Extracting Diverse Sentiment Expressions with Target-dependent Polarity from Twitter

Lu Chen[1] chen@knoesis.org
Wenbo Wang[1] wenbo@knoesis.org
Meenakshi Nagarajan[2] MeenaNagarajan@us.ibm.com
Shaojun Wang[1] shaojun.wang@wright.edu
Amit P. Sheth[1] amit@knoesis.org

[1]Kno.e.sis Center, Wright State University
Dayton, OH 45435 USA

[2]IBM Almaden Research Center
San Jose, CA 95120 USA

Abstract

The problem of automatic extraction of sentiment expressions from informal text, as in microblogs such as tweets is a recent area of investigation. Compared to formal text, such as in product reviews or news articles, one of the key challenges lies in the wide diversity and informal nature of sentiment expressions that cannot be trivially enumerated or captured using predefined lexical patterns. In this work, we present an optimization-based approach to automatically extract sentiment expressions for a given target (e.g., movie, or person) from a corpus of unlabeled tweets. Specifically, we make three contributions: (i) we recognize a diverse and richer set of sentiment-bearing expressions in tweets, including formal and slang words/phrases, not limited to pre-specified syntactic patterns; (ii) instead of associating sentiment with an entire tweet, we assess the target-dependent polarity of each sentiment expression. The polarity of sentiment expression is determined by the nature of its target; (iii) we provide a novel formulation of assigning polarity to a sentiment expression as a constrained optimization problem over the tweet corpus. Experiments conducted on two domains, tweets mentioning movie and person entities, show that our approach improves accuracy in comparison with several baseline methods, and that the improvement becomes more prominent with increasing corpus sizes.

Introduction

Twitter provides a convenient and instant way for people to share their sentiments on various topics anytime and anywhere. The ever growing volume of Twitter messages (i.e., tweets) offers a wealth of data that can be used for learning and understanding people's sentiment. There have been several studies and applications analyzing sentiments in tweets. While most work has focused on classifying tweets as positive, negative or neutral, few approaches attempt to identify the actual expressions of sentiment in tweets. Compared to the overall sentiment polarity, sentiment expressions usually provide more fine-grained information, and can be useful for applications such as opinion question answering, opinion summarization and opinion retrieval. Moreover, they can also be used for sentiment classification task, in both the lexicon-based classifier (e.g., used as the sentiment lexicon) and the machine learning classifier (e.g., providing useful features).

In this paper, we focus on the problem of extracting sentiment expressions and assessing their polarities for a given target from a corpus of unlabeled tweets. We define the *sentiment expression* as a word or phrase that attributes a sentiment polarity on a target in the text. To understand the challenges in this problem, consider the following example tweets, in which we denote potential sentiment expressions and their targets in italic and boldface, respectively.

1. Saw the movie **Friends With Benefits**. So *predictable*! I *want my money back*.
2. Alright enough of **Taylor Swift**. She is *gud* but I am still *not a fan*.
3. **The King's Speech** was *bloody brilliant*. **Colin Firth and Geoffrey Rush** were *fantastic*!

First, sentiment expressions in tweets can be very diverse. They vary from single words (e.g., "*predictable*", "*fantastic*") to multi-word phrases of different lengths (e.g., "*want my money back*", "*bloody brilliant*"), and can be formal or slang expressions, including abbreviations and spelling variations (e.g., "*gud*"). Our quantitative study (refer to Table 1 in the section of Experiments) of 3,000 tweets shows that 45.76% and 28.62% of the sentiment expressions in the movie domain and the person domain, respectively, are multi-word phrases. The phrasal expressions vary from 2 to 9 words long. Furthermore, there is a considerable number of sentiment expressions that are slang (15.25% and 11.59% in the movie and the person domain, respectively). The extraction algorithm should be able to deal with such diversity and identify the sentiment expressions.

Second, the polarity of a sentiment expression is sensitive to its target. For example, "*predictable*" in example 1 is negative towards its target - movie "**Friends With Benefits**," while it could indicate positive sentiment regarding other targets such as stocks. The algorithm should be capable of extracting the sentiment expressions associated with the target and assessing their target-dependent polarities.

Previous approaches for extracting sentiment expressions from formal text (e.g., product reviews, or news articles) do not translate effectively to tweets. They usually focus on the words or phrases belonging to certain linguistic patterns,

e.g., adjectives or an adjective followed by a noun. However, the diverse forms of sentiment expressions in tweets cannot be fully captured by a few predefined patterns. Moreover, the informal nature of language usage and writing style in tweets poses considerable difficulties for part-of-speech taggers and parsers, which typically rely on standard spelling and grammar.

In this work, we present an optimization-based approach to automatically extract sentiment expressions associated with a given target in a corpus of tweets. This approach not only captures the diversity of expressions, but also assesses their target-dependent polarities. Specifically, it consists of four main steps: 1) we obtain a comprehensive set of *root words* from both traditional and slang lexical resources; 2) to identify a diverse and richer set of sentiment expressions, we let the candidate expressions be any on-target n-grams that contain at least one root word; 3) we construct two networks to encode the consistency and inconsistency relations of the candidate expressions over the tweet corpus; and 4) finally combine the information encoded in the networks into an optimization model to estimate the target-dependent polarity of each candidate.

We conduct experiments on two tweet corpora, with 168K and 258K tweets on the movie and the person domain, respectively. The results show that our approach can effectively extract diverse sentiment expressions and assess their target-dependent polarities. The advantage of our approach is demonstrated through comparison with several baselines. It achieves absolute F-measure gains of 8.49%-21.09% and 8.58%-25.12% in the movie and the person domain, respectively. Moreover, higher gains came with larger corpora. To demonstrate how this work can benefit sentiment classification application, we apply the extracted sentiment expressions to classify tweets into different sentiment categories. The results show that the diverse and richer set of sentiment expressions with target-dependent polarity extracted by our approach improves the sentiment classification of tweets.

RELATED WORK

Sentiment Expression Extraction

Extraction of sentiment words has been explored in many studies, either as the main task, e.g., sentiment lexicon construction (Hatzivassiloglou and McKeown 1997; Kanayama and Nasukawa 2006; Qiu et al. 2009; Lu et al, 2011; Peng and Park 2011), or as the subtask of sentence or document level sentiment analysis (Hu and Liu, 2004; Choi et al, 2009). These efforts typically consider words of some specific part-of-speech (e.g., adjectives or verbs) as candidates.

Fewer efforts have focused on extracting phrasal sentiment expressions. Turney (2002) uses a few part-of-speech patterns (e.g., "JJ JJ" for two consecutive adjectives) to extract two-word phrases, and estimates the polarities of these phrases using point-wise mutual information by querying a search engine. Some studies (Yi et al. 2003; Kamal and Hurst 2006) identify the phrasal expressions by syntactic parsing, and estimate the polarity of a phrase in a pattern-based manner, using component words from a predefined sentiment lexicon. However, as expressions in tweets vary in length and variety, it is almost impossible to capture such diversity using predefined patterns. Moreover, the predefined lexicon is usually too general to provide target-dependent polarity. There is also some work in applying supervised methods to identify contextual polarity of sentiment phrases (Wilson et al. 2005; Agarwal et al. 2009), but the phrases are manually recognized. Velikovich et al. (2010) extract n-grams as candidates and apply a graph propagation framework to build sentiment lexicon from the web documents. However, they have not considered the target dependence of the sentiment expressions.

While existing research has been focused more on the formal text, very few studies explore the issue of slang. Gruhl et al. (2010) use a sentiment lexicon which is built upon Urban Dictionary[1] (UD) to identify sentiment words (especially slang words) from user comments. Inspired by their work, we also exploit UD for identifying slang expressions, but our work is not limited to the words in UD. In addition, we assess the target-dependent polarity of expressions via an optimization model over the corpus, instead of mining their general polarity from UD.

Sentiment Analysis of Microblogs

Many studies adopt a supervised approach to classify tweets as positive, negative or neutral. Besides manual annotation, efforts have explored different ways of obtaining training data from Twitter. For example, Barbosa and Feng (2010) obtain labeled data from a few sentiment detection websites over Twitter. Some studies (Davidov et al. 2010; Kouloumpis et al. 2011) leverage the hashtags and emoticons in tweets for building training data. Zhang et al. (2011) use a lexicon-based method to perform sentiment classification with high precision, and then apply a supervised classifier to improve the recall using the training examples provided by the previous lexicon-based approach. Jiang et al. (2011) take the sentiment target into consideration, and classify tweets according to whether they contain positive, negative or neutral sentiments about a given target.

To the best of our knowledge, automatic extraction of sentiment expressions associated with given targets has not been investigated for microblogs. The work presented in this paper can be used to identify lexical knowledge features (e.g., the count of positive/negative expressions) that can benefit existing sentiment classification systems.

The Proposed Approach

Let Δ be a corpus of tweets mentioning a given target T. Without loss of generality, T can be an individual topic or a set of topics of the same type in the domain of interest (e.g., a specific restaurant/movie, or a set of restaurants/movies). Spotting the target in text is not the focus of this paper, so we assume that the target has been identified in the tweets of Δ. Our objective is to extract sentiment expressions associated with T from Δ, and assign each extracted expression its target-dependent polarity, i.e., positive or negative.

The proposed approach can be summarized as follows. First, we collect a set of sentiment-bearing root words,

[1] http://www.urbandictionary.com/

which is then used to extract candidate expressions. Then the candidate expressions are connected by their consistency and inconsistency relations (consistent if two expressions are positive or both are negative) in two networks. Finally, based on these two networks, the polarities of the candidate expressions are estimated using an optimization model.

Root Words and Candidate Expressions

We define a root word as a word that is considered sentiment-bearing in the general sense. Many existing general-purpose sentiment lexicons provide such words, in which each word is assigned a prior polarity without regarding to any specific target. A sentiment expression usually contains at least one root word, but its polarity (especially the target-dependent polarity) is not necessarily relevant to the prior polarity of the root words it contains. We use root words for candidate selection, but we assess the target-dependent polarity of the candidates from the tweet corpus.

Collecting Root Words We build a comprehensive set of root words containing both formal and slang words. Formal words are collected from the general-purpose sentiment lexicons. One such lexicon is SentiWordNet[2], in which each synset of WordNet is assigned a *PosScore* and a *NegScore* according to its positivity and negativity. We collect the words with the *PosScore* or *NegScore* higher than 0.75, or the difference between *PosScore* and *NegScore* higher than 0.50 from SentiWordNet. We also incorporate all the 8,221 words from the MPQA[3] subjective lexicon. Another resource incorporated is the General Inquirer[4], from which we collect 1,915 positive words in the *Positiv* category and 2,291 negative words in the *Negativ* category.

Slang words are collected from Urban Dictionary (UD). UD is a popular online slang dictionary with definitions written and voted on by users. In addition to the glossary definitions, each word defined in UD is associated with a list of related words to interpret the word itself. For example, the word "*rockin*" has the following related words in UD: "*awesome, cool, sweet, rock, rocking, amazing, hot, etc.*"

We employ a propagation algorithm that leverages the related word connections to identify the sentiment-bearing slang words from UD. The algorithm starts with a seed word set S^0, which consists of 133 positive and 130 negative sentiment words. These seed words are manually selected, and the polarities of these words are always positive or negative regardless of the targets, e.g., "excellent" or "nasty".

At the beginning, the algorithm initializes a query set Q by including all the seed words from S^0. For a word w in Q, the algorithm queries UD to obtain its related word list. The first ten related words in the list along with w itself are treated as a "document." A frequency matrix is created to record the frequency of the co-occurrence of any pair of words in any document. This matrix is updated with every newly obtained document from UD. Q is also updated by removing w and including its related words in the document. Only the words that have not been added to Q can be added to Q. This process is recursively repeated until Q becomes empty. As the next step, the algorithm identifies a positive/negative slang word according to the dominant polarity of the top five sentiment words (in SentiWordNet, MPQA or GI) that most frequently co-occur with it in the frequency matrix, and add them into the root word set. For example, the word "*rockin*" most frequently co-occurs with sentiment words "*amazing, sexy, sweet, great, awesome*". Since all of the five words are positive, the word "*rockin*" is identified as positive and added to the root word set.

Using this algorithm, a total of 3,521 slang words are collected from UD. Together with words from SentiWordNet, MPQA and GI, the root word set contains 13,606 words, including 4,315 positive words, 8,721 negative words and 570 neutral words. We denote the root word set as Γ.

Extracting Candidate Expressions To extract candidate sentiment expressions associated with the target T from the tweets in Δ, we first identify the root words that act on the target T from each tweet. Specifically, for each tweet in Δ, we use SentParBreaker[5] to perform sentence splitting, and parse each sentence using Stanford Parser[6] to get the dependency relations of words. After stemming, we spot all the root words in the tweet based on Γ. A root word is selected as "on-target" if (1) there is a dependency relation between the word and the target, or (2) the word is proximate to the target (in the experiments, we specify it as within four words distance). The dependency relation and proximity are two widely used ways to determine the association between the sentiment expression and the target (Kessler and Nicolov 2009). Unlike some other studies (Qiu et al. 2009) that limit the dependency relations to some specific types (e.g., *mod* for modifier), we relax the dependency relations to any type to avoid missing proper expressions due to the informal language usage in tweets. After selecting the on-target root words, we extract all the n-grams that contain at least one selected root word as candidates. We limit the n-grams up to an empirically observed threshold length ($length <= 5$) in the experiments.

Inter-Expression Relations

In this step, we connect the candidate expressions via two types of inter-expression relations – consistency relation and inconsistency relation, denoting whether the sentiments of two expressions are consistent (e.g., both are positive or both are negative) or inconsistent (e.g., one is negative and the other is positive) in the tweets of Δ. Let c_i and c_j be two candidate expressions in one tweet. The algorithm for identifying their relations are as following.

Identifying Inconsistency Relations: Generally, a sentiment expression is inconsistent with its negation; two sentiment expressions linked by contrasting conjunctions are likely to be inconsistent. Based on these general ideas, c_i and c_j are identified *inconsistent* with each other if (1) c_i is a part of c_j (but not equal to c_j), and c_j starts with a *negation* and ends with c_i; or (2) c_i appears before c_j (without overlap between them), where there is no extra negation applied

[2]http://sentiwordnet.isti.cnr.it/

[3]http://www.cs.pitt.edu/mpqa/

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

[5]http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector

[6]http://nlp.stanford.edu/software/lex-parser.shtml

to them, and they are connected by *contrasting conjunctions* (e.g., but, although, etc.) Here the "extra negation" means that the negation part of c_j or c_i. For example, in tweet "Alright enough of Taylor Swift. She is *gud* but I am still *not a fan*.", "*fan*" and "*not a fan*" are inconsistent according to (1), and "*gud*" and "*not a fan*" are inconsistent according to (2). "*gud*" and "*fan*" are not inconsistent since there is an extra negation "not" before "fan".

Identifying Consistency Relations: c_i and c_j are identified *consistent* with each other if c_i appears before c_j (without overlap between them), and there is no extra negation applied to them or no contrasting conjunction connecting them. For example, "*predictable*" and "*want my money back*' share a consistency relation in tweet "Saw the movie Friends With Benefits. So *predictable*! I *want my money back*."

In the above described manner, the algorithm identifies the consistency and inconsistency relations of the candidate expressions from the tweets in Δ. In each tweet, only the relations of two adjacent candidates (i.e., there is no other candidates in between) are considered in the algorithm, as it is difficult to tell how two expressions are related if they are distant from each other. Note that our work extends the existing methods (Hatzivassiloglou and McKeown 1997; Kanayama and Nasukawa 2006) of using conjunctions to assess the polarity relations of sentiment words. Since we consider n-grams instead of single words as candidates, the algorithm deals with not only negations and conjunctions, but also the position relations of two expressions, such as overlap and containment.

We construct two networks, in which the candidate expressions are connected by their consistency and inconsistency relations. Specifically, candidates are connected by their consistency relations in the *consistency network* $N^{cons}(P, R^{cons})$. P is a node set in which each node denotes one candidate, and R^{cons} is a set of weighted edges in which each edge denotes the consistency relation between two candidates. The weight of the edge is the frequency of that consistency relation in the whole tweet corpus. Similarly, candidates are connected by their inconsistency relations in the *inconsistency network* $N^{incons}(P, R^{incons})$.

These two networks encode the correlations of the target-dependent polarity of the candidate expressions over the entire tweet corpus. In the previous example, "*predictable*" and "*want my money back*" are consistent towards a movie target. It suggests that "*predictable*" should have the same polarity as "*want my money back*", i.e., both of them are negative. Given two networks with all the candidate expressions associated with movie target, the more "*predictable*" connects with negative expressions in the consistency network, or connects with positive expressions in the inconsistency network, the more likely it is negative with respect to movie.

An Optimization Model

We apply an optimization model to assess the target-dependent polarity of each candidate expression with the input of two relation networks. Instead of estimating the polarity directly, we assesses the *polarity probability* of each candidate, and the polarity can be determined accordingly.

Polarity probability is the measure of how likely an expression is positive or negative. Specifically, an expression c_i has two types of Polarity probability – ***P-Probability*** $Pr^P(c_i)$ is the probability that c_i indicates *positive* sentiment, and ***N-Probability*** $Pr^N(c_i)$ is the probability that c_i indicates *negative* sentiment. To model the intuition that the more likely c_i is positive (negative), the less likely it is negative (positive), we let $Pr^P(c_i) + Pr^N(c_i) = 1$.

The polarity of each expression can be determined according to their polarity probability. For example, an expression with P-Probability of 0.9, and N-Probability of 0.1 is highly positive. Another expression having its positive and negative probability as 0.45 and 0.55 does not have clear polarity and should be filtered out. Recall that we are only concerned with positive and negative expressions, and identifying neutral expressions is not in the scope of this paper.

Based on the P-Probability and N-Probability of each expression, we can obtain the probability that the sentiments of two expressions are consistent or inconsistent. We define the *consistency probability* of two expressions c_i and c_j as the probability that they carry the consistent sentiments, i.e., both c_i and c_j are positive (or negative). Assuming the polarity probability of c_i is independent of that of c_j, *consistency probability* becomes $Pr^P(c_i)Pr^P(c_j) + Pr^N(c_i)Pr^N(c_j)$. Similarly, their *inconsistency probability* is the probability that they carry the inconsistent sentiments, which is $Pr^P(c_i)Pr^N(c_j) + Pr^N(c_i)Pr^P(c_j)$.

A consistency relation between c_i and c_j in the network N^{cons} suggests that they indicate consistent sentiments in one tweet, i.e., the expectation of their consistency probability is 1. The difference between the consistency probability and its expectation can be measured by the squared error: $(1 - Pr^P(c_i)Pr^P(c_j) - Pr^N(c_i)Pr^N(c_j))^2$. Similarly, for an inconsistency relation in the network N^{incons}, the difference can be measured by $(1 - Pr^P(c_i)Pr^N(c_j) - Pr^N(c_i)Pr^P(c_j))^2$. The sum of the squared errors (SSE) for all the relations in two networks is:

$$SSE = \sum_{i=1}^{n-1}\sum_{j>i}^{n} (w_{ij}^{cons}(1 - Pr^P(c_i)Pr^P(c_j) - Pr^N(c_i)Pr^N(c_j))^2 + w_{ij}^{incons}(1 - Pr^P(c_i)Pr^N(c_j) - Pr^N(c_i)Pr^P(c_j))^2)$$

where w_{ij}^{cons} and w_{ij}^{incons} are the weights of the edges (i.e., the frequency of the relations) between c_i and c_j in N^{cons} and N^{incons}, respectively, and n is the total number of candidate expressions. Note that the squared error (instead of absolute error) is employed so that the two kinds of relations cannot cancel each other.

We want the P-Probabilities and N-Probabilities of the candidates to minimize the SSE, so that the corresponding consistency and inconsistency probabilities will be closest to their expectations suggested by the networks. By replacing $Pr^N(c_i)$ with $1 - Pr^P(c_i)$, and abbreviating $Pr^P(c_i)$ and $Pr^P(c_j)$ to x_i and x_j, we get the objective function:

$$minimize\{\sum_{i=1}^{n-1}\sum_{j>i}^{n}(w_{ij}^{cons}(x_i + x_j - 2x_ix_j)^2 +$$

$$w_{ij}^{incons}(1 - x_i - x_j + 2x_i x_j)^2)\}$$

subject to,

$$0 \leq x_i \leq 1, \qquad for \quad i = 1, \cdots, n$$

If a candidate c_i is contained in the seed word set S^0, we simply let its P-Probability x_i be 1 (or 0) if c_i is positive (or negative) according to S^0. The reason is that S^0 is created to contain the words that are always positive or negative regardless of the targets. The P-Probabilities of other candidates will be obtained by solving this optimization problem.

We choose to use the L-BFGS-B[7] algorithm to solve this constrained optimization problem with simple bounds. This algorithm is based on the gradient projection method to determine a set of active constraints at each iteration, and uses a limited memory BFGS matrix to approximate the Hessian of the objective function. Byrd et al. (1995) show that the L-BFGS-B takes advantage of the form of the limited memory approximation to implement the algorithm efficiently. The initial guess for the parameters (the P-Probabilities of candidate expressions) are needed as the input of the L-BFGS-B algorithm. We implement and compare two methods to initialize the P-Probabilities in the experiments.

As a result, we get the P-Probability and N-Probability of each candidate. Only candidates with P-Probability or N-Probability higher than threshold τ are identified as positive or negative expressions. Other expressions falling below the threshold are removed from the result. However, there might be some irrelevant candidates with high polarity probability that are not filtered out. The main reason is that the assessment of the polarity probability of some expressions is based on very sparse data. A candidate, which only appears very few times in the corpus and happens to be consistent with positive expressions, could be assigned a high P-Probability. To deal with this problem, we use another score to measure the confidence of the polarity assessment. For each candidate c_i, the score is calculated as: $\varepsilon(c_i) = \frac{max(Pr^P(c_i), Pr^N(c_i)) * df(c_i)}{n_{words}(c_i)}$, where $df(c_i)$ is the number of tweets containing c_i, and $n_{words}(c_i)$ is the number of words it contains. Note that ε is biased towards shorter phrases. The reason is that the shorter phrases tend to have more relations in the relation networks, therefore their polarity assessments are more reliable compared with these of longer phrases. c_i is removed from the result if $\varepsilon(c_i)$ is less than threshold σ. Empirically, we set $\tau = 0.6$ and $\sigma = 0.6$ in the experiments.

Experiments

First, we describe the experimental setup. Then we examine the quality of the sentiment expressions extracted by our method in comparison with several baseline methods, and investigate the performance of our approach and other baselines with various sizes of corpora. In addition, to show the usefulness of the extracted sentiment expressions in applications, we apply them to the task of sentiment classification of tweets.

[7] http://www.mini.pw.edu.pl/∼mkobos/programs/lbfgsb_wrapper/index.html

N-gram	1	2	3	4	5	>5
Movie Dom.(%)	54.24	21.02	10.17	6.44	4.74	3.39
Person Dom.(%)	71.38	17.75	7.25	1.81	1.45	0.36

Part-of-speech	Adj.	Verb	Noun	Oth.
Movie Dom.(%)	57.63	26.10	13.22	3.05
Person Dom.(%)	45.29	31.52	21.02	2.17

Table 1: Distributions of N-grams and Part-of-speech of the Sentiment Expressions in the Gold Standard Data Set

Sentiment Category	Pos.	Neg.	Neut.	Obj.
Movie Dom.(%)	28	6.4	2.4	63.2
Person Dom.(%)	19.47	7	0.47	73.06

Table 2: Distribution of Sentiment Categories of the Tweets in the Gold Standard Data Set

Experimental Setup

We use two collections of tweets: one contains 168,005 tweets about movies, and the other contains 258,655 tweets about persons. Each tweet of the two collections contains either a movie or a person as the target. The data has been made publicly available[8].

Gold Standard: We created gold standard using 1,500 tweets randomly sampled from the corpus for each domain (totally 3,000 tweets for both domains). The 3,000 tweets were given to two groups of human annotators, and each group consisted of three annotators. One group of annotators recognized the sentiment expressions for the given target from each tweet. The other group of annotators classified each tweet as *positive*, *negative*, *neutral* or *objective* according to the overall sentiment towards the target.

We selected the sentiment expressions which were agreed on by at least two annotators, and finally got 295 and 276 expressions in movie and person domains, respectively. Table 1 shows the distribution of these expressions. We can see that both the lengths and the part-of-speech of the sentiment expressions exhibit diversity. We also got the 1,500 tweets for each domain labeled with their sentiment categories. Table 2 illustrates the distribution of the tweets belonging to different sentiment categories.

Baseline Methods: The following baselines were chosen for comparison, in which MPQA, General Inquirer and SentiWordNet are benchmark polarity lexicons which are often used to evaluate the extraction algorithms, and PROP is a propagation approach proposed by Qiu et al. (2009). Prior methods that support phrase extraction either lack consideration of the sentiment target or need extra effort to develop syntactic patterns, thus we do not employ them here.

MPQA, GI, SWN: For each extracted root word regarding the target, simply look up its polarity in MPQA, General Inquirer and SentiWordNet, respectively.

PROP: Starting with some seed sentiment words (here we apply the seed word set S^0), this method extracts new sentiment words and sentiment targets through a double propagation process over the corpus. It uses a set of extraction rules based on different relations between sentiment words and targets, and also sentiment words and targets themselves. In our setting, sentiment targets have been specified, so we

[8] http://knoesis.org/projects/sentiment

Method	Precision	Recall	F-measure
Movie Domain			
MPQA	0.3542	0.5136	0.4193
GI	0.3318	0.4320	0.3753
SWN	0.2876	0.4898	0.3624
PROP	0.4742	0.5034	0.4884
COM-const	**0.6433**	0.5170	**0.5733**
COM-gelex	0.5164	**0.5578**	0.5363
Person Domain			
MPQA	0.3523	0.4746	0.4045
GI	0.2949	0.4058	0.3416
SWN	0.2161	0.3659	0.2718
PROP	0.5352	0.3696	0.4372
COM-const	**0.5879**	0.4710	**0.5230**
COM-gelex	0.4599	**0.5507**	0.5012

Table 3: Quality of the Extracted Sentiment Expressions from the 1,500 Tweets in the Gold Standard Data Set

adapt the method to extract only sentiment words. The original method only concerns adjectives, and we extend it to extract adjectives, verbs, nouns and adverbs by relaxing the constraints of extraction rules.

Our method is represented as "COM" (Constrained Optimization Model). For candidates contained in the seed word set S^0, we have discussed their P-Probabilities in the section of *An Optimization Model*. For other candidates, we initialize their P-Probabilities in two different ways:

COM-const: Assign 0.5 to all the candidates as their initial P-Probabilities.

COM-gelex: We leverage the prior polarities of words in the root word set Γ to initialize the candidate polarities. Specifically, assign 1 (or 0) to a candidate as its initial P-Probability, if the candidate contains some positive (or negative) words but no negative (or positive) words according to Γ; otherwise, assign 0.5 to the candidate.

Evaluation Measurement: The quality of the extracted expressions is measured by precision, recall and F-measure. We define $precision = \frac{N_{agree}}{N_{result}}$, $recall = \frac{N_{cover}}{N_{gold}}$, and $F - measure = \frac{2 \times precision \times recall}{precision + recall}$, where N_{agree} is the number of extracted expressions that are agreed with the gold standard, N_{result} is the number of extracted expressions, N_{cover} is the number of expressions in the gold standard that are agreed with the extraction result, and N_{gold} is the number of expressions in gold standard. We use the *contain rule* to decide whether an expression is agreed with another expression. Thus, positive expression "good" is agreed with positive expression "pretty good" or vice versa. We also deal with the negation, thus, positive expression "good" is agreed with negative expression "not good" or vice versa.

Quality of the Extracted Sentiment Expressions

Table 3 shows the precision, recall and F-measure on evaluating the sentiment expressions extracted from 1,500 tweets of gold standard for both domains. We can see that both versions of our method outperform the baseline methods. Specifically, our best F-measure in movie domain is 8.49%-21.09% higher than that of baselines, and in person domain, our best F-measure is 8.58%-25.12% higher than that of baselines. In both domains, the highest precision is achieved by COM-const, and the highest recall is achieved by COM-gelex. Among all of the three lexicon-based methods (MPQA, GI and SWN), MPQA provides the best result, however, its precision is relatively low. The PROP method performs quite well in terms of precision, but it suffers from low recall, especially in person domain.

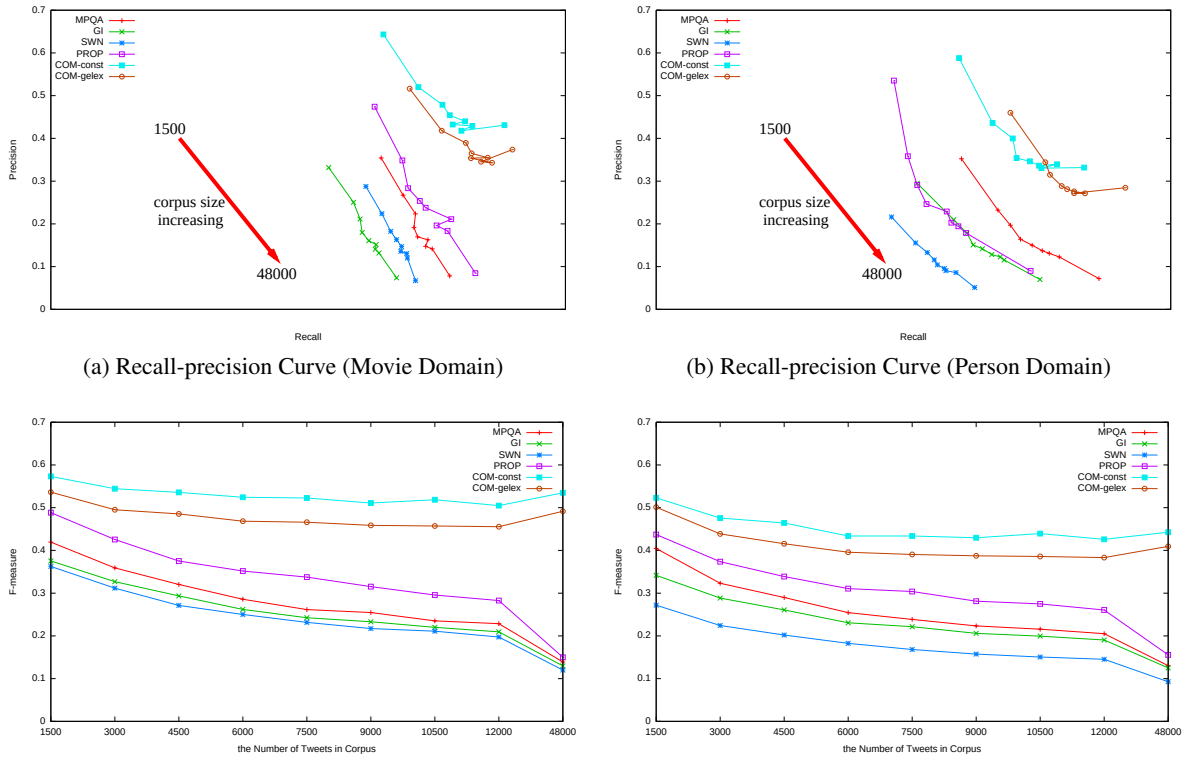
Compared with the lexicon-based methods, our method gets significantly higher precision. The main reason is that the polarity of expressions is sensitive to the target, which can be captured by our method. The lexicon-based methods do not use the information of the corpus so that they cannot handle the target-dependent polarity of expressions. Compared with the PROP method, which estimates the polarity of expressions in a rule-based manner, our method also shows great F-measure gains in both domains. It demonstrates the advantage of our optimization-based approach over the rule-based manner in polarity assessment.

We also conduct experiments to investigate the effect of corpus size on the quality of extraction results. We expect to get higher quality results as more inter-expression relations are learned from larger corpora. We evaluate all approaches over the corpora of sizes from 1,500 to 48,000. Since it is not practical to manually label such a large amount of tweets, we compare results extracted from corpora of different sizes against the same 1,500 tweets of the gold standard data set. To make the comparison meaningful, we make sure that all the corpora of different sizes are randomly selected from the tweet collections, and each of them includes the 1,500 tweets of the gold standard data set.

Figure 1 shows how precision, recall and F-measure change as we increase the sizes of corpora. Note that the precision may be worse than the true quality obtainable using a larger corpus, since the gold standards are generated from a subset of tweets. To gain more insights into the results, we show both precision-recall curve and F-measure to examine the relative performance of different methods.

Figures 1a and 1b show how our method outperforms the baselines. Specifically, COM-const tends to get the highest precision and COM-gelex tends to get the highest recall. Among all the baselines, PROP works best for the movie data, especially on the recall aspect, while MPQA provides the best results for person domain. However, all baseline methods suffer from a sharp decline of precision with the increasing recall. By manually checking the extraction results of the baseline methods, we find many irrelevant words that do not indicate sentiment with respect to the target. Our approach can effectively filter these noises because it assesses target-dependent polarities based on the relation networks generated over the whole corpus. Note that both versions of our approach make increases on both precision and recall when we increase the size of corpora from 12,000 (the second right most point of each line) to 48,000 (the right most point of each line). It suggests that our method could benefit from more relations extracted from larger corpora.

Observing from Figures 1c and 1d, in both domains, our method makes significant improvement on F-measure over four baselines, and COM-const provides the best results. F-measures of our approach decline a little as we increase the corpus size ($\leq$ 6,000). Then they maintain at the same level



(a) Recall-precision Curve (Movie Domain)

(b) Recall-precision Curve (Person Domain)

(c) Average F-measure (Movie Domain)

(d) Average F-measure (Person Domain)

Figure 1: Results of Sentiment Expression Extraction with Various Corpora Sizes

or decrease very slightly until the size of corpus reaches 12,000. From 12,000 to 48,000, the F-measures even go up a little. Table 4 illustrates a small sample of extracted sentiment expressions by our method (with a corpus of 48,000 tweets in movie domain). Most of these expressions are not identified by the baselines. For both positive and negative categories, we present expressions with target-dependent polarities (e.g., "bomb", "predictable"), multi-word phrases (e.g., "must see" "thumbs down"), and slang (e.g., "luv", "stoopid"). These concrete examples show that our method captures the diversity of expressions and assesses the target-dependent polarities in an intuitively satisfactory manner.

Sentiment Classification of Tweets

We apply the sentiment expressions extracted by different methods for the task of classifying tweets as *positive*, *negative*, *neutral* or *objective*. The 1,500 tweets of gold standard data set for each domain are used for testing. Accordingly, the sentiment expressions extracted from the testing set are used for the classification task. Specifically, for each tweet, we identify the sentiment expressions based on the extraction results, and remove the ones that are not regarding the specified targets or are parts of other expressions (e.g., "good" is removed if "not good" appears in the same tweet). Then we assign a score to each remaining expression (i.e., 1 for positive and -1 for negative expression), and get the sum of scores to determine the sentiment category of the tweet (i.e., *positive* with the sum > 0, *negative* with the sum < 0, and *neutral* otherwise). If no sentiment expression is identified from the tweet, it is labeled as *objective*.

We use precision, recall and F-measure to measure the result of sentiment classification, and count only three sentiment categories, i.e., *positive*, *negative* and *neutral*. The results on both domains are shown in Table 5. We can see that the performance of different methods is quite consistent with the quality of the extraction results they obtain. Our method achieves the best F-measure on both domains. Currently, this method has been deployed in Twitris system [9] to analyze the entity specific sentiment.

Conclusion

In this paper, we present an optimization-based approach for extracting diverse sentiment expressions for a given target from a corpus of unlabeled tweets. To the best of our knowledge, extracting sentiment expressions associated with given targets has not been studied on tweets. Previous approaches on formal text are usually limited to extracting words/phrases belonging to certain patterns, and are not capable of capturing the diversity of expressions in tweets. Our approach exploits multiple lexical resources to collect general sentiment-bearing words as root words, which cover both formal and slang words. It then extracts n-grams containing on-target root words as candidates. To assess the target-dependent polarity, inter-expression relations are ex-

[9] http://twitris.knoesis.org

Positive			Negative		
Target-dependent	**Multi-word Expressions**	**Slang**	**Target-dependent**	**Multi-word Expressions**	**Slang**
bomb	must see	aight	average	thumbs down	stoopid
intense	eye candy	tight	sleepy	screwed up	craptastic
kick ass	rate 5 stars	rad	predictable	nothing special	rediculous
light-hearted	funny as hell	luv	copying	pretty lame	wacky
pretty crazy	pretty damn funny	awsome	cheapest	not even funny	superbad
cried alot	better than i expected	kool	little slow	sucked big time	dense
rules box office	even better the second time	rockin	little long	don't waste your money	crapest

Table 4: Diverse Forms of Expressions Extracted by the Proposed Method

Method	Precision	Recall	F-measure
Movie Domain			
MPQA	0.6566	0.5507	0.5990
GI	0.6381	0.4982	0.5595
SWN	0.5266	0.5018	0.5139
PROP	0.7677	0.5507	0.6413
COM-const	**0.8015**	0.5851	**0.6764**
COM-gelex	0.7164	**0.5905**	0.6474
Person Domain			
MPQA	0.5250	0.3639	0.4299
GI	0.4419	0.3292	0.3773
SWN	0.2979	0.3119	0.3047
PROP	0.5371	0.3045	0.3887
COM-const	**0.6351**	0.3317	0.4358
COM-gelex	0.5925	**0.3886**	**0.4694**

Table 5: Performance of Tweet Sentiment Classification Using the Extracted Sentiment Expressions

tracted from the corpus and incorporated into an optimization model to estimate the polarity probability of each candidate. Using tweets from two domains we demonstrate that our approach is able to extract diverse sentiment expressions, and predict their target-dependent polarities. We also show how this approach greatly improves the performance compared with several baseline methods, in terms of both quality and scalability with respect to the size of corpora.

Acknowledgements

This work is supported by NSF (IIS-1111182, 09/01/2011-08/31/2014) SoCS program.

References

Agarwal, A.; Biadsy, F.; and Mckeown, K. R. 2009. Contextual phrase-level polarity analysis using lexical affect scoring and syntactic N-grams. *In Proc. of EACL.*

Barbosa, L., and Feng, J. 2010. Robust sentiment detection on Twitter from biased and noisy data. *In Proc. of COLING*.

Bermingham, A., and Smeaton, A. 2010. Classifying sentiment in microblogs: is brevity an advantage?. *In Proc. of CIKM*.

Bertsekas, D. 1999. Nonlinear Programming, 2nd Edition. *Athena Scientific*.

Byrd, R. H.; Lu, P.; and Nocedal, J. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*.

Choi, Y.; Kim, Y; and Myaeng, S. 2009. Domain-specific sentiment analysis using contextual feature generation. *In Proc. of TSA*.

Davidov, D.; Tsur, O.; and Rappoport, A. 2010. Enhanced sentiment learning using Twitter hashtags and smileys. *In Proc. of COLING*.

Gruhl, D.; Nagarajan, M.; Pieper, J.; Robson, C.; and Sheth, A. 2010. Multimodal social intelligence in a real-time dashboard system. *The VLDB Journal*, v.19 n.6.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. *In Proc. of EACL.*

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. *In Proc. of SIGKDD*.

Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; and Zhao, T. 2011. Target-dependent Twitter sentiment classi?cation. *In Proc. of ACL.*

Kanayama, H., and Nasukawa, T. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. *In Proc. of EMNLP*.

Kessler, J. S., and Nicolov, N. 2009. Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations. *In Proc. of ICWSM*.

Kouloumpis, E.; Wilson, T.; and Moore, J. 2010. Twitter sentiment analysis: the good the bad and the OMG! *In Proc. of ICWSM*.

Lu, Y.; Castellanos, M.; Dayal, U.; and Zhai, C. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. *In Proc. of WWW*.

Nigam, K., and Hurst, M. 2006. Towards a robust metric of polarity. *J. Shanahan, Y. Qu and J. Wiebe. Computing Attitude and Affect in Text: Theory and Applications.* Springer.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*.

Peng, W., and Park, D. H. 2011. Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization. *In Proc. of ICWSM*.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2009. Expanding domain sentiment lexicon through double propagation. *In Proc. of IJCAI*.

Turney, P. D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *In Proc. of ACL.*

Velikovich, L.; Blair-Goldensohn, S.; Hannan, K.; and McDonald, R. 2010. The viability of web-derived polarity lexicons. *In Proc. of HLT/NAACL.*

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. *In Proc. of HLT/EMNLP*.

Somasundaran, S.; Namata, G.; Wiebe, J.; and Getoor, L. 2009. Supervised and Unsupervised Methods in Employing Discourse Relations for Improving Opinion Polarity Classification. *In Proc. of EMNLP*.

Yi, J.; Nasukawa, T.; Bunescu, R.; and Niblack, W. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. *In Proc. of ICDM*.

Zhang, L.; Ghosh, R.; Dekhil, M.; Hsu, M.; and Liu, B. 2011. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. *Hewlett-Packard Labs Technical Report*, HPL-2011-89.