Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization

Wei Peng
Xerox Innovation Group
Xerox Corporation
Rochester, NY, 14580
wei.peng@xerox.com

Dae Hoon Park
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
dpark34@illinois.edu

Abstract

Although sentiment analysis has attracted a lot of research, little work has been done on social media data compared to product and movie reviews. This is due to the low accuracy that results from the more informal writing seen in social media data. Currently, most of sentiment analysis tools on social media choose the lexicon-based approach instead of the machine learning approach because the latter requires the huge challenge of obtaining enough human-labeled training data for extremely large-scale and diverse social opinion data. The lexicon-based approach requires a sentiment dictionary to determine opinion polarity. This dictionary can also provide useful features for any supervised learning method of the machine learning approach. However, many benchmark sentiment dictionaries do not cover the many informal and spoken words used in social media. In addition, they are not able to update frequently to include newly generated words online. In this paper, we present an automatic sentiment dictionary generation method, called Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) algorithm, to assign polarity scores to each word in the dictionary, on a large social media corpus - digg.com. Moreover, we will demonstrate our study of Amazon Mechanical Turk (AMT) on social media word polarity, using both the human-labeled dictionaries from AMT and the General Inquirer Lexicon to compare our generated dictionary with. In our experiment, we show that combining links from both WordNet and the corpus to generate sentiment dictionaries does outperform using only one of them, and the words with higher sentiment scores yield better precision. Finally, we conducted a lexicon-based sentiment analysis on human-labeled social comments using our generated sentiment dictionary [1] to show the effectiveness of our method.

Introduction

Since social network web sites have become popular media for people to share their opinions, enterprises have sought the opportunities to leverage this data for business intelligence applications such as enterprise marketing services and customer relationship management. It has become critical for enterprises to unlock customer sentiment embedded in the huge amount of social media data so that they can quickly respond to complaints and improve their product quality. Sentiment analysis is the study of using a machine to determine the polarity of an opinion - whether it is positive, negative, or neutral. However, while sentiment analysis (Pang, Lee, and Vaithyanathan 2002; Hu and Liu 2004) is no short-term hot research topic, few work has focused on social media data. Rather, research has been on more structured language such as product and movie reviews, due to the low accuracy resulting from much more informal writing, short sentences, sarcasm, and abbreviations seen in social media data. It has been studied that the probability of human strict agreement on opinion polarity is around 82% with two annotators (Wilson, Wiebe, and Hoffmann 2005a). However, our research shows that this number drops dramatically on social media data. We posted 8620 comments from Digg [2] onto Amazon Mechanical Turk [3], assigning each comment to three annotators, and found the rate of strict agreement to be 43.68% with two out of three annotators randomly selected for each comment. These numbers show the degree to which sentiment analysis on social media is more difficult than on well-written opinions.

There are typically two sentiment analysis approaches: machine learning based and lexicon based. The machine learning approach uses classification techniques to learn from human-labeled text sentences. The lexicon-based approach uses a sentiment dictionary with positive and negative words to match the words inside sentences to determine their polarity. This dictionary can also provide useful features for machine learning approaches; plus when the data is sparse, it is more important to rely on effective features. Most of current sentiment analysis tools on social media choose a lexicon-based approach because it is a huge challenge to obtain enough human-labeled training data for extremely large-scale and diverse social opinion needed for the machine learning approach.

One essential factor driving the performance of a lexicon-based sentiment analysis approach, is the sentiment (or polarity) dictionary used. We found that benchmark polarity

[1] It can be downloaded from http://users.cis.fiu.edu/∼wpeng002/sentiDict/

[2] http://www.digg.com

[3] https://www.mturk.com

dictionaries such as General Inquirer (GI) [4] and SentiWordNet (Esuli and Sebastiani 2006) are not very appropriate for social media because they do not cover or correctly tag some informal and spoken words frequently used on social media. For example, some words tagged as neutral in a benchmark polarity dictionary are actually negative (e.g. uptight, downtrodden, etc.), some words tagged as negative are actually positive (e.g. sassy, cool, etc.), and some words only exist on a particular social media (diggable, diggworthy, etc.). In this paper, we present an automatic sentiment dictionary generation method, which assigns polarity scores to each word in the dictionary, from a large social media corpus - digg.com. This method not only expands the lexicon coverage, but also can be easily updated to include newly created words online, which potentially can be extracted from different contexts to adapt to a particular topic.

Specifically, we propose to automatically generate an adjective sentiment dictionary from social media data with the following steps: (1) obtain a set of seed positive and negative adjective words and expand it using synonym and antonym relations from the WordNet (Fellbaum 1998) [5]; (2) extract all the adjectives linked to the adjective set by ‘and’ and ‘but’ using Part-Of-Speech (POS) technique on social media corpus; (3) construct a graph matrix (or a nonnegative symmetric matrix) where each entry is the edge weight between two adjectives calculated from the synonym relations from WordNet and the ‘and’ conjunction relations; (4) construct a constraint matrix (a nonnegative symmetric matrix) where each non-zero entry value denotes a Cannot-link weight between two adjectives calculated from the antonym relation from WordNet and the ‘but’ conjunction relation; (5) use our proposed Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) algorithm to iteratively cut this adjective graph into positive and negative sets, where each adjective is assigned a positive score and a negative score.

While there has been prior research on automatic sentiment dictionary generation, none of them construct the dictionary from a large-scale social network corpus combined with WordNet. Words from social media are more informal and diverse, and thus harder to determine the polarity. From our study shown in the following sections, people have higher disagreement on word sentiment from social media versus normal opinion data. In addition, we developed an algorithm called Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) to assign sentiment strength to adjectives, which can be seen as a constrained spectral clustering. Other than minimizing the clustering objective function, a penalty function formulated from the constraint matrix is also considered in the clustering process. To the best of our knowledge, there is no previous work combining both WordNet and corpus data and defining constraints from ‘but’ and ‘antonym’ relations to guide/improve clustering on sentiment words. In addition, the word sentiment scores generated by our method are verified to correctly indicate the true sentiment strength, which are not shown in previous literature. The contributions of this paper are listed below:

- Construct adjective sentiment dictionaries by combining both WordNet and a large-scale social network data, and show that it outperforms dictionaries by only using WordNet or corpus data.
- Develop the CSNMF algorithm, which can be regarded as a “semi-supervised” clustering, to take advantages of both ‘attraction’ and ‘repulsion’ between adjectives to better assign sentiment strength scores to the adjectives. The precision of the top ranked words are shown to be higher in our experiment.
- Demonstrate our study of Amazon Mechanical Turk (AMT) on social media word polarity, and verify our dictionaries by both General Inquirer Lexicon and human-labeled dictionaries from AMT.
- Compare our method with the existing approaches to demonstrate the effectiveness of our method.

Related Work

Sentiment Analysis generally has two research directions: lexicon-based approaches and machine learning approaches. Lexicon-based approaches use the positions of words, and linguistic analysis to discover the patterns to determine the polarity of opinions (Hu and Liu 2004; Kim and Hovy 2004). Machine learning approaches learn from human-labeled text for sentiment classification (Pang, Lee, and Vaithyanathan 2002; Dave, Lawrence, and Pennock 2003). For lexicon-based approaches, a set of words labeled with sentiments is often required. The work for automatically generating this set of words can be categorized as corpora-based approach (Hatzivassiloglou and McKeown 1997) and thesaurus-based approach. Hatzivassiloglou and McKeown (Hatzivassiloglou and McKeown 1997) first proposed corpora-based word level sentiment analysis. It extends the adjectives by using conjunction rules extracted from a large document corpus. Turney (Turney 2002) first defines a set of positive seed terms and negative seed terms, then searches the target term and seed terms to measure their point-wise mutual information (PMI). The orientation of the target term is the sum of weights of its semantic association with positive seed terms minus that with negative seed terms. For the thesaurus-based word level sentiment analysis, Kim and Hovy (Kim and Hovy 2004) expand seed sentiment words on WordNet with synonym and antonym relations. The polarity of a term is determined by observing the number of its neighbors that are positive or negative. Esuli and Sebastiani (Esuli and Sebastiani 2006) build ternary classifiers on the WordNet synsets, a small set of which are manually labeled and extended into the final training sets. Kamps et al. (Kamps et al. 2004) link terms on the WordNet with synonym relationships to generate a graph, with the polarity of a term computed by measuring its shortest distance to ‘good’ and ‘bad’. Our work differs from theirs: (1) We expand our seed adjective terms using both corpus and WordNet - thus it is corpora-based as well as Thesaurus-based. (2) The relations between terms are distinguished as ‘attractions’ (the ‘and’ conjunctions and synonyms) and ‘repulsions’ (the ‘but’ conjunctions and antonyms). The former serves as a graph edges, and the latter is formulated as

[4] http://www.wjh.harvard.edu/ inquirer/

[5] http://wordnet.princeton.edu/

the constraints/penalties. Each term is assigned both positive and negative polarity scores. (3) We use social network data as our corpus to build the sentiment dictionary.

There are many semi-supervised clustering methods. One important approach is Nonnegative Matrix Factorization (NMF) by penalizing its objective function with constraints. NMF factorizes an input nonnegative matrix into a product of two new matrices with lower rank. A lot of work (Ding et al. 2006; Xu, Liu, and Gong 2003) show the usefulness of NMF for clustering with experiments on document collections. Semi-supervised clustering using nonnegative matrix factorization (Semi-NMF) was first formulated by (Li, Ding, and Jordan 2007). Our proposed Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) algorithm especially designed to cluster undirected graph nodes (adjective words) with constraints, where the input graph matrix and constraint matrix are both symmetric.

Word Sentiment Agreement Study on AMT

Amazon Mechanical Turk (AMT) is a marketplace for *Human Intelligence Tasks* (HITs). AMT has two types of users: providers and workers. Providers pay a small fee to post HITs on AMT, which workers can search and complete to gain money. Providers can reject the work if they are not satisfied with the work quality.

In order for us to obtain a 'ground truth' for our word-level sentiment analysis evaluation, and compare the sentiment agreement rate between words from social media with that from normal opinion data, we conducted a word sentiment agreement study on AMT. We had 7,221 candidate words obtained from the expansion of seed words on both WordNet and Digg (over 90% of words are from Digg). For each word, we asked three distinct workers on AMT to label it as positive, negative, or neutral. The word sentiment labels are merged by a *strict* or *generous* policy. With the strict policy, a target word obtains a sentiment label only when all three annotators agree. With the generous policy, agreement by only a majority of annotators (at least two annotators) is required. To compare with the human agreement study in paper (Kim and Hovy 2004), we show the agreement percentage when two annotators are randomly selected from three in Table 1. Note that we got 58.29% agreement which is much lower than 76.19% obtained by (Kim and Hovy 2004) with the strict policy. Although 93.8% agreement achieved by three annotators with the generous policy looks very satisfactory, the probability of all three annotators assigning different polarities is as low as 22.22% even in the random case. The reasons why the human agreement percentage on words from social network is much lower than that from normal opinion data could be (1) Each job is assigned to three distinct workers on AMT, thus the entire task is completed by a relatively big number of annotators who can be quite different from one another. However, in work (Kim and Hovy 2004), only three annotators are selected and they might share the same features (e.g. same school, same major, etc.), so they may tend to agree with each other more. (2) Annotators from AMT have not done the jobs with high quality. We tried to solve that by filtering out the jobs with low quality (jobs completed too fast, jobs done by workers who often disagree with other workers, etc.). The cleaned annotations are observed to have a reasonable quality.(3) Most importantly, words from social network are more difficult to determine the polarity. The words are extracted from a very wide range of topics, and many of them are informal words withholding different polarities.

	Two Annotators	Three Annotators
Strict	58.29%	40.51%
Generous	58.29%	93.8%

Table 1: Human agreement percentage.

The Proposed Method

In the following, our proposed approach is described in detail. First, we specify how the candidate words are extracted and how to construct the input matrices for our algorithm. Then, we present the proposed model and algorithm.

Adjective Expansion and Input Matrices

We start from 27 positive and 25 negative seed adjective words, only 52 in total. This seed word set can be denoted as S^0. Then, we expand them on WordNet by including their synonyms and antonyms. This results in the 1-level word set S^1 (around 400 words, 165 positive and 216 negative where the positive set is S^1_+ and the negative set is S^1_-), in which the positive word set $S^1_+ = S^0_+ \bigcup Synonym(S^0_+) \bigcup antonym(S^0_-)$, and the negative set is $S^1_- = S^0_- \bigcup Synonym(S^0_-) \bigcup antonym(S^0_+)$. More words can be added by expanding the adjective word set S^1 on WordNet. In our experiment, to guarantee the higher precision, we further crawl only 1 or 2 levels on WordNet because the more levels we go, the weaker sentiment words we will get. The final word set obtained from the WordNet is $S_{WordNet}$. The number of times that a word i and word j appear to be synonyms is denoted as w^{ij}_+. The number of times that a word i and word j appear to be antonym is w^{ij}_- (they are either 0 or 1).

We collected over 2GB Digg data from October 2009 to June 2010. All comments are parsed using Part-Of-Speech (POS) technique, and all pairs of adjectives linked with 'and' and 'but' are extracted (Note that we tried to expand words by more types of conjunction rules (e.g. 'or'), but the final experimental result shows a bad performance). Similarly, we use the word set $S_{WordNet}$ to find all words linked to them in Digg data, and then adding more words by searching a few levels of links. In our case, we only include the words that are at most 3 levels/hops away from the words in $S_{WordNet}$ to ensure the precision. The frequency of the word i and word j being linked with 'and' is represented as d^{ij}_+, and d^{ij}_- is the frequency of the word i and word j being linked with 'but'. The final sentiment word set is denoted as S.

To run our algorithm, two input matrices, the graph matrix X and the constraint matrix C, are needed. They are both nonnegative symmetric matrices with the same cardinality to represent 'attractions' and 'repulsions' between the words in S. The entry x_{ij} in X is the graph edge weight between word i and word j. It is calculated from the synonym relations from WordNet and the ‘and’ conjunction relations, and can be formulated as

$$x_{ij} = w_+^{ij} + Log(d_+^{ij} + 1). \tag{1}$$

We adopt a logarithmic transform of the sum of the linkages to avoid the huge edge weights that exist among frequent words. The constraint matrix C has each non-zero entry value denoting a Cannot-link weight between two adjectives calculated from the antonym relations from WordNet and the ‘but’ conjunction relations. It can be written as

$$c_{ij} = I_{ij} + Log(d_-^{ij} + 1), \tag{2}$$

where $I_{ij} = 1$ if word i is in S_+^1 and word j is in S_-^1 or vice versa, otherwise $I_{ij} = 0$.

CSNMF Model

Once the input matrices are constructed, our next step is to assign positive and negative sentiment scores to the words. We propose a new NMF method called CSNMF for doing this, which has the advantage of considering both associations and repulsion between words. NMF (Lee and Seung 1999) originally from linear algebra is mainly used in pattern recognition and dimensionality reduction. It performs singular value decomposition with non-negative constraints. The NMF fitting algorithm minimizes the Euclidean distance (the least square error) or DL-divergence (I-divergence) between the original matrix and the reconstructed matrix by usually using multiplicative update rules to ensure the nonnegativity. It has been proven that NMF is equivalent (in terms of the equivalent objective functions and the factorization) to Probabilistic Latent Semantic Analysis (PLSA) (Hofmann 2001) when minimizing the KL-divergence objective function (Ding, Li, and Peng 2008). Recently a lot of work (Ding et al. 2006; Xu, Liu, and Gong 2003) show the usefulness of NMF for clustering with experiments on documents collections. The characteristic of NMF is that the entries of the input matrices and the output component matrices are nonnegative. The obvious benefit of nonnegative solutions is that they are easy to interpret, and the clustering quality is not degraded due to additional approximation in the discretization process. NMF usually factorizes an input nonnegative matrix into a product of two new matrices with lower rank by minimizing the following objective function

$$J_{NMF} = ||\mathbf{X} - \mathbf{H}\mathbf{V}^{\mathbf{T}}||_F^2,$$

where $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\mathbf{H} \in \mathbb{R}^{n \times k}$, and $\mathbf{V} \in \mathbb{R}^{m \times k}$. $\parallel \mathbf{X} \parallel_F = \sqrt{\sum_{ij} x_{ij}^2}$ is the Frobenius norm of the matrix $\mathbf{X}$. Entries in matrices $\mathbf{X}$, $\mathbf{H}$, and $\mathbf{V}$ are all nonnegative.

NMF has been extended into various factorization models. One factorization model is to decompose a nonnegative symmetric matrix X with $X = HH^T$, or the weighted version $X = HSH^T$. It is proven to be equivalent to Kernel K-means clustering and the Laplacian-based spectral clustering (Ding, He, and Simon 2005). Our CSNMF algorithm further extends the symmetric nonnegative matrix factorization by adding constraints on graph nodes. It can be regarded as constraint spectral clustering. CSNMF iteratively cuts the graph of adjectives into positive and negative sets, where each adjective word is assigned a positive score and a negative score. The following is the objective function of CSNMF needed to be minimized

$$J_{CSNMF} = ||\mathbf{X} - \mathbf{H}\mathbf{S}\mathbf{H}^{\mathbf{T}}||_F^2 + \alpha \mathbf{Tr}(\mathbf{H}^{\mathbf{T}}\mathbf{C}\mathbf{H}), \tag{3}$$

where $\mathbf{X} \in \mathbb{R}^{n \times n}$, $\mathbf{C} \in \mathbb{R}^{n \times n}$, and α are the input. $\mathbf{X}$ is a nonnegative symmetric graph matrix, and can be derived from Equation 1. $\mathbf{C}$ is the constraint weighting matrix as shown in Equation 2. α is the input parameter to adjust the influence of the penalty term $\mathbf{Tr}(\mathbf{H}^{\mathbf{T}}\mathbf{C}\mathbf{H})$. The output is $\mathbf{H} \in \mathbb{R}^{n \times k}$ and $\mathbf{S} \in \mathbb{R}^{k \times k}$. $\mathbf{H}$ is the class indicator matrix where h_{il} can be seen as the ‘weight’ or ‘load’ of the object i in class l. In our case, it indicates the probability of the word i belonging to sentiment class l (positive or negative thus k is 2). Thus, $\mathbf{H}$ provides the sentiment scores including the positive score and negative score for each word. $\mathbf{S}$ provides the extra degrees of freedom that allows $\mathbf{H}$ to be closer to the form of cluster indicators.

Equation 3 contains two parts. If we set α to zero, Equation 3 remains the first part $||\mathbf{X} - \mathbf{H}\mathbf{S}\mathbf{H}^{\mathbf{T}}||_F^2$, which minimizes the reconstruction error, and can be seen as normal spectral clustering. The second part $\mathbf{Tr}(\mathbf{H}^{\mathbf{T}}\mathbf{C}\mathbf{H})$ is the penalty term that we aim to minimize as well. We add this penalty term to enforce the Cannot-link constraint. Suppose word i and word j should not be in the same sentiment cluster with the penalty weight c_{ij}, then we want to minimize $c_{ij}(H^T H)_{ij}$. Thus the sum of penalties is $\sum_{\mathbf{ij}} \mathbf{c}'_{\mathbf{ij}}$ $(\mathbf{H}^{\mathbf{T}}\mathbf{H})_{\mathbf{ij}} = \mathbf{Tr}(\mathbf{H}^{\mathbf{T}}\mathbf{C}'\mathbf{H})$.

Algorithm for CSNMF

Since the negative *loadings* are hard to explain, all output component matrices are constrained to have positive entries. We employs the nonnegative multiplicative least square update algorithm (Lee and Seung 1999) to minimize the objective function in Equation 3, in which updating one component while fixing others.

Input: $\mathbf{X}$, $\mathbf{C}$, and α.

Initialization: Initialize $\mathbf{S}$ with random nonnegative entries while putting larger values on the diagonal. Since we already know the polarity of the seed words, $\mathbf{H}$ is initialized such that $h_{i1} = 1$ and $h_{i2} = \sigma$ if word $i \in \mathbf{S}_+^{\mathbf{1}}$, where $\sigma \ll 1$ (suppose the column 1 in $\mathbf{H}$ contains positive ‘weight’ and the column 2 has negative ‘weight’). Similarly $h_{i2} = 1$ and $h_{i1} = \sigma$ if word $i \in \mathbf{S}_-^{\mathbf{1}}$. The rest entries of $\mathbf{H}$ are randomly initialized.

Update H: Using the nonnegative multiplicative least square algorithm, update $\mathbf{H}$ while fixing $\mathbf{S}$.

$$h_{ij} = h_{ij} \frac{(\mathbf{XHS})_{\mathbf{ij}}}{(\mathbf{H}\mathbf{S}\mathbf{H}^{\mathbf{T}}\mathbf{H}\mathbf{S} + \alpha\mathbf{C}\mathbf{H})_{\mathbf{ij}}}. \tag{4}$$

Update S: Update $\mathbf{S}$ while fixing $\mathbf{H}$

$$s_{ij} = s_{ij} \frac{(\mathbf{H}^{\mathbf{T}}\mathbf{X}\mathbf{H})_{\mathbf{ij}}}{(\mathbf{H}^{\mathbf{T}}\mathbf{H}\mathbf{S}\mathbf{H}^{\mathbf{T}}\mathbf{H})_{\mathbf{ij}}}. \tag{5}$$

The above rules are updated iteratively until convergence. It can be proven to be correct and convergent. For example, to derive Equation 4, we need to introduce the Lagrangian multiplier λ_{ij} to make each entry of H nonnegative. Let $L = J_{CSNMF} + \sum_{ij} \lambda_{ij} H_{ij}$. The KKT condition is

$$\frac{\partial L}{\partial H_{ij}} = \frac{\partial J_{CSNMF}}{\partial H_{ij}} + \lambda_{ij} = 0, \; and \; \lambda_{ij} H_{ij} = 0.$$

Thus the KKT condition leads to the fixed point relation:

$$(-XHS + HSH^T HS + \alpha CH)_{ij} H_{ij} = 0.$$

The above can be reformulated to be Equation 4. The further proof of the correctness and convergence of the algorithm will not be specified here due to space limitations. The computational complexity of the above algorithm is $O(n \times n \times k)$. The sentiment scores are derived from matrix H. For example, the positive and negative sentiment scores of word i are $h_{i1}/\max_j(h_{j1})$ and $h_{i2}/\max_j(h_{j2})$.

Experiment and Evaluation

In this section, we will first describe the data set, then the experimental results and the comparative study will be presented together with the quantitative evaluations and discussions.

Data Set

We use the Digg API [6] to crawl $255,492$ stories (each of which with at least 1 comment) from October 1, 2009 to June 30, 2010. Each story is composed of an external link, summary, author, topic, title, and time stamp. The number of associated comments and replies are $1,813,691$ and $1,345,138$ respectively. We also collected users, friend relationships, and other kinds of information. Digg has diversified topics, and Digg Stories are categorized hierarchically into 10 categories: *Business, Entertainment, Game, Lifestyle, Offbeat, Politics, Science, Sports, Technology, World News*. Each category contains topics such as 'linux-unix', 'music', 'environment', 'general-sciences', 'people', 'political-opinion', 'movies', and 'pets-animals', etc., with 51 topics in total. In our experiment, we split the data into two sets. One set consists of all 9-month comments and replies, called Digg9. The other set contains 6-month comments and replies (from October 1, 2009 to March 31, 2010), called Digg6.

We use Natural Language ToolKit (NLTK) [7] to perform *part-of-speech* (POS) *tagging* on all the Digg comments and replies. Pairs of adjectives that are linked by 'and' or 'but' are extracted. In total, there are 19241 'and' pairs (9706 distinct pairs), 509 'but' pairs (371 distinct pairs) from Digg6. For Digg9, there are 50241 'and' pairs (24061 distinct pairs), 1328 'but' pairs (1042 distinct pairs). Our seed word set has 27 positive words and 25 negative words.

Evaluation

Ground-truth Dictionaries We evaluate our automatic generated sentiment dictionary against two dictionaries. The first is the General Inquirer (GI), which is often used by researchers to evaluate their own dictionary. However, we found that the GI dictionary does not correctly tag some informal sentiment words. For example, 'cool' is tagged as negative. In addition, some informal words are not included in GI. Thus, we decide to obtain another gold standard labeled by humans. We posted 7221 adjectives generated from our dictionary onto AMT, and obtained their sentiment labels according to the 'strict' and 'generous' policy specified in previous sections. With the 'strict' policy, 1761 words (849 positive, 912 negative, and 1164 neutral) have valid sentiment labels (neutral words are omitted). On the other hand, with the 'generous' policy, 3847 words (1943 positive, 1904 negative, and 2930 neutral) have valid sentiment labels.

Evaluation Results **Evaluation by GI:**We generate sentiment dictionaries under various level settings on both WordNet and Digg conjunction links, and then evaluate them against GI in terms of precision and recall. The experimental results are shown in Table 2. They illustrate the resulting dictionaries generated from 6 month and 9 month Digg data. In Table 2, each column indicates a level setting for the corresponding dataset. For instance, W1C0 means that the dictionary is generated by extending the seed word set one level on WordNet, and no expansion on conjunction links. W1C1 is the dictionary generated by expanding the seed words one level on WordNet and one level on conjunction links. The observations we can obtain from this experimental result are

- The more levels we crawl, the less precision and higher recall we get.
- The expansion on WordNet without the 'assistance' from conjunction links might yield a lower precision as well as a lower recall, as seen from W2C0 compared to W1C1 from Digg6 (we compare them because they have the similar number of words).
- The expansion on conjunction links from corpus also produces dictionaries with lower precisions and lower recalls compared to dictionaries by expanding on both WordNet and the corpus (e.g. W0C3 and W0C4 compared to W3C3 from Digg6, W0C2 and W0C3 compared to W1C2 from Digg9).

As specified in the last section, each word can be ranked based on its sentiment scores (from output H). Some examples words, together with their positive and negative polarity scores, are listed in Table . Suppose word i has the positive sentiment score h_i^{pos} and the negative sentiment score h_i^{neg}, words are ranked by $\max(h_i^{pos}, h_i^{neg})$ in descending order. The top ranked words have higher confidence to be assigned to their sentiment classes. For example, the top ranked 1000 words of W3C3 from Digg6 have 91.33% precision and 12.33% recall, the top ranked 2000 words have 86.87% precision and 17.94%, and the top 3000 words have 83.62% precision and 22.57% recall (evaluated by GI).

In order to make sure that words are correctly labeled on AMT, we evaluate our AMT word labels by GI. We measure the precisions of 'strict' and 'generous' labels obtained from AMT. They get fairly high precisions, 98.85% and 95.62% respectively, implying that the labels from AMT should not contain much noise. Note that although high consistency is

[6] http://developers.digg.com/

[7] http://www.nltk.org/

	Digg6										Digg9			
	W1C0	W2C0	W1C1	W1C2	W1C3	W2C2	W2C3	W0C3	W0C4	**W3C3**	W1C1	W0C2	W0C3	**W1C2**
Precision	91.86%	82.76%	86.54%	81.1%	79.2%	79.87%	80.99%	76.53%	73.6%	**79.73%**	84.92%	74.78%	73.65%	**77.73%**
Recall	4.71%	9.53%	14.42%	21.88%	23.3%	20.91%	25.5%	25.91%	26.35%	**28.65%**	22.16%	30.68%	32.95%	**32.35%**
# Words	385	1135	1262	2671	3793	3466	3793	3632	3769	**4464**	2518	5703	6993	**6364**

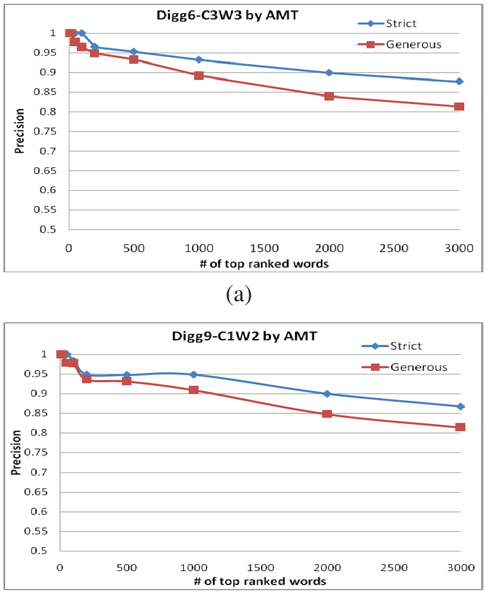
Table 2: Dictionaries generated from Digg6 and Digg9 are evaluated by GI.

Word	Positive Score	Negative Score
'cocking'	0.902	0
'sassy'	0.8836	0
'new'	0.1511	0.0116
'yucky'	0	0.9095
'irksome'	0	0.8994
'long-winded'	0	0.8895
'dark'	0.0228	0.0297

Table 3: Example words with sentiment scores.

observed between GI and AMT labels, this high precision exists in the overlapping word set between them. Some informal words are not covered by GI, thus AMT labels are a more appropriate ground truth for our experimental result evaluation. Because W3C3 from Digg6 and W1C2 from Digg9 have higher recall without losing too much on precision, we combine these two dictionaries to further evaluate them by manual labels obtained from AMT. The 7221 adjectives we posted on AMT is the union of W3C3 from Digg6 and W1C2 from Digg9.

Evaluation by AMT: As specified before, we obtained sentiment labels including *positive, negative, and neutral* for 7221 adjectives according to both 'strict' policy and 'generous' policy. However, our method does not specifically identify *neutral* words, thus neutral words are omitted for evaluation by AMT.



(a)

(b)

Figure 1: The performance of dictionary generation from Digg6 and Digg9 are evaluated by AMT.

Manual Dict	SNMF	CorpusConj	WordNetDis	**CSNMF**
AMT 'strict'	79.99%	78.48%	67.82%	**84.51%**
AMT 'generous'	74.89%	74.77%	65.45%	**78.15%**
GI	77.25%	76.82%	66.87%	**81.18%**

Table 4: The precision of our automatic generated dictionary C3W3 from Digg6 is compared with dictionaries generated using existing approaches.

In Figure 1, the precision of C3W3 from Digg6 and C1W2 from Digg9 are plotted in subfigures 1(a) and 1(b) respectively. The x axis indicates the number of top ranked words in the dictionary that are extracted for evaluation. The y axis is the precision ranging from 0 to 1. It can be observed that

- the precision of the dictionary evaluated by AMT labels with 'strict' policy is always above that with 'generous' policy since the 'strict' AMT labels have less noise or mislabels.
- the precision decreases as the number of top ranked words increases. It means that the sentiment scores generated by our proposed algorithm does reflect the 'true' scores. The higher score a word receives, the higher sentiment strength it has to make people be certain about its label. Thus, the words with higher sentiment scores have higher probabilities to be correctly labeled.
- the precision values of C3W3 from Digg6 and C1W2 from Digg9 are almost equal when the number of extracted top ranked words is the same. This makes sense because Digg6 is the subset of Digg9. Although the edge values and the constraint values are different if you look at a specific pair of adjectives, the overall distribution of edge and constraint weights should be very similar. The newly added words from Digg9 do not have obvious polarity, so the words from Digg6 are adequate for our sentiment analysis task.

Comparative Study

CSNMF is compared with the following three approaches:

1. Symmetric Nonnegative Matrix Factorization (SNMF) proposed by Ding et al. in (Ding, Li, and Peng 2008). CSNMF adds constraints to SNMF to penalize the breakage of antonym and 'but' conjunction rules.
2. The method proposed by Hatzivassiloglou and McKeown in (Hatzivassiloglou and McKeown 1997) (we call it CorpusConj). The weight of 'but' conjunctions is regarded as dissimilarity embedded in the graph matrix. The entry is ranging from 0 to 1. The neutral similarity is 0.5 when

two words have no links. Two words with more different-orientation links ('but' conjunction links and antonym links) than same-orientation links ('and' conjunction links and synonym links) will have the link weight less than 0.5, and vice versa.

3. The approach proposed by Kamps et al. in (Kamps et al. 2004) (we call it WordNetDis). The polarity of a word is determined by measuring its shortest distance to 'good' and 'bad'. We extract the words that are contained by WordNet from our dictionary for comparison in our experiment.

The above algorithms and our proposed algorithm are all tested and compared on Digg6 dataset. In order to have a fair comparison, the words are extracted from both WordNet and the corpus though the existing approaches are proposed to run only on WordNet or the corpus. (From the experimental results shown in the previous section, the combination of WordNet and the corpus achieved better results). Thus their recalls are equal by running on the same word set except WordNetDis.

The comparison experimental results are listed in table 4 in terms of precisions. We use three ground-truth dictionaries to verify them: AMT 'strict' dictionary, AMT 'generous' dictionary, and GI dictionary. It can be observed that our proposed method CSNMF outperforms all of them, where the improvement is statistically significant according to paired t-test ($p < 0.05$). Note that except for WordNetDis, the other methods SNMF, CorpusConj, and CSNMF obtain the same recalls, which are 0.5241 for AMT 'strict', 0.4949 for AMT 'generous', and 0.2605 for GI. WordNetDis obtains comparably very low recalls - 0.247, 0.2287, and 0.1328, almost half the recalls that the other methods get. 1845 words are assigned 'Neutral' because their shortest distances to 'good' are equal to their shortest distances to 'bad'.

Lexicon Classification using Automatic Generated Sentiment Dictionary

In addition to evaluating our automatic generated sentiment dictionary by GI and word labels from AMT, we want to know the performance of the lexicon classification using our dictionary on Digg data compared with using the manually labeled words from AMT. In order to evaluate the lexicon classification, we need 'ground-truth' sentiment labels of a set of example comments from Digg. We randomly select 90 stories from each topic in Digg. For each story, two comments with at least 10 words and at most 100 words are randomly chosen. In total, we got 8620 comments (some stories do not have two comments) as our example dataset. Then, we created HITs on AMT that each consisted of 2 stories together with their 4 comments. Each comment had 3 distinct annotators from AMT to label it to be *positive*, *negative*, *mixed*, or *neutral*. Similar to the word sentiment annotation on AMT, we obtained two set of labels by using 'strict' policy and 'generous' policy. The results of human agreement on these 8620 Digg comment sentiment are illustrated in Table 5. We can see that the agreement is as low as 25.37% by 'strict' policy. If we randomly select two annotators from three for each comment, the probability of human agreement is 43.68%, which is much lower than the probability of human agreement (82%) reported in (Wilson, Wiebe, and Hoffmann 2005b).

Policy	# Pos	# Neg	# Mixed	# Neut
Strict	10.3%	12.9%	1%	1.8%
Generous	28.2%	36.5%	7.8%	11.2%

Table 5: Human Agreement on 8620 Digg comment sentiment from AMT with 3 annotators.

Since our purpose is to classify the comments into only positive and negative classes, we remove the comments with 'Neutral' or 'Mixed' labels. Therefore, we have 1761 comments with labels by the 'strict' policy and 3847 labeled comments by the 'generous' policy. In this paper, we apply a very simple lexicon classification approach to verify the effectiveness of our dictionary. The steps are listed as follows:

1. Get the dictionary, including words and their POS tags (very small set of verbs and nouns are added).
2. Parse comments into POS tags. The words with the right tags are counted. For instance, 'good' is positive when it is an adjective.
3. Calculate the sentiment score of each comment. When using manually labeled words for lexicon classification, the sentiment scores are calculated by simply counting the number of positive words and negative words in the comment. As for our automatic generated sentiment dictionary, the sentiment score of a comment is the sum of positive polarity scores minus negative polarity scores of the contained words.
4. Negation is also considered. A sentence with negation will add a minus sign to its sentiment score.

Three adjective sentiment dictionaries (W1C2 from Digg9, manually labeled dictionary from AMT by the 'strict policy', and manually labeled dictionary from AMT by the 'generous' policy, together with a small set of verbs and nouns) are used to classify 1761 comments (labeled by the 'strict' policy) into the positive class and negative class. The classification performance is measured in terms of precision, recall, and the F score as shown in Table 6. Generally, W1C2 performs worse than ground-truth AMT dictionary on precision, which is certainly expected. However, the W1C2 obtains much higher recall, it also outperforms AMT 'generous' dictionary in terms of F measure, and only a little worse than AMT 'strict' dictionary.

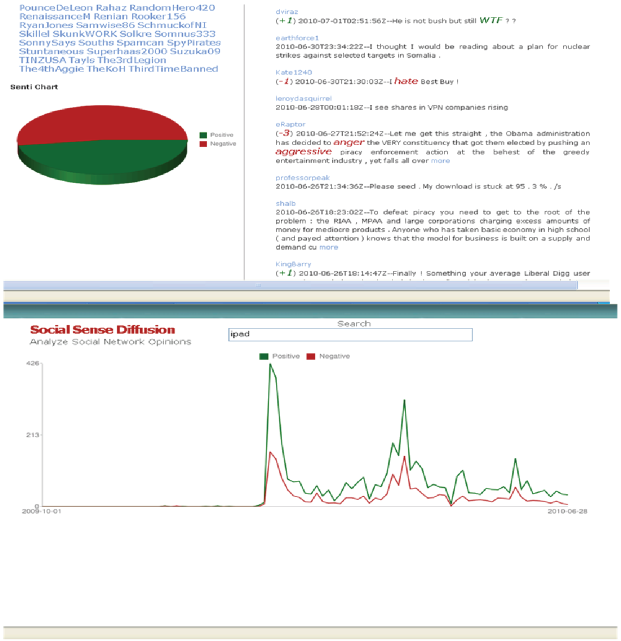
The proposed method is implemented in our research prototype, called "social sense diffusion", to help understand how positive and negative messages about a brand/product propagate on the real social network. Two screenshots of the prototype related to sentiment analysis are posted in Figure 2. They are the dive-in page when a particular digg story is clicked, and the sentiment trend page when searching on a particular product, brand, people, etc.

Conclusion

In this paper, we targeted to generate adjective dictionary automatically by combining both WordNet and the corpus

# words	W1C2 Auto Dictionary										AMT ‘strict’	AMT ‘generous’
	6343	3000	2000	1000	500	200	100	50	14	10	1761	3847
Precision	**0.7208**	0.7294	0.7341	0.741	0.7667	0.7979	0.8014	0.7984	0.8069	0.8074	0.81	0.7698
Recall	**0.8941**	0.7939	0.7657	0.7407	0.6598	0.5366	0.5304	0.5202	0.49	0.4859	0.6675	0.7223
F1	**0.6974**	0.6821	0.6793	0.6785	0.676	0.6599	0.6599	0.6552	0.6504	0.6494	0.7069	0.6949

Table 6: The performance comparisons of automatic generated dictionary W1C2 and AMT manually labeled dictionaries on the lexicon classification.


PounceDeLeon Rahaz RandomHero420
RenaissanceM Renian Rooker156
RyanJones Samwise86 SchmuckofNI
Skillel SkunkWORK Solkre Somnus333
SonnySays Souths Spamcan SpyPirates
Stuntaneous Superhaas2000 Suzuka09
TINZUSA Tayls The3rdLegion
The4thAggie TheKoH ThirdTimeBanned
Senti Chart
Positive
Negative
Social Sense Diffusion
Analyze Social Network Opinions
Search
ipad
Positive
Negative
426
213
0
2009-10-01
2010-06-28

dviraz
(+1) 2010-07-01T02:51:56Z--He is not bush but still WTF ? ?

earthforce1
2010-06-30T23:34:22Z--I thought I would be reading about a plan for nuclear strikes against selected targets in Somalia .

Kate1240
(-1) 2010-06-30T21:30:03Z--I hate Best Buy !

leroydasquirrel
2010-06-28T00:01:18Z--I see shares in VPN companies rising

eRaptor
(-3) 2010-06-27T21:52:24Z--Let me get this straight , the Obama administration has decided to anger the VERY constituency that got them elected by pushing an aggressive piracy enforcement action at the behest of the greedy entertainment industry , yet falls all over more

professorpeak
2010-06-26T21:34:36Z--Please seed . My download is stuck at 95 . 3 % . /s

shalb
2010-06-26T18:23:02Z--To defeat piracy you need to get to the root of the problem : the RIAA , MPAA and large corporations charging excess amounts of money for mediocre products . Anyone who has taken basic economy in high school (and payed attention) knows that the model for business is built on a supply and demand cu more

KingBarry
(+1) 2010-06-26T18:14:47Z--Finally ! Something your average Liberal Digg user

Figure 2: Two screenshots of our prototype.

from social network data. We proposed a new algorithm called CSNMF to cluster word nodes connected by various relations and constraints into the positive class and the negative class. In our experiments, We show that combining links from both WordNet and the corpus to generate sentiment dictionaries does outperform using only one of them. Comparisons between our method and some existing approaches show that our performance improvement is statistically significant. Our proposed method can also assign the sentiment strength score to each word in the dictionary, in which the top ranked words yield better precision. Finally, our dictionary shows comparable performance in determining the sentiment score for social network comments, compared to the human labeled ground-truth dictionaries.

References

Dave, K.; Lawrence, S.; and Pennock, D. M. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW ’03*, 519–528.

Ding, C.; Li, T.; Peng, W.; and Park, H. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD ’06*, 126–135.

Ding, C.; He, X.; and Simon, H. D. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the SIAM Data Mining Conference*, 606–610.

Ding, C.; Li, T.; and Peng, W. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computation Statistics and Data Analysis* 52(8):3913–3927.

Esuli, A., and Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC06*, 417–422.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. In *EACL ’97*, 174–181.

Hofmann, T. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning* 42(1-2):177–196.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *KDD ’04*, 168–177.

Kamps, J.; Marx, M.; Mokken, R. J.; and Rijke, M. D. 2004. Using wordnet to measure semantic orientation of adjectives. In *National Institute for*, 1115–1118.

Kim, S.-M., and Hovy, E. 2004. Determining the sentiment of opinions. In *COLING ’04*, 1367.

Lee, D. D., and Seung, S. H. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788–791.

Li, T.; Ding, C.; and Jordan, M. I. 2007. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM ’07*, 577–582.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP ’02*, 79–86.

Turney, P. D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL ’02*, 417–424.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005a. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT ’05*, 347–354.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT ’05*, 347–354.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *SIGIR ’03*, 267–273.