews is much lower than that in (editor or customer) software reviews. Correspondingly, we see in Table 3 that by incorporating machine learning, *pSenti* could achieve a much larger performance improvement over the pure lexicon-based approach on movie reviews rather than on software reviews.

One reasons for poor sentiment separability in movie reviews is that many movie reviews in the given dataset contain a plot description and many quotes from the movie. For example, in the sentence "when you get out of jail, you can kill him", the reviewer has used several negative words, but he or she is just quoting one of an actor's utterances rather than expressing any opinion. Such blocks of objective text could be a significant source of sentiment value distortion. Pang and Lee have demonstrated that by removing objective text from movie reviews they are able to obtain significant improvement in sentiment analysis accuracy [14]. We have also tried to apply a similar subjectivity detection algorithm (based on graph min-cut) to our experimental datasets, but it did not generate a noticeable positive effect on the overall system performance. Nevertheless, as we can see from the results shown in Tables 3, that even without subjectivity detection, our hybrid approach *pSenti* can achieve 82.30% accuracy, which is only slightly below the bag-of-words SVM. The development of a more effective subjectivity detection algorithm is part of our future work.

5. CONCLUSIONS

We have shown that the sentiment analysis results produced by our hybrid approach are favourable compared to the lexicon-only and learning-only baselines. For both sentiment polarity classification and sentiment strength detection, our *pSenti* system, based on the proposed hybrid approach, achieves high accuracy that is very close to the pure learning-based system, and much higher than the pure lexicon-based system. Furthermore, *pSenti* can provide sentiment analysis results in a structured and readable way by dividing the overall sentiment into aspects (e.g., product features) and their corresponding views. Moreover, it has much better tolerance to the writing style of text, as demonstrated by our cross-style experiments, where the system is trained on editor reviews and then tested on customer reviews, or vice versa. Compared with a representative state-of-the-art sentiment analysis system *SentiStrength*, our *pSenti* system works consistently and significantly better. In summary, our proposed hybrid approach is able to combine the best of two worlds — the stability as well as readability from a carefully designed lexicon, and the high accuracy from a powerful supervised learning algorithm.

It would be promising to further explore the potential of this approach, e.g., for cross-domain sentiment analysis, objective/subjective text classification, and other advanced opinion mining tasks.

6. ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] A. Andreevskaia and S. Bergler. When specialists and generalists work together: overcoming domain dependence in sentiment tagging. In *In Proceedings of ACL-08: HLT*, 2008.

[2] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg, and S. Levitan. Stylistic text classification using functional lexical features: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):802–822, Apr. 2007.

[3] E. Cambria, M. Grassi, A. Hussain, and C. Havasi. Sentic computing for social media marketing. *Multimedia Tools and Applications (MTA)*, 59(2):557–577, 2012.

[4] E. Cambria and A. Hussain. *Sentic Computing: Techniques, Tools, and Applications*, volume 2 of *SpringerBriefs in Cognitive Computation*. Springer, Heidelberg, 2012.

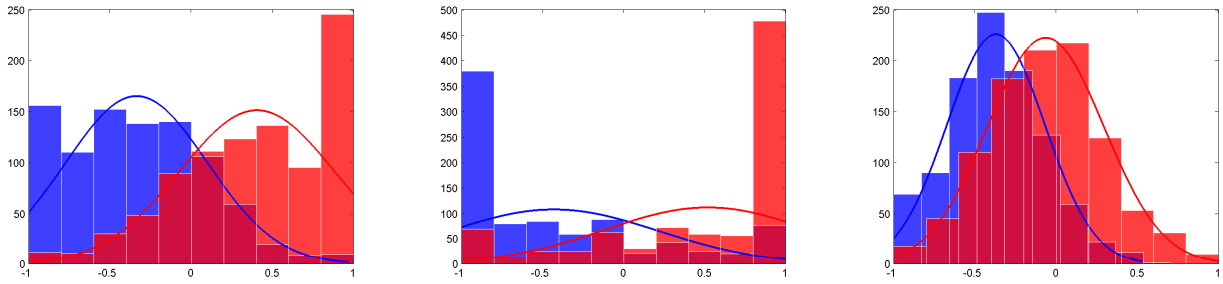
[5] E. Cambria, A. Hussain, C. Havasi, and C. Eckl. Sentic computing: Exploitation of common sense for the development of emotion-sensitive systems. In *Proceedings of the 2nd COST 2102 International Training School*, pages 148–156, Dublin, Ireland, 2009.

[6] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 231–240, New York, NY, USA, 2008. ACM.

[7] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

Table 5: The sentiment polarity classification performance (accuracy) in the cross-style setting.

Training	Testing	*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Browser (Customer)	Miscellaneous (Editor)	77.47%	79.40%	71.92%	64.93%
Browser (Customer)	Browser (Editor)	77.78%	76.94%	75.28%	62.77%
Miscellaneous (Editor)	Browser (Customer)	77.10%	74.50%	68.55%	52.25%
Browser (Editor)	Browser (Customer)	75.90%	74.50%	65.80%	52.25%



(a) Software Reviews (Editor) (b) Software Reviews (Customer) (c) Movie Reviews

Figure 3: The sentiment separability of reviews shown by the positive (red) and negative (blue) sentiment value distributions.

[8] Y. He. Incorporating sentiment prior knowledge for weakly supervised sentiment analysis. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(2):4:1–4:19, June 2012.

[9] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[10] M. Hu and B. Liu. Opinion feature extraction using class sequential rules. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 61–66, Stanford, CA, USA, 2006.

[11] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 815–824, New York, NY, USA, 2011. ACM.

[12] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 375–384, New York, NY, USA, 2009. ACM.

[13] S. Owsley, S. Sood, and K. J. Hammond. Domain specific affective classification of documents. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 181–183, Stanford, CA, USA, 2006.

[14] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

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

[16] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[17] J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pages 43–48, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[18] E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 105–112, Sapporo, Japan, 2003.

[19] B. Schuller and T. Knaup. Learning and knowledge-based sentiment analysis in movie review key excerpts. In *Proceedings of the 3rd COST 2102 International Training School*, pages 448–472, Caserta, Italy, 2010.

[20] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology (JASIST)*, 63(1):163–173, 2012.

[21] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[22] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417–424, Philadelphia, PA, USA, 2002.

[23] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 129–136, Sapporo, Japan, 2003.