Web Intelligence and Agent Systems: An International Journal 8 (2010) 291–302
DOI 10.3233/WIA-2010-0192
IOS Press

Leveraging sentiment analysis for topic detection

Keke Cai [a,*], Scott Spangler [b], Ying Chen [b] and Li Zhang [a]
[a]*IBM China Research Lab, Beijing, China*
[b]*IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, USA*

Abstract. The emergence of new social media such as blogs, message boards, news, and web content in general has dramatically changed the ecosystems of corporations. Consumers, non-profit organizations, and other forms of communities are extremely vocal about their opinions and perceptions on companies and their brands on the web. The ability to leverage such "voice of the web" to gain consumer, brand, and market insights can be truly differentiating and valuable to today's corporations. In particular, one important form of insights can be derived from sentiment analysis on web content. Sentiment analysis traditionally emphasizes on classification of web comments into positive, neutral, and negative categories. This paper goes beyond sentiment classification by focusing on techniques that could detect the topics that are highly correlated with the positive and negative opinions. Such techniques, when coupled with sentiment classification, can help the business analysts to understand both the overall sentiment scope as well as the drivers behind the sentiment. In this paper, we describe our overall sentiment analysis system that consists of such sentiment analysis techniques, including the bootstrapping method for word polarities weighting, automatic filtering and expansion for domain word, and a sentiment classification method. We then detail a novel topic detection method using point-wise mutual information and term frequency distribution. We demonstrate the effectiveness of our overall approaches via several case studies on different social media data sets.

Keywords: Sentiment topic detection, consumer generated media, pointwise mutual information, business insight workbench and sentiment analysis

1. Introduction

The widespread availability of consumer generated media (CGM) such as blogs, message boards, and news articles post great opportunities as well as risks to today's enterprises. Corporations could tab into such content to understand consumer opinions about their products and services, and hence creating new innovation opportunities and competitive advantages. On the other hand, when not paying attention to such consumer generated media, companies could create significant risks in brand image and corporate reputation when certain issues are not handled early and effectively, since the spread and the speed of such CGM information over the internet could render the publicity uncontrollable.

Clearly, new analytical methods that leverage such CGM content to understand consumer opinions are desperately needed. In this paper, we survey existing key techniques in sentiment mining, i.e., sentiment classification that attempts to understand the sentiment tonality of the web comments by classifying comments into positive, negative, and neutral categories. Such analysis is useful, but it lacks insights on the drivers behind the sentiments. To address this problem, we introduce our sentiment analysis approach which combines a unique sentiment classification approach with a topic detection approach that discovers terms that are highly correlated to different sentiment classification categories. The overall solution not only determines the sentiment about a given topic, but also uncovers the potential root causes of the sentiments.

* Corresponding author. E-mail: caikeke@cn.ibm.com.

1.1. Related work

Existing studies reported in the literature of opinion analysis has significant focuses on sentiment classification, which intends to differentiate user opinions and classify opinion comments into positive, negative and neutral categories. The general classification approaches can be summarized into two categories:

Semantic-based approach. This type of approaches mainly relies on opinion word collection in the form of sentiment dictionary [1,2] or a large-scale knowledge base [3] to assign sentiments to individual documents. Here, opinion words refer to those sentimental words possessing positive or negative sentiment, e.g., “good”, “well”, “terrible”, etc. Once the sentiment word collection is established, the common approach adopted for sentiment classification is to evaluate the average semantic orientation of all sentiment words in each document. Obviously, the key element of this approach lies in the techniques to establish appropriate semantic word base. We found three representative approaches in the literature to construct such sentiment words base, including manual construction [4], semi-automatic construction [5,6] and automatic construction [7–9]. Such semantic approaches are often adaptive and easy to use. However, establishing the baseline sentiment word base can be challenging. We describe a machine-aided and iterative approach in this paper, which intends to alleviate such pains and go beyond simple semantic-based sentiment classification.

Learning-based approach. The common learning-based approaches for sentiment classification typically leverage the manually labeled documents as the training set, and then apply traditional learning techniques, such as, Naïve Bayes, Maximum Entropy and SVMs, to do sentiment classification. Features describing each comment can be simple words [10,11], n-grams [12,13], and syntactic relations [12–14]. The disadvantages of such approaches are the following: First, building labeled documents can be challenging. Some approaches try to use existing movie reviews as training sets. But it is unclear if movie reviews would translate to other domains very well, such as food or financial industry. This leads to the second drawback. That is, the approach may not be adaptive enough to work across different data sets or domains.

1.2. Overview of our sentiment approach

In this paper, we describe an overall approach that leverages sentiment classification, but goes beyond it. Sentiment classification alone is often useful, but insufficient. Since opinions are always expressed in certain background, it is critical to understand what is behind the sentiment. Considering the following three sample customer comments:

1. I love Snickers because they satisfy my hunger.
2. I love to have Snickers for a snack when I get hungry in the afternoon.
3. Nothing makes me happier than eating a Snicker when I get hungry for a snack.

Although the current opinion analysis techniques would have correctly classified all of these three reviews as positive sentiment, there is no indication on why people expressed positive sentiments. The manual analysis of these reviews reveals that the word “hunger/hungry” is commonly associated with this positive sentiment. Hence, to certain degree, “hunger” can be considered as the driver of the positive sentiment.

Clearly uncovering the reasons for the positive/negative sentiment is critical and significantly more insightful than sentiment classification in and of itself. We call such “reasons” the “sentiment topics” associated with the sentiments. This paper presents the techniques used to identify such sentiment topics, which can be described as key words closely associated with each sentiment category. We have not found extensive prior work in this area.

In summary, the key contributions of this paper include the following:

1. We present an end-to-end sentiment analysis framework which combines sentiment classification approaches with sentiment topic detection approaches in one system.
2. We present our semantic-based sentiment classification method.
3. We define the sentiment topic concept and present sentiment topic detection approach (*STD*).
4. We verify the effectiveness of our approach on CGM data sets using real-world usage cases.

The rest of this paper is organized as follows. Section 2 introduces the overall sentiment analysis framework. Section 3 provides the detailed description of the key processes and components of the sentiment analysis, including sentiment classification and sentiment topic detection. We show experimental results in Section 4. Finally, we draw conclusions and outline future work in Section 5.

2. Framework

2.1. Definitions

Before introducing the overall sentiment analysis framework, let us first provide some definitions involved in our discussions below.

Snippet: A snippet is a small text segment around a specified keyword in a given document. The text segment can be defined by sentence boundaries, or the number of words. In general, snippets are built around core keywords, e.g., brand names or corporation names. Snippetization is important for analyzing web content, since web contents often are noisy. They may cover diverse topics in one document, even though only a few sentences might be relevant to the analysis subject. Snippets allow users to focus on the relevant text segments. This is especially important to sentiment analysis, since sentiment analysis of the overall document is likely to bias the opinion of the concerned subject, which on-topic snippet-based sentiment analysis could be much more meaningful. In this paper, all of our analysis is done on snippets. We usually select pre- and post- 1 or 2 sentences to construct each snippet.

Semantic Dictionary: Our overall sentiment analysis contains a semantic-based classification technique. Our approach utilizes two types of semantic dictionary for sentiment classification, i.e. domain-specific and domain-independent. Words that have general sentiment meanings are in domain-independent dictionaries, such as "good" and "bad". Others that may carry different meanings in different domains will show up in different domain-specific semantic dictionaries.

Domain Dictionary: Domain dictionary is broader than sentiment dictionary in that it covers common words specific to a given domain, whether or not they are sentiment words, non-sentiment words. For example, "therapy" is a healthcare domain related word.

Sentiment Topics: Sentiment topics represent a type of background or correlated information behind each concerned sentiment. In our approach, each topic will be described through a set of representative words.

2.2. Sentiment analysis framework overview

As shown in Fig. 1, the overall framework of our sentiment analysis consists of two key components: the sentiment classification component and the sentiment topic recognition component. The sentiment classification component computes the sentiment polarity of each snippet and creates sentiment taxonomy. Based on the result of this component, the topic detection component further identifies the most significant information related to each sentiment category. The overall process follows the steps listed below:

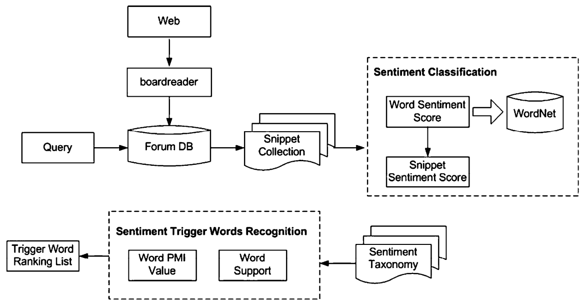

Fig. 1. Framework of the sentiment analysis system.

1. Collect the related web comments discussing certain object from external content repository, such as blogs, message boards, and news articles etc, and create the web content data warehouse.
2. For given sets of subjects to be analyzed, such as brands or products, extract its related snippets from the data warehouse.
3. Calculate the sentiment score of each snippet.
4. Classify snippets into different sentiment categories based on their sentiment scores and create the sentiment taxonomy.
5. Identify the most significant topics related to each sentiment category.

2.3. Key sentiment analysis components

2.3.1. Sentiment classification component

Our classification approach is based on semantic-based approaches as described previously. However, it not only considers the positive or negative polarity of word, but also evaluates the degree of sentiment each word expresses and assigns sentiment scores to words based on their definitions. Such techniques form the basis of the classification. Overall, our sentiment classification approach consists of four sub-steps: 1) construct sentiment lexicon; 2) measure individual word's sentiment; 3) combine words contained in snippet to form the final sentiment score for the snippet; 4) create sentiment classes based on the snippet scores. Section 3.1 describes the detailed approaches. We do not repeat in Section 2.3.1.

2.3.2. Sentiment topic words recognition component

Sentiment classification summarizes people's opinions but does not disclose the underlying reasons. Sentiment topic recognition intends to tackle this problem and disclose the most representative topics discussed behind each sentiment. Sentiment topic here is represented by one or more words. From these topic words, we can easily get an overall knowledge of on which aspects people have positive or negative opinions. The reason that we use the original format of word to describe topic is that it is far more natural for the user to directly understand these important words, rather than scan the abstracted objects containing these words.

In our sentiment topic recognition component, topic word is identified based on its importance, which is further evaluated from two aspects. One is word Pointwise Mutual Information (PMI) value and another is word support in category. PMI (also called specific mutual information) is a measure of association used in information theory and statistics[1]. PMI value between two discrete random variables reflects their dependence. The two variables are independent if the value is zero, and perfectly associated if the value is much more than zero, and complementary to each other if the value is much less than zero. PMI can discern the association between variables, but is always biased towards infrequent words [15]. In our approach, we also consider the factor of word support to balance the evaluation of association. We describe the detailed algorithm in the following section.

3. Sentiment topic detection algorithm

3.1. Sentiment based taxonomies

Our overall sentiment analysis starts with creating an effective sentiment based taxonomy. We use a statistically based approach for sentiment analysis which does not assume or attempt to determine any particular subject or object for the sentiment expressed. We simply measure the relative sentiment (on a positive/negative scale) expressed by the words in each snippet and use this numeric score as a way to partition the snippets into positive/negative/neutral categories, and hence creating a sentiment taxonomy.

[1] http://en.wikipedia.org/wiki/Pointwise_mutual_information.

3.1.1. Establish positive/negative words lists

To construct a sentiment lexicon we begin by creating positive and negative word lists. We utilized two external NLP resources viz. (i) The Inquirer database[2] and (ii) WordNet[3] for such purposes.

The Inquirer database contains more than 4,000 unique words, mostly adjectives. For each word it defines ~200 Boolean attributes. Some examples of attributes are: isPositive, isNegative, isHostile, isRelatedToPleasure, isRelatedToPain. This word-attribute matrix is very sparse, and only a few attributes for each word have the value true. Some of these attributes are used to decide whether the word is used mostly in positive sense or in the negative sense. For example, 'able' is positive whereas 'abnormal' is negative. WordNet is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs and adjectives are organized into synonym sets, each representing one underlying concept.

From the list of adjectives initially compiled, an adjective is marked as positive (*PosWord*) if it has isPositive and/or some other similar attribute is true according to Inquirer. If not found in Inquirer, we gather its first level synonyms from WordNet and determine their orientation using Inquirer. If most of the synonyms of a word are positive then the word is marked as positive. A similar procedure is executed to obtain the negative adjectives (*NegWord*). Examples of *PosWords* are "excellent", "caring", "jubilant", "mindful", etc. Whereas "agony", "arrogant", "blemish", "blunder", etc are examples of *NegWords*. This process resulted in a baseline list of 1905 Positive words and 2282 Negative words, which is then used for sentiment scoring.

3.1.2. Iterative induction of word sentiment degree

In order to accurately score relative sentiment between different posts that both use positive/negative words, we attempt to characterize the degree (amount) of positive/negative sentiment each sentiment word conveys. Here we present a novel technique that bootstraps word polarities using WordNet – specifically the synonym lists and the gloss entries from WordNet. The idea is that positive/negative sentiment words are not equivalent in character, but differ in the degree of sentiment they each convey. We try to capture the sentiment content of their definitions. But as we refine the sentiment content of each word we may benefit from recalculating the sentiment of

[2] http://www.wjh.harvard.edu/~inquirer/.
[3] http://www.cogsci.princeton.edu/~wn/.

the definitions (since the value of each sentiment word has changed). Thus we iterate looking for improvement.

Briefly speaking, the initial sentiment degree is done by looking up the words dictionary definition in WordNet and counting the occurrence of positive minus negative words in the definition. To normalize, we divide the sum by the total number of definitions. The occurrence of the word itself in its own definition is counted only once, where as other positive/negative words can be counted multiple times. As a further refinement, only adjective definitions are used, no other part of speech definitions are considered. This raw summation gives the relative amount of sentiment each individual word has. For example, by this method the word "wonderful" has a positive score of 13, because its one definition contains 13 positive words. The word "amnesty" has a much lower score of 1.25, because its four definitions contain 5 positive words. Then, the sentiment degree of each word is iteratively induced by recalculating the sentiment of the definitions as each word's sentiment value changes.

Detailed method is described as follows: Firstly, each word is looked up in WordNet. Since WordNet senses are organized in decreasing order of sense frequencies in the Brown corpus, the top "k%" senses are picked for each part of speech, for each word in the positive/negative dictionary. The gloss of one of the senses of *'altruistic'* in WordNet looks like:

"altruistic, selfless – (showing unselfish concern for the welfare of others)"

where *'selfless'* is a synonym for *'altruistic'*, while the sentence within brackets is the *gloss* for *'altruistic'*. The words in the gloss and the synonymy list that occur in the *PosWord* list will be called *PosTriggers*, while those that occur in the *NegWord* list will be called *NegTriggers*. The gloss of each sense (*s*) corresponding to each part of speech type (*p*) for each word (*w*) is parsed and tokenized and the number of *PosTriggers* (*|PosTriggers(w,p,s)|*) and *NegTriggers* (*|NegTriggers(w,p,s)|*) are computed. This is done for each word *w* in the original dictionaries *PosWords* and *NegWords*. The positive polarity *PosPol(w)* of each such word $w \in PosWords$ is computed as:

$$PosPol(w) = \frac{\sum_{p \in Pos(w)} \sum_{s \in senses(w,p)} |PosTriggers(w,p,s)|}{length(w)} \quad (1)$$

where *Pos(w)* is the set of part of speech types that *w* has according WordNet and *Senses(w,p)* is the set of all senses of word *w* for part of speech type *p*. The quantity *length(w)* can be computed either as (i) the number of characters, (ii) number of words or the (iii) number of *triggers* (both positive and negative) within all the glosses of *w* across all its part of speech types *p* and sense types *s*.

Further, this process can be continued iteratively to compute *PosPol(w,i)* in the $(i+1)^{th}$ iteration using the polarities *PosPol(w,i-1)* computed in the i^{th} iteration. *PosPol(w,0)* can be initialized to the *PosPol(w)* computed above.

$$PosPol(w, i+1) = \frac{\sum_{p \in Pos(w)} \sum_{s \in senses(w,p)} (|PosTriggers(w,p,s,i)|}{length(w)} \quad (2)$$

where, *|PosTriggers(w,p,s,i)|* is the sum of the polarities of positive triggers in the s^{th} sense of the p^{th} part of speech of word *w*, each trigger *t* weighted by its corresponding *PosPol(t,i)* value. These update rules can be executed for several iterations until there is negligible change in the polarities (least squares or sum of absolute values of distances) across two iterations.

The negative polarity *NegPol(w)* for every word $w \in NegWords$ can be computed in a similar manner as:

$$NegPol(w) = \frac{\sum_{p \in Pos(w)} \sum_{s \in senses(w,p)} |NegTriggers(w,p,s)|}{length(w)} \quad (3)$$

Similarly, the iterative update rule for *NegPol(w,i)* is expressed as:

$$NegPol(w, i+1) = \frac{\sum_{p \in Pos(w)} \sum_{s \in senses(w,p)} (|NegTriggers(w,p,s,i)|}{length(w)} \quad (4)$$

where, *|NegTriggers(w,p,s,i)|* is the sum of the polarities of negative triggers in the s^{th} sense of the p^{th} part of speech of word *w*, each trigger *t* weighted by its corresponding *NegPol(t,i)* value. This helps in propagating positivity and negativity.

The same technique used for scoring negative/positive words in the original word lists can be used to score any word in the WordNet dictionary. Only in the case of the dictionary, each word may have both a positive and a negative impact based on having both positive and negative words in its definition. This will generally give the dictionary terms less individual impact on the sentiment score than the words in the original positive/negative word list. Words that are not defined in WordNet are ignored for the purposes of sentiment classification. In this way, the sentiment dictionary is expanded to incorporate all words in domain that are defined in WordNet.

3.1.3. Domain sentiment word filtering and expansion

When applying sentiment analysis in different domains, the general sentiment words are not always adapting to the specific domain and therefore arises a problem. In the sentiment word dictionary, some opinion words are domain independent, but some are domain dependent. For example, the words "health" and "care" by default are positive sentiment words, but in specific domain, it is not appropriate to treat them as positive sentiment words.

We propose a method of domain specific sentiment word detection. It goes beyond the traditional applied sentiment dictionary by providing each specific domain the unique sentiment dictionary, and therefore can guarantee the high accuracy of sentiment analysis to some extent. Our method consists of two parts, domain-specific sentiment word filtering and domain-specific sentiment word expansion. Our approach for word filtering simultaneously considers word occurrence frequency in domain and background and finally identifies which part of words should be removed from general sentiment dictionary. The approach for word expansion realizes expansion from the perspective of word morph. The assumption is that words with different morphs have the same sentiment polarity.

In detail, domain specific sentiment word filtering is to estimate whether or not a general sentiment word can be applied for specific domain. Given a general sentiment word and a specific domain, the applicability of word to domain is decided by factors of 1) the ratio of word domain frequency to word background frequency, and 2) word domain frequency. So, the filtering method consists of following steps:

```
FOR each general sentiment word in the dictionary DO
    Step 1 Detect word distributions in background dataset;
    Step 2 Detect word distributions in domain dataset;
    Step 3 Filter word by synthetically considering:
      1) The ratio of word domain frequency to background
         frequency;
      2) The word frequency in the domain;
    Step 4 Get the list of word to be removed.
```

Domain specific sentiment word expansion is to detect sentiment words that can be expanded to specific domain. Given a word, it will be firstly mapped to general sentiment word based on stemming process and then evaluated by word filtering check for domain. Words finally left will be identified as expansion words for domain. The purpose is to identify all different forms of sentiment words and then add them to sentiment dictionary. If the word has been included in sentiment dictionary, then we can expand its all base forms to sentiment dictionary. And if any base form of the word has been included in the sentiment dictionary, then we can expand this word and its other forms to sentiment dictionary. Detailed steps are:

```
  Step 1 Identify word candidates expanded for general sentiment dictionary;
      Step 1.1 Detect the mappings between stems of original sentiment words and stems of words in background dataset and identify word candidates for expansion;
      Step 1.2 Based on the utilities of WordNet, filter word candidates for expansion with informal styles.
  Step 2 Filter words obtained in step 1 using domain specific word filtering approach.
  Step 3 Get a list of expanded words for specific domain.
```

After getting the list of filtered words and expansion words, the original sentiment dictionary will be further processed by deleting the filtered words and expanding with the expansion words.

3.1.4. Score snippets and partition into quintiles

We employ the polarity scores *PosPol(w)* and *NegPol(w)* determined above, to detect the polarity *Pol(S)* of any snippet *S*. This can be computed as the sum of polarity scores of all *PosWords* and subtract from that, the sum of polarities of all *NegWords* that occur in the snippet, normalized by the length *length(S)* of the snippet *S*. If any word *w'* is 'modified' by some negation word (such as '*not*') from a list *N*, its polarity score *PosPol(w')* or *NegPol(w')* is negated.

The normalization by *length(S)* helps capture the density of occurrences of polarized words within *S*. As before, *length(S)* can be computed as either the number of characters, the number of words or the number of polarized words in *S*. Optionally, the resulting score can be multiplied by 10 in order to bring the number somewhere close to 1.0. The sum of the positive sentiment word scores for a given snippet, minus the sum of the negative sentiment word scores divided by the square root of the overall length of the snippet gives the overall snippet sentiment score.

$$Pol(S) = \frac{\sum_{w \in S \cap PosWords} PosPol(w)\partial(N,w,S) + \sum_{w \in S \cap NegWords} NegPol(w)\partial(N,w,S)}{\sqrt{length(S)}} \times 10 \qquad (5)$$

where $\partial(N,w,S)$ is –1 if w is preceded in S by a negation word from N and is +1 otherwise.

This scoring method was validated against human rated data and found to have a high degree of correlation to notions of sentiment. Given this scoring we then partition the data into five classes by sorting the snippets by sentiment score and creating a category for each quintile of data. The two extreme quintiles form a positive and a negative class, and the three middle quintiles are merged to form a single "neutral" class. Thus the final taxonomy contains three categories: Positive, Negative, and Neutral.

3.2. Sentiment topic words recognition

We use two factors to detect sentiment topic words, i.e., word PMI value and word support. PMI value evaluates the uniqueness of word to each sentiment category. The following algorithm is adopted to calculate the PMI value of word w against the category s.

$$PMI(w,s) = \log(\, p(w,s)/(p(s) * (p(w) + 0.05))\,) \qquad (6)$$

where, $p(w, s)$ describes the co-occurrence between w and s, $p(s)$ represents the distribution of category s and $p(w)$ evaluates the distribution of word w in the whole snippet collection. Considering that the factor of $p(s)$ has no influence on words ranking for each category, and therefore can be ignored.

Word support evaluates the importance of word in category. It is calculated as the following:

$$Freq(w,s) = N(w,s) \Big/ \sum_{s \in Positive, Negative, Neutral} N(w,s) \qquad (7)$$

where, $N(w, s)$ denotes the number of word w in category s.

PMI and word support evaluate the importance of a word to each sentiment category from different points of view. When considered simultaneously, they can detect relevant sentiment topic words. On the other hand, ignorance of any of them may negatively impact the ability to detect the important sentiment topical words. For example, suppose that words A and B have equal PMI value to positive category, but their frequency in positive category could be 1 and 100 respectively. Without the word support metric, the importance of A and B to the positive category may be considered equal, which could be misleading. On the contrary, assuming that the frequencies of words A and B in positive category were both 100, but their occurrence frequencies in negative category were 1000 and 0 respectively. Clearly, considering word frequency in single sentiment category alone may not be indicative to the difference of A and B to positive category.

By collectively considering the factors of word PMI and support frequency, significant topic words related to each sentiment category can be identified effectively.

Overall, topic words are identified through the following steps:

1. Classify documents into categories of positive, negative and neutral using sentiment classification techniques described in Section 3.1.
2. Identify all words in documents and filter all stop words and sentimental words, to keep only non-sentiment words as sentiment topical word candidates.
3. Calculate the frequency of the words remaining in step 2 in each single sentiment category as well as across all categories to establish word supports.
4. Calculate the PMI value of the words in each sentiment category based on the formula 2.
5. Combine the frequency of the words in each category with its PMI value and select the top frequent words with high PMI value as the final sentiment topic words.

4. Implementation and experiments

To evaluate the effectiveness of our overall sentiment analysis approach, we implemented and embedded our techniques in a general purpose analytics workbench, called **Business Insights Workbench (BIW)**. We then evaluated our approach using two real-world usage scenarios. **BIW** is a solution developed by IBM's Almaden Research Center that provides integrated structured and unstructured information mining. BIW embeds a suite of information analytics and data processing technologies to improve the caliber of decision making for enterprises (see [16] for detailed descriptions about BIW).

4.1. General sentiment words generation and weighting

We first used Eqs (1) and (3) to generate *PosPol(w)* for each $w \in PosWords$ and *NegPol(w)* for each $w \in NegWords$. We restricted *Pos(w)* – the set

Table 1

Word samples with low NegPol value

WordName	RawCount	IterativeScore
root	2	0.133333
study	2	0.125
complex	1	0.2
confession	1	0.2
renunciation	1	0.25
stern	1	0.333333
fine	1	0.2
fine	1	0.2
needle	1	0.166667
board	3	0.230769
backward	1	0.2
fed	2	0.153846
raise	6	0.193548
order	6	0.25
point	8	0.216216
mix	2	0.222222
immature	1	0.2

of parts of speech of a word w to consider in the scoring in Eqs (1) and (3) to just adjectives and adverbs. Further, in Eq. (1) as well as Eq. (3) |*PosTriggers(w,p,s)*| and |*NegTriggers(w,p,s)*| accounted for only single occurrences of each trigger word across the gloss as well as synonymy list of sense *s*. Finally, the word itself was counted exactly once.

Table 1 presents some negative words with low values of *NegPol* and their scores as produced using our iterative scoring technique. The scores are obtained using Eq. (1). The first value "RawCount" is the number against any word is the raw count of trigger words. The second value "IterativeScore" corresponds to the *NegPol* score with *length(w)* computed as the "total number of senses of *w*, across all its pos types". As can be seen, the computed polarity reflects the actual polarity of the word quite closely (at least relative to other words).

4.2. Domain sentiment word filtering and expansion

In order to filter and expand domain sentiment words, we collect a data set belonging to several different industries, such as banking, consumer electronics, telecomm, energy, retail, and information technology, etc. For a specific domain, we use all the data as background data. Table 2 lists the words filtered from the negative word list for a data set consisting of snippets for the "Target" department store. Table 3 lists the expanded words for "Tesco" corporation.

Table 2

Filtered negative words for Target department store

fall, drop, even, wait, deal, box, order, try, decline, cost, close, cut

Table 3

Expanded positive words for Tesco corporation

firm, effectively, standardization, convinced, precisely, reclaimed, coherence, renew, caring, exert, reinforce, cleansing

4.3. Sentiment classification evaluation

We use two data sets to evaluate the performance of sentiment classification results. One data set is called "Review Rating dataset". The other is "vegemite reviews".

4.3.1. Review rating dataset

Given a review about a product or a service, we need to do the sentiment analysis at the sentence level and assign a rating between 1 and K, such that the sentences having a rating below (K+1)/2 are considered to be negatively opinionated sentences while the sentences having rating more than (K-1)/2 are considered to be positively opinionated sentences. More formally the problem can be stated as follows:

- Input: Customer Reviews such that each sentence is tagged with a rating between 1 and K. Our evaluation data has K = 1 to 7, i.e., a sentence is considered to be negative opinionated if the rating is less than 4 and positive opinionated if the rating is greater than 4.
- Output: Customer Reviews such that each sentence is tagged with a (real-valued) rating between 1 and K. Our evaluation data has K = 1 to 7, i.e., a sentence is considered to be negative opinionated if the rating is less than 4 and positive opinionated if the rating is greater than 4.
  - Rating 7: (highly praised) it's very sleek looking with a very good front panel button layout and it has a great feature set.
  - Rating 1: (highly criticized) this player is not worth any price and I recommend that you don't purchase it.

Table 4

Review rating dataset statistics

S.No	File Name	1	2	3	4	5	6	7	Total
1	apex.txt	83	94	15	495	25	94	33	839
2	cannon.txt	7	34	14	403	30	96	58	642
3	creative.txt	74	148	68	1091	55	247	128	1811
4	nikon.txt	1	20	9	220	12	75	43	380
5	nokia.txt	4	45	24	321	33	114	45	586
6	TOTAL	169	341	130	2530	155	626	307	4258

- Rating 4: (neutral) if this doesn't bring back the picture, please try pressing this button to play a DVD.

Some statistics from the evaluation data are provided in Table 4. The dataset had reviews about 5 products, provided as 5 different files. Each sentence in the dataset was tagged with the opinionated word and the strength of the opinion, ranging between 1 and 7. While the evaluation data had integral ratings, our method produces real-valued ratings. The column headings (for columns 3 through 9) correspond to the ratings or polarities of the sentences while entries in the columns correspond to the number of sentences that the dataset for that row has, rated with that polarity.

4.3.2. Vegemite reviews

The Vegemite data set consisted of 800 snippets around the term “vegemite” and similar products (e.g. “marmite”, “nutella”). Each snippet was around three sentences in length. Eight human reviewers rated 100 snippets each on a scale of 1–7 (much like the review rating data set). The reviewers were asked to rate the overall sentiment of each text snippet (1= most negative, 7 = most positive, 4 = neutral). An eighth human reviewer rated a random sample of 300 snippets to determine correlation between different human opinions.

4.3.3. Results on scoring reviews

We used a Pearson correlation statistic (1.0 = perfect correlation, 0.0 = no correlation) to evaluate the different approaches. In each case we compared the method described in the invention to the baseline method that scores text based on the counts of positive dictionary terms minus counts of negative dictionary terms divided by the square root of the snippet length. We compared the simple unary (positive words all get +1 score, negative words get –1 score) scoring of snippets to the offline iterative induction of word weights. The results are shown below.

Review Ratings Datasets (combined):
Pearson correlation (human, simple unary) = .322
Pearson correlation (human, offline iterative) = .340

Vegemite Snippets data:
Pearson correlation (human, simple unary) = .259
Pearson correlation (human, offline iterative) = .323
Pearson correlation (human1, human2) = .56

From the above result, we can see that offline iteration method is more similar to human's results than simple unary method, and agreement among human annotators is at most 0.56.

4.4. Sentiment topic words recognition

Our experiments were conducted upon three different types of CGM contents, i.e., blogs, message boards, and news articles. The first usage scenario intends to understand a particular sense of negative sentiment around an Australian Brand called “Vegemite” in a given time period. Such negative sentiment was discovered as part of our overall analysis on Vegemite brand perceptions. Due to space limitations, we do not describe the overall brand perception analysis. Instead, we focus on this specific negative sentiment issue.

Overall, the vegemite database contains 34,702 postings. Through our sentiment classification analysis, we found an outstanding negative sentiment in late October 2006. By manually examining postings in that period, we found that certain customs officials thought that the importation of Vegemite into the U.S. was supposed to be banned, even through there was no official “banning” of the product in U.S. This particular news caused great consternation among American expatriates from Australia and New Zealand and others who enjoy Vegemite. It can be seen from following real examples:

1. The United States has slapped a ban on Vegemite, ***outraging Australian*** expatriates there. The bizarre crackdown was prompted because

Table 5

Data set description of scenario 1

Topic	Vegemite
Issue description	Vegemite ban in America
Total # of Docs	34,702
# of relevant Snippets	1,104

Table 6

Main ground-truth words for Vegemite issue

Banned FDA vitamin Americans "illegal food" war folate "by-product beer" crisis taste "food law" import USA "folic acid" contraband Aussie "outraging Australian" Bush crackdown Australian expatriates policy

Vegemite contains ***folate***, which in the US can be added only to breads and cereals.

2. I feel very sorry for all those ***home sick Australian*** that can't get their daily fix of Vegemite! I find it ***unbelievable*** that people hate Vegemite! Vegemite is the only thing that goes on my toast.

Such examples not only expressed strong negative emotions for the ban, but also disclosed several topics behind this negative sentiment, such as "outraging Australian" and "folate". To detect the root causes of this negative sentiment, we extracted all snippets related to this accident, which resulted in 1,104 snippets. Table 5 summarizes the data set characteristics.

To find the topic words mostly related with the given sentiment, we adopted the standard evaluation measures of precision, recall and p@n (precision of the top n results) to measure the performance of our approach. To obtain the ground truths, we selected several team members and manually examined postings in our test case scenarios individually, and hand-labeled ground truths. We then validated the ground truths selection by comparing multiple team members' results and compiled the final ground truths for each of the cases by picking the words with higher votes from the teammates. Since the actual scenarios are known ahead of time, such a manual process is possible to do. The selected final ground truths are listed in Table 6.

We compared our approach (denoted as *STD*) with the Chi-square test based approach (*CHI-Square*) [17]. Chi-square test uses the distinction between word real distribution and expected distribution in sentiment category to measure word significance. Tables 7 and 8 illustrate the experimental results. As we expected, the *STD* approach is significantly better than *CHI-Square* approach in both precision and recall. The highest precision occurs in the top 10 terms and the recall achieves 61.3% among 46 selected terms. The performance of *CHI-Square* approach is unsatisfactory. This is probably because *CHI-Square* tends to bias the unique topics in category. Since our approach simultaneously considered word uniqueness and frequency, we can effectively avoid the ignorance of popular topics possessing by both sentiment categories.

Table 7

Top topic words by STD and CHI-Square

STD	folate Americans Australian add food ban states expatriates crackdown USA
CHI-Square	"outraging Australian" "vegemite outraging" outraging "states slapped" slapped crackdown extract "folate be" arms "yeast extract"

Table 8

Performance comparison on Vegemite issue

	STD	*CHI-Square*
p@10	70%	30%
P@20	55%	30%
p@30	43.3%	26.7%
precision	41.3%	17.4%
recall	61.3%	25.8%

Our second experiment is about an insurance company, InsuranceCo. For privacy purposes, we do not disclose the company name here. Similar to usage scenario 1, by sentiment classification analysis, we observed an event that heavily influenced people opinions on InsuranceCo. That is, the death of a girl due to the denial of InsuranceCo on liver transplant caused significant negative emotions over the internet (see examples below with anonymous names and locations):

1. Attorney Smith said he plans to ask the district attorney to press ***murder*** or ***manslaughter charges*** against ***InsuranceCo*** in the case. The insurer 'maliciously killed her' because it did not want to bear the ***expense*** of her ***transplant*** and ***aftercare***, Smith said.
2. The company reversed the decision Thursday as about 150 nurses and community members ***rallied*** outside of its office in X location. ***Natalie died*** just hours later.

Again, we extract 2,817 snippets around this subject from the InsuranceCo database which contains total of 34,654 postings. We again compared our approach with *CHI-Square*. Tables 9 and 10 respectively give the data set for the second scenario and

Table 9

Data set description of scenario 2

Topic	InsuranceCo
Issue description	Death of a girl due to the denial of InsuranceCo on liver transplant
# of Docs	34,654
# of Snippets	2,817

Table 10

Main ground-truth words for InsuranceCo issue

Leukemia “liver transplant” “teen death” death “Nataline sarkisyan” criminal murder “maliciously kill” spokeswoman aftercare “charged murder” “refused pay” “girl die” “declined comment” insurer cancer “charges InsuranceCo” manslaughter

Table 11

Top topic words by STD and CHI-Square

STD	district “charges InsuranceCo” “liver transplant” geragos family “Nataline sarkisyan” insurer attorney leukemia “charged murder”
CHI-Square	inappropriate “spokeswoman declined” “geragos plans” “inappropriate geragos” “charges InsuranceCo” maliciously “healthcare case” submits bear “insurer maliciously”

Table 12

Performance comparison on InsuranceCo issue

	STD	*CHI-Square*
p@10	80%	30%
P@20	70%	25%
p@30	63.3%	33.3%
precision	58.1%	31%
Recall	55.6%	28.9%

the ground truth words selected manually. Tables 11 and 12 illustrate the experimental results.

Result of the second experiment once again proved the effectiveness of the overall sentiment analysis approach. Performance of the precision and recall can always keep an effective balance. Furthermore, we can get the most relevant topics within the top 10 identified words. Comparatively, Chi-square test one is not particularly effective.

5. Conclusion

In this paper, we present an overall solution for sentiment analysis which includes a sentiment classification scheme as well as a sentiment topic detection scheme. Particularly we present a novel approach to bootstraps word polarities using WordNet, and automatically filter and expand sentiment words for a specific domain data. The sentiment classification component measures the relative sentiment (on a positive/negative scale) expressed by the words in each snippet and then partitions the snippets into positive/negative/neutral categories. The sentiment topic detection component detects the most significant topics hidden behind each sentiment category using a combined PMI and word support metrics. The combination of these approaches not only identifies sentiments, but also discloses the implicit root causes of the sentiment.

In the future, we will further enhance our topic words detection schemes by leveraging techniques such as part-of-speech and word syntactic relationships. We are also interested in techniques that automatically form high level topical models from basic topical words.

Acknowledgement

The authors gratefully acknowledge the contributions of Ganesh Ramakrishnan, Sachindra Joshi, and Raghuram Krishnapuram for their help in developing the sentiment classification approach and the BIW team for their support in the overall sentiment analysis implementation.

References

[1] T. Nasukawa and J. Yi, *Sentiment Analysis: Capturing Favorability Using Natural Language Processing*, Proceedings of the International Conference on Knowledge Capture, 2003, pp. 70–77.

[2] P. Subasic and A. Huettner, *Affect Analysis of Text Using Fuzzy Semantic Typing*, Proceedings of the Tenth IEEE International Conference on Fuzzy Systems, 2001, pp. 483–496.

[3] H. Liu, H. Lieberman and T. Selker, *A Model of Textual Affect Sensing using Real-World Knowledge*, Proceedings of the Seventh Conference on Intelligent User Interfaces, 2003, pp. 125–132.

[4] S. Das and M. Chen. *Yahoo! for Amazon: Extracting market sentiment from stock message boards*, Proceedings of the 8th Asia Pacific Finance Association (APFA), 2001.

[5] M. Hu and B. Liu, *Mining and summarizing customer reviews*, Proceedings of the 10th international conference on Knowledge discovery and data mining (KDD), 2004, pp. 168–177.

[6] L. Zhuang, F. Jing, and X.Y. Zhu, *Movie review mining and summarization*, Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM), 2006, pp. 43–50.

[7] J. Kamps and M. Marx, *Words with attitude*, Proceedings of the First International Conference on Global WordNet, 2002, pp. 332–341.

[8] P. Turney, *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 417–424.

[9] P.D. Turney and M.L. Michael, Measuring praise and criticism: Inference of semantic orientation from association, *ACM Trans. On Information Systems*, **21**(4) (2003), 315–346.

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

[11] K.T. Durant and M.D. Smith, Predicting the Political Sentiment of Web Log Posts Using Supervised Machine Learning Techniques Coupled with Feature Selection, *Lecture Notes in Computer Science*, Springer, **4811** (2007), 187–206.

[12] V. Ng, S. Dasgupta and S.M.N. Arifin, *Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews*, Proceedings of the 21st International Conference on Computational Linguistics (COLIN) and 44th Annual Meeting of the Association for Computational Linguistics (ACL), 2006, pp. 611–618.

[13] K. Dave, S. Lawrence and D. Pennock, *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews*, Proceedings of the Twelfth International World Wide Web Conference, 2003, pp. 519–528.

[14] J. Yi, T. Nasukawa, R. Bunescu and W. Niblack, *Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques*, Proceedings of the IEEE International Conference on Data Mining (ICDM), 2003, pp. 427–434.

[15] P. Pantel and D. Lin, *Discovering word senses from text*, Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2002, pp. 613–619.

[16] A. Behal, Y. Chen, C. Kieliszewski, A. Lelescu, B. He, J. Cui, J. Kreulen, J. Rhodes and W.S. Spangler, *Business Insights Workbench – An Interactive Insights Discovery Solution*, Proceedings of the 12th Int'l Conf. on Human-Computer Interaction. 2007, pp. 834–843.

[17] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Edition. New York: Cambridge University Press, 1992, pp. 620–623.