MuSES: Multilingual Sentiment Elicitation System for Social Media Data

Yusheng Xie, Zhengzhang Chen, Kunpeng Zhang, Yu Cheng, Daniel K. Honbo, Ankit Agrawal, and Alok N. Choudhary, *Northwestern University*

A multilingual sentiment identification system (MuSES) implements three different sentiment identification algorithms. In addition, a proposed label-free process transfers multilingual sentiment knowledge between different languages.

The rapid growth in volume of user-generated Web texts from social network sites such as Facebook and Twitter drives us to analyze unstructured textual data through computational techniques with minimal manual intervention. Identifying their sentiments becomes an important challenge. Due to economic globalization, most social media channels, including Facebook and Twitter, provide global services (see www.insidefacebook.com/2010/05/24). Diversified demographics entail diversity in the languages spoken on Facebook. As of May 2010, just 52 percent of all active Facebook users access Facebook in English. In other words, social media channels, most of which are just as global as Facebook, are essentially uncategorized, mixed sources of text containing multiple languages. As a result, the challenge to understand the massive amount of social opinions is two-fold: infer the language in which a sentence or a paragraph is written; and understand its sentiment given its language inference.

With this in mind, we developed MuSES, a multilingual sentiment identification system. Our technical contributions include

- a novel zero-effort labeling system that leverages knowledge bases like Wikipedia, and labels word-level sentiment for non-English words;
- an improved compositional semantic rule algorithm that considers unique semantics in social media text;
- a scoring-based sentiment algorithm that assigns numeric scores to phrasal patterns at finer granularity than any previous efforts; and
- a novel, rule-based algorithm that's made especially effective in social media context by considering emoticons and domain knowledge.

In our experiments, we demonstrate (through meta learning models) that combining the outcomes from individual algorithms can achieve significantly higher accuracy. For others' work, see the related sidebar.

Data Preprocessing and Text Cleaning

We use some existing tools to preprocess and clean incoming social media text, a process that can help achieve better sentiment classification.

1541-1672/14/$31.00 © 2014 IEEE
Published by the IEEE Computer Society

Related Work in Sentiment Analysis

Sentiment analysis is well studied for English content. Researchers have developed a lot of rule-based, machine learning techniques. Techniques in non-English sentiment analysis are relatively underdeveloped, but are catching up rapidly.

Monolingual Sentiment Processing

A wide range of sentiment research has been done on machine learning techniques. In the work of Bo Pang and his colleagues[1] as well as Xia Hu and his colleagues,[2] Naive Bayes, maximum entropy classification, and support vector machine techniques are explored to classify overall document sentiments. However, in their experiments, such machine learning methods didn't perform as well for sentiment classification as for traditional topic-based categorization. We think treating sentiment analysis purely as a statistical classification problem might not be a viable approach after all. In the work of Ramanathan Narayanan and his colleagues, the authors present a linguistic analysis of conditional sentences, and build some supervised learning models to determine if sentiments expressed on different topics in a conditional sentence are positive, negative, or neutral.[3] Several researchers have also studied feature-based sentiment analysis.[4] Their objective is to extract topics or product features in sentences and determine whether the sentiments expressed in them are positive or negative. Jingjing Liu and Stephanie Seneff propose an approach to extract adverb-adjective-noun phrases based on clause structure obtained by parsing sentences into a hierarchical representation.[5] Some of our previous work also proposes similar techniques.[6]

Language Detection

Thanks to the global popularity of various social media websites, multilingual sentiment identification is drawing the attention of many research efforts. Machine translation is one way to handle multilingual sentiment when a system is an expert in sentiment identification for one language.[7] Approaches based on machine translation in general suffer from its strong dependency on the quality of the translation, optimizing this method would require a large amount of training data and the subsequent computational cost of the training. On the other hand, dictionary-based methods are more computationally efficient, but would still require a large amount of labeling.[8] In the work of Jianxin Yao and his colleagues, they describe a lexicon-based approach to determine the sentiment for each Chinese word by looking it up in multiple Chinese-to-English dictionaries and claim accuracy of around 90 percent.[8]

However, a few aspects prevent the results in Yao's work from being practical. First, contextual information is ignored in such approaches; to say the least, contextual information isn't considered until the Chinese text is translated into English. It's better to handle contextual information natively before translation, because coarse word-level translations often destroy some important contextual links. Second, manual labeling is essential in their model. Admittedly, researchers can assume the existence of labeled sentiment words in English. But in the model used by Yao, further labeling is required on the feature-represented Chinese words. Given today's infrastructure on the Internet such as Wikipedia, labeling efforts should be minimized by leveraging existing labels or weak labels.

References

1. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up? Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. Empirical Methods in Natural Language Processing* (EMNLP), 2002, pp. 79–86.
2. X. Hu et al., "Exploiting Social Relations for Sentiment Analysis in Microblogging," *Proc. 6th ACM Int'l Conf. Web Search and Data Mining* (WSDM), 2013, pp. 537–546.
3. R. Narayanan, B. Liu, and A. Choudhary, "Sentiment Analysis of Conditional Sentences," *Proc. Conf. Empirical Methods in Natural Language Processing* (EMNLP), 2009, pp. 180–189.
4. Q. Mei et al., "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs," *Proc. 16th Int'l Conf. World Wide Web*, 2007, pp. 171–180.
5. J. Liu and S. Seneff, "Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm," *Proc. Conf. Empirical Methods in Natural Language Processing* (EMNLP), 2009, pp. 161–169.
6. K. Zhang et al., "SES: Sentiment Elicitation System for Social Media Data," *Proc. IEEE Int'l Conf. Data Mining* (ICDM), 2011, pp. 129–136.
7. M. Bautin, L. Vijayarenu, S. Skiena, "International Sentiment Analysis for News and Blogs," *Proc. 3rd Int'l AAAI Conf. Weblogs and Social Media*, 2008, pp. 19–26.
8. J. Yao et al., "Using Bilingual Lexicon to Judge Sentiment Orientation of Chinese Words," *Proc. IEEE 6th Int'l Conf. Computer and Information Technology* (CIT), 2006; doi:10.1109/CIT.2006.190.

It's not uncommon to see multiple positive and negative opinions in a single comment. For example, "The new Amazon Kindle has a really nice screen, but its Wi-Fi radio is awful!" To simplify this problem, we split a comment into sentences. Each sentence will be assigned as positive, neutral, or negative. We use MaxTerminator to split messages into sentences.[1] We also remove hyperlinks and correct misspellings when dealing with social and Web text. We've manually collected 138 pairs of misspelled and corrected words in English, 129 in German, 134 in Korean, and 178 in Chinese. In addition, most sentiment-bearing words are adjectives, adverbs, verbs, and negation words. Certain important rules defined in our algorithms rely on part-of-speech (POS) information in a sentence. In MuSES, we employ the Stanford POS tagger to label each POS for every English sentence as well as Chinese and German.[2] The Stanford tagger doesn't handle Korean, so we apply the open source HanNanum project (see http://hannanum.sourceforge.net) for Korean POS tagging.

Handling Multilingualism

Handling multilingualism involves two steps: language detection and sentiment

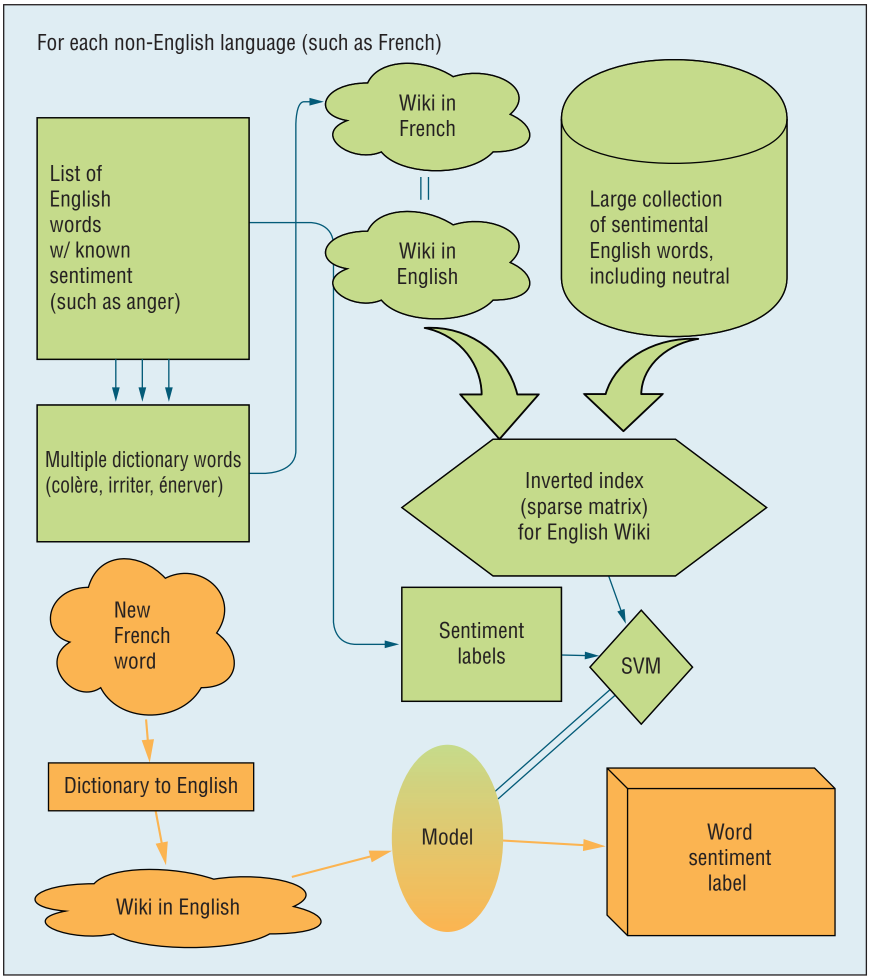

Figure 1. Zero-effort multilingual label system (MLLS). The training process is marked in blue and the predicting process is marked in red. The non-English language we're dealing with is French.

translation. While our system handles language detection differently for different languages, we propose a unified sentiment translation process.

Language Detection

We perform European language detection through an existing package called LingPipe (see http://alias-i.com/lingpipe). In practice, the detection performance is close to perfect. This is because our multilingual dataset is only from Amazon reviews, which have strong contextual information and sufficient length in most cases. On the other hand, detection on shorter texts like Facebook comments or tweets would be more challenging due to insufficient contextual information, which can be compensated for by the strong social information from such sources. For example, we can determine a user's language preference by learning from his/her activities on the entire social network. It's easy to detect most popular Asian languages because we can simply look at the Unicode spectrum.

Lexicon-Based Sentiment Translation

Our lexicon-based approach distinguishes itself from most previous efforts by avoiding intensive manual labeling for international languages. For a given non-English language, there are two types of labeling required: word- and sentence-level.

The word-level sentiment label is the foundation for most sentiment algorithms to infer sentence-level sentiment. Such labels are widely available for English and only a few other popular languages. However, even based on predictive models,[3] obtaining these labels for a new language as a training set still requires a large amount of manual work. We propose a multilingual label system (MLLS) that's lexicon-based and is label-free.

Figure 1 is a two-tone diagram that illustrates MLLS, where the training process is marked in blue and the predicting process is marked in red. To illustrate the flow of MLLS, suppose the non-English language we're dealing with is French.

The training process is described in Algorithm 1 (see Figure 2). A key step of training MLLS is to build the inverted index on BOW_E based on $\bar{\varepsilon}$. In other words, MLLS maps each bag-of-words in BOW_E to vectors in a $|\bar{\varepsilon}|$-dimensional binary feature space. The set of these feature vectors are denoted by FTR_E. Each vector in FTR_E, with the sentiment label of its corresponding element from the pre-labeled set ε can be fed into a support vector machine (SVM) classifier. To conclude the training process of MLLS, the classifier outputs a model file. The only labels used in the training process are the pre-collected ones for English sentiment words, which we assume are widely available. The training process only happens once for each non-English language.

The predicting process in MLLS is relatively simple. For a given French word, MLLS retrieves its English translations by looking up French-to-English dictionaries; then the system queries Wikipedia to download the corresponding English Wikipedia documents; next, the feature vectors for these documents are sent to the model file for classification on sentiment. The output is what we call a *sentiment label* on the input French word. In our formulation, we assume sentiment labels are nominal-valued, but this condition isn't necessary. In addition, the reason why we chose Wikipedia pages—despite the possible noise they might introduce—over other bilingual dictionaries is mainly because dictionary entries are always lacking in detail and will result in sparse vectors in the feature space.

Sentence-level sentiment for non-English sentences is derivable based on word-level sentiment. Given the sentences' sentiment representation, our algorithms in the next section for deriving sentence-level sentiment from word-level sentiment are independent of the underlying language.

Three Sentiment Algorithms

Here, we'll describe three algorithms for identifying the sentiment of a single sentence. The first one is based on compositional semantic rules. A few new rules are devised and added to the basic compositional rules proposed by Yejin Choi and Claire Cardie[4] to accommodate social media context. In addition to rules, we also propose a more sophisticated compose function for processing the larger number of compositional rules. The output is one of five integers ranging from –2 to +2 (+2 means strongly positive, –2 means strongly negative, and 0 means neutral). Our second algorithm distinguishes sentiment degrees and reflects them by numeric values between –5 and 5. The third algorithm identifies sentiments by checking some rules defined on emoticons, contextual negation words, and domain-specific phrases. With the help of MLLS, the three algorithms can work with different languages, although their previous versions only deal with English.

Input: ε, pre-labeled English words with clear positive or negative sentiment (for example, angry, happy, and so on); $\overline{\varepsilon}$, a superset of ε including more words and phrases that express any degree of emotion.

Output: M, a classification model that predicts word sentiment.

```
F ← empty list of French words
WIKI_F ← empty list of French Wikipedia links
WIKI_E ← empty list of English Wikipedia links
BOW_E ← empty list of bag-of-words representations of
items in WIKI_E
FTR_E ← empty list of English Wikipedia links
for each e ∈ ε do
    f ← Find e in English-French dictionaries
    append f to F
end
for each f ∈ F do
    w_F ← Find f in Wikipedia French
    append w_F to WIKI_F
end
for each w_F ∈ WIKI_F do
    w_E ← Find w_F's link in Wikipedia English
    append w_E to WIKI_E
end
for each w_E ∈ WIKI_E do
    b_E ← page contents from w_E's link in Wikipedia
    English
    append b_E to BOW_E
end
for each b_E ∈ BOW_E do
    for each w ∈ b_E do
        remove w if w ∉ ε̄
    end
    append b_E to FTR_E
end
for each f_E ∈ FTR_E do
    Train SVM model M with f_E and its label in ε
end
return M
```

Figure 2. Algorithm 1. Zero-effort Multilingual Label System (MLLS) Training Algorithm (using French as an example choice of multilingualism).

Compositional Semantic Rule Algorithm

Table 1 shows the compositional rules and corresponding examples. We propose the first seven rules as general not specific for social media and can wrongly parse the sentences in a social domain. For example, "thumbs up" is ambiguous because this can technically be a phrase omitting the verb "[put] thumbs up" or using "thumbs" as a verb. Since English grammar in social media texts and POS tagging isn't perfect, we design some new rules (rules 8 and 13, as shown in Table 1) to catch some of the errors.

For other languages, we try to translate all rules to each language. But not all rules can be translated. For German, we can translate rules 1 and 12 with virtually no change; for Chinese, some rules collapse into one rule and we can preserve 1, 2, 3, 5, 8, 9, 11, and 12; for Korean, we keep 1, 2, 6, 9, 11, and 12.

Table 1. The 13 compositional semantic rules.

Rule number	Semantic rules*	Examples
1	Polarity(not arg1) = –Polarity(arg1)	Not bad.
2	Polarity(VP1 NP1) = Compose(VP1, NP1)	Destroyed terrorism.
3	Polarity(VP1 to VP2) = Compose(VP1, VP2)	Refused to deceive the man.
4	Polarity(ADJ to VP1) = Compose(ADJ, VP1)	Unlikely to destroy the planet.
5	Polarity(NP1 of NP2) = Compose(NP1, NP2)	Lack of crime in rural areas.
6	Polarity(NP1 VP1) = Compose(NP1, VP1)	Crime has decreased.
7	Polarity(NP1 be ADJ) = Compose(ADJ, NP1)	Damage is minimal.
8	Polarity(NP1 of VP1) = Compose(NP1, VP1)	Lack of killing in rural areas.
9	Polarity(as ADJ as NP) = $1_{\{Polarity(NP)=0\}} \cdot$ Polarity(ADJ) + $1_{\{Polarity(NP)\neq 0\}} \cdot$ Polarity(NP)	As ugly as a rock.
10	Polarity(not as ADJ as NP) = –Polarity(ADJ)	That wasn't as bad as the original.
11	If sentence contains "but," disregard all previous sentiment and only take the sentiment of the part after "but."	And I've never liked that director, but I loved this movie.
12	If sentence contains "despite," only take the sentiment of the part before "despite."	I love that movie, despite the fact that I hate that director.
13	If sentence contains "unless" and "unless" is followed by a negative clause, disregard the "unless" clause.	Everyone likes this video unless he is a sociopath.

The blue/red color indicates which word is which part. For example, for rule number 3, we use the color to indicate that "refused" is VP1 and "deceive" is VP2. VP = verb phrase, NP = noun phrase, and ADJ = adjective.

Table 2. Compose functions used to detect polarity of an expression.*

Compose functions	Algorithms
Compose1 (arg1, arg2)	1. Return –Polarity(arg2) if arg1 is negation. 2. Return Polarity(arg1) if Polarity(arg1) = Polarity(arg2). 3. Otherwise, return the majority term polarity in arg1 and arg2.
Compose2 (arg1, arg2)	1. Return Polarity(arg2) if arg1 is negative and arg2 is not neutral. 2. Return –1 if arg1 is negative and arg2 is neutral. 3. Return Polarity(arg2) if arg1 is positive and arg2 is not neutral. 4. Return 2*Polarity(arg1) if Polarity(arg1) = Polarity(arg2). 5. Return Polarity(arg1) + Polarity(arg2) if arg1 is positive and arg2 is neutral. 6. Return Polarity(arg1) + Polarity(arg2) if arg2 is positive and arg1 is neutral. 7. Otherwise, return 0.

* Compose1, used in others' work,[4] only produces –1, 0, or 1; we propose Compose2, which produces integers between –2 and 2.

In addition to these rules, we still need to have a composition function that effectively incorporates the rules into the decision making of the sentence sentiment. A key component of Compose2 is that in addition to sentiment polarity, it also produces sentiment strength. As Table 2 shows, the previous Compose1[4] doesn't assign sentiment strength.

The goal of this algorithm isn't to directly compete with others' work.[5] We don't intend for this algorithm to provide an exhaustive list of applicable rules for sentiment identification. We built this algorithm as a simple and efficient component of the larger MuSES system.

Numeric Sentiment Identification Algorithm

The numeric sentiment identification algorithm solves two major problems: how to associate numeric scores with the degree of textual sentiment; and how to combine all the scores of multiple words for a sentence if we assign a score to each word/phrase.

Scores for words. Our approach hypothesizes that two kinds of phrases can associate with numeric scores: adverb-adjective-noun (AAN) and verb-adverb (VA) phrase. For example, "a very good question" is an AAN type and "do not like it very much" is a VA type. We only consider the words that form an AAN/VA pattern in this algorithm. Our method is similar in spirit to Jingjing Liu and Stephanie Seneff's approach.[6] Liu and Seneff assign sentiment scores to words based on their appearances in star rating reviews. By collecting a large volume of reviews with star ratings, we can effectively associate words, which appear in the reviews, with ratings. The ratings will then be used as sentiment labels for those words. However, we argue that the sentiment scores aren't only associated with user star ratings, but also with word appearance frequency. By associating user star ratings and frequency with each phrase extracted from review texts, we can easily associate numeric scores with textual sentiment. For both "adjective" and "adverb adjective" words, we define their scores and polarity values by averaging their star ratings to get their scores. The details are given in Equation 1:

$$\begin{aligned}\text{Score}(w) &\equiv \sum_{i\in P} n(r_i)\cdot r_i \cdot f_i \sum_{j\in N} n(r_j), \text{and}\\ \text{Pol}(w) &\equiv \text{sgn}\big(\text{Score}(w)\big). \qquad (1)\end{aligned}$$

In Equation 1, P represents the set of reviews that contain word w, r_i represents the associated star rating in the ith review that contain w, N represents total number of reviews used in the entire dataset, $n(r_i)$ represents the number of reviews with star rating r_i, and f_i is the number of times w appears in the ith review. The score is averaged over all appearances, weighted by the frequency count for removing bias toward any words.

To assign a score to each adverb, we find all entries containing this adverb from the training dataset such that the adverb is followed by an adjective. Then, for each adjective in the list, we define the score for an adverb in Equation 2 to be the difference between the score of the adverb-adjective phrase and the score of the adjective:

$$\text{Score(adv)} \equiv \text{Pol(adj)} \cdot (\text{Score("adv adj")} - \text{Score(adj)}). \qquad (2)$$

Scores for phrases and sentences. After obtaining the scores for adverbs and adjectives, the next step is to assign the strength of sentiment to each AAN or VA phrase. The scores for AAN or VA phrase, defined in Equations 3 and 4, are based on compositions of the scores and polarities of individual adverbs/adjectives:

$$\text{Score("adv adj noun")} \equiv \text{Score(adj)} + \text{Pol(adj)} \cdot \text{Score(adv)), and} \qquad (3)$$

$$\text{Score("adv verb")} \equiv \text{Pol(verb)} \cdot \text{Score(adv)} + \text{Score(verb)}. \qquad (4)$$

In addition to AAN and VA phrases, negation words are given special consideration. Equations 5 and 6 define the scorings for negation-AAN and negation-VA phrases, which are based on Equation 3 and 4:

$$\text{Score("neg adv verb")} \equiv \text{Score(verb)} + \text{Pol(verb)} \cdot \text{Score(adv)} + \text{Pol(verb)} \cdot \text{Score(neg), and} \qquad (5)$$

$$\text{Score("neg adv adj")} \equiv \text{Score(adj)} + \text{Pol(adj)} \times \text{Score(adv)} + \text{Pol(adj)} \cdot \text{Score(neg)}. \qquad (6)$$

The sentence score is calculated based on the summation of all patterns we discussed previously. The details are shown in Algorithm 2 (see Figure 3). The higher/lower score means more strongly positive/negative.

1. Assign scores to all adjective and adverbs by Equations 1 and 2.
   Extract all phrases (P) and calculate Score(P) for each P by Equations 3, 4, 5, and 6.
2. $S \leftarrow \sum_{i=1}^{m} Score\,(P_i)$, where m is the number of phrases.

Figure 3. Algorithm 2. Calculating numeric scores for a sentence.

Bag-of-Words and Rule-Based Algorithm

Due to the special characteristics of social media texts, we define some domain rules to analyze sentiments. What people write on Facebook or Twitter is different in style from traditional product reviews or blog articles. Social media texts are often short and contain Internet idioms.

Table 3. Emoticon appearances in our datasets. We strip whitespace in preprocessing—that is, we take :) as :).

Emoticons	Number of appearances
:)	46
<3	15
:(	8
:-)	7
;)	6
=)	5
:D	5
:))	5
;-)	2
;P	2
:0)	1
=o)	1
^_^	1
;3	1

Our first rule is designed to capture emoticons. For example, many Facebook comments and Twitter tweets contain emoticons like :) for positive sentiments or :(for negative sentiments, which almost always convey the underlying sentiment. We believe that there are few cases where the underlying sentiment or polarity of the comment or tweet is different from that which the emoticon represents. This claim is verified in the experimental dataset: 103 out of 105 emoticon-containing sentences are correctly classified in the three-way sentiment classification. Emoticons would betray the true sentiment of a sentence if the author is being sarcastic, which is a notoriously difficult case for machines to classify.

In MuSES, we collect 77 positive emoticons and 59 negative emoticons from Wikipedia and 14 of them appear in our datasets. Table 3 lists the most frequently used positive and

Input: *S*, given sentence; *PS*, set of positive domain words and pattern rules; *NS*, set of negative domain words and pattern rules; *NG*, set of negation words; *PE*, set of positive emoticons; *NE*, set of negative emoticons.

Output: "*P*," "*N*," or "*O*" (sentiment of *S*)

```
if e ∈ S such that e ∈ PE ∪ NE then
     return Polarity(e)
else
W ← empty list; W' ← empty list; pos_counter ← 0;
neg_counter ← 0
split S and append each word to W without changing
their order in S
for each w ∈ W do
    append w to W' if w ∈ PS ∪ NS ∪ NG
end
for each w ∈ W' do
     1. ignore v and w if v, w ∈ NG and v
     immediately precedes w
     2. reverse sentiment of w if v ∈ NG and v
     immediately precedes w
     3. increment pos_counter if w ∈ PS
     4. increment neg_counter if w ∈ NS
 end
 return 'P' if pos_counter > neg_counter
 return 'N' if pos_counter < neg_counter
 return 'O'
 end
```

Figure 4. Algorithm 3. Bag-of-words and rule-based algorithm.

negative emoticons in our datasets. Emoticons are extremely powerful in inferring the underlying sentiment if they're present in the sentence. But they aren't very useful as a standalone classifier because only a small portion of the comment sentences will ever contain emoticons. A feasible way of harvesting the emoticon knowledge is to incorporate the emoticon rule in a learning model with other sentiment algorithms, which we explain in our experimental section.

In addition to emoticons, the second characteristic of social media texts is that they're often short,[7,8] using Internet language such as "1st!" "Thank you, Obama" "Go Bulls!" "Thumbs up!" and so on. In MuSES, we introduce an additional rule to process this situation: if the sentence belongs to the pattern of [thank you/go], [a company name/organization/a person's name], we label it as positive. Also, we've added some domain-specific keywords into sentiment word sets; for instance, "Yum, Yummy" is a positive word for food comments. Algorithm 3 (see Figure 4) presents the details.

Experimental Results

We conducted experiments on English Facebook comments, English Twitter tweets, and multilingual Amazon reviews. We manually performed three-way (positive, negative, or neutral) labeling for 500 English tweets; 1,000 Facebook English comments; 270 Amazon review sentences (from 20 full reviews) in Chinese; 180 Amazon review sentences (from 10 full reviews) in Korean; 180 Amazon review sentences (from 12 full reviews) in German; and 200 Amazon review sentences (from 20 full reviews) in English.

To unify the independent results from the three individual algorithms, MuSES employs high-level meta learning techniques. We use decision tree, neural networks, random forest model, and logistic regression as our sentimental classification models. We consider five discriminative features with these models. These include three basic and two derived features:

- S_1, integer output from compositional semantic rules;
- S_2, float output from the numeric sentimental identification algorithm;
- S_3, output from the back-of-word and rule-based algorithm ("*P*," "*N*," or "*O*");
- $S_1 + S_2$; and
- $S_1 - S_2$.

For model evaluation, we employ 10-fold cross-validation. We use different metrics including the weighted average values of precision, recall, F-measure, receiver operating characteristic (ROC), and accuracy.

Figure 5 compares the combined results versus individual algorithms across languages and datasets.

In Figure 5, the random forest model and neural networks perform better than decision tree and logistic regression in most cases. Our system performs consistently and strongly for English across different channels, while it has slightly varied performances across different languages and families of languages. On the Amazon review dataset, the performance in German is comparable to the performance in English. Although the performances on Asian languages are consistently lower than the performance on European languages, it's worth mentioning how relatively strong the decision tree classifier is on both Korean and Chinese reviews.

Figure 5. Results and comparisons for Facebook and Twitter data in English, and multilingual results on Amazon review data.

In addition to these experiments, we also tested the individual contributions from emoticons and domain-specific keywords. In Figures 2g and 2h, CSR is the performance just from compositional semantic rules; NSI is from numeric sentiment identification; CMB1 is the best meta learning result combining CSR, NSI, and emoticons; CMB2 combines CSR, NSI, emoticons, and domain-specific keywords; and CMB3 combines everything presented in this article, which shows the best results.

MuSES has handled multilingualism through a label-free, wikification-based solution. In the future, we plan to investigate collaborative sentiment-learning across multiple languages, instead of processing each language independently. ■

Acknowledgments

This work is supported in part by the following grants: US National Science Foundation awards CCF-0833131, CNS-0830927, IIS-0905205, CCF-0938000, CCF-1029166, and OCI-1144061; US Department of Energy awards DE-FG02-08ER25848, DE-SC0001283, DE-SC0005309, DESC0005340, and DESC0007456; US Air Force Office of Scientific Research award FA9550-12-1-0458.

References

1. J.C. Reynar and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," *Proc. 5th Conf. Applied Natural Language Processing* (ANLP), 1997, pp. 16–19.
2. K. Toutanova et al., "Feature-Rich Part-of-Speech Tagging with a Cyclic

THE AUTHORS

Yusheng Xie is a doctoral candidate in computer engineering at Northwestern University. His research interests include social media mining and natural language processing. Xie has a BS in computer engineering and applied mathematics from Northwestern University. Contact him at yxi389@eecs.northwestern.edu.

Zhengzhang Chen is a research staff member at NEC Lab America. His research interests include graph mining and anomaly detection. Chen has a PhD in computer science from North Carolina State University. Contact him at zzc472@eecs.northwestern.edu.

Kunpeng Zhang is an assistant professor at the University of Illinois at Chicago. His research interests include social media mining and machine learning. Zhang has a PhD in computer science from Northwestern University. Contact him at kzh980@eecs.northwestern.edu.

Yu Cheng is a doctoral student at Northwestern University. His research interests include machine learning and video mining. Cheng has a BS in computer science from Tsinghua University. Contact him at ych133@eecs.northwestern.edu.

Daniel K. Honbo is a software engineer at 4C Insights. His research interests include field-programmable gate arrays and distributed algorithms. Honbo has a PhD in computer science from Northwestern University. Contact him at dkh301@eecs.northwestern.edu.

Ankit Agrawal is a research associate professor at Northwestern University. His research interests include bioinformatics and materials informatics. Agrawal has a PhD in computer science from Iowa State University. Contact him at ankitag@eecs.northwestern.edu.

Alok N. Choudhary is a professor at Northwestern University. His research interests include high-performance computing and social media mining. Choudhary has a PhD in electrical and computer engineering from the University of Illinois, Urbana-Champaign. He is a Fellow of IEEE, ACM, and American Association for the Advancement of Science. Contact him at choudhary@eecs.northwestern.edu.

Dependency Network," *Proc. Human Language Technology and North American Chapter Assoc. Computational Linguistics Conf.* (HLT-NAACL), 2003, pp. 173–180.
3. J. Yao et al., "Using Bilingual Lexicon to Judge Sentiment Orientation of Chinese Words," *Proc. IEEE 6th Int'l Conf. Computer and Information Technology* (CIT), 2006, pp. 553–561.
4. Y. Choi and C. Cardie, "Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis," *Proc. Conf. Empirical Methods in Natural Language Processing* (EMNLP), 2008, pp. 793–801.
5. A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Affect Analysis Model: Novel Rule-Based Approach to Affect Sensing from Text," *Int'l J. Natural Language Eng.*, vol. 17, no. 1, 2011, pp. 95–135.
6. J. Liu and S. Seneff, "Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm," *Proc. Conf. Empirical Methods in Natural Language Processing* (EMNLP), 2009, pp. 161–169.
7. X. Hu et al., "Unsupervised Sentiment Analysis with Emotional Signals," *Proc. 22nd Int'l Conf. World Wide Web*, 2013, pp. 607–618.
8. X. Hu et al., "Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge," *Proc. ACM Conf. Information and Knowledge Management* (CIKM), 2009, pp. 919–928.