Biomedical Informatics Insights

ORIGINAL RESEARCH

Suicide Note Sentiment Classification: A Supervised Approach Augmented by Web Data

Yan Xu[1,2], Yue Wang[2,3], Jiahua Liu[2,4], Zhuowen Tu[2,5], Jian-Tao Sun[2], Junichi Tsujii[2] and Eric Chang[2]

[1]State Key Laboratory of Software Development Environment, Key Laboratory of Biomechanics and Mechanobiology of Ministry of Education, Beihang University, Beijing, China. [2]Microsoft Research Asia, Beijing, China. [3]School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, China. [4]Department of Computer Science and Technology, Tsinghua University, Beijing, China. [5]Lab of Neuro Imaging, Department of Neurology and Department of Computer Science, University of California, Los Angeles, USA. Corresponding author email: eric.chang@microsoft.com

Abstract

Objective: To create a sentiment classification system for the Fifth i2b2/VA Challenge Track 2, which can identify thirteen subjective categories and two objective categories.

Design: We developed a hybrid system using Support Vector Machine (SVM) classifiers with augmented training data from the Internet. Our system consists of three types of classification-based systems: the first system uses spanning n-gram features for subjective categories, the second one uses bag-of-n-gram features for objective categories, and the third one uses pattern matching for infrequent or subtle emotion categories. The spanning n-gram features are selected by a feature selection algorithm that leverages emotional corpus from weblogs. Special normalization of objective sentences is generalized with shallow parsing and external web knowledge. We utilize three sources of web data: the weblog of LiveJournal which helps to improve the feature selection, the eBay List which assists in special normalization of *information* and *instructions* categories, and the suicide project web which provides unlabeled data with similar properties as suicide notes.

Measurements: The performance is evaluated by the overall micro-averaged precision, recall and F-measure.

Result: Our system achieved an overall micro-averaged F-measure of 0.59. *Happiness_peacefulness* had the highest F-measure of 0.81. We were ranked as the second best out of 26 competing teams.

Conclusion: Our results indicated that classifying fine-grained sentiments at sentence level is a non-trivial task. It is effective to divide categories into different groups according to their semantic properties. In addition, our system performance benefits from external knowledge extracted from publically available web data of other purposes; performance can be further enhanced when more training data is available.

Keywords: sentiment analysis, suicide note, spanning n-gram, web data, supervised approach

Biomedical Informatics Insights 2012:5 (Suppl. 1) 31–41

doi: 10.4137/BII.S8956

This article is available from http://www.la-press.com.

Introduction

A person who has committed suicide often experienced a cumulative psychological process which ultimately led to the decision of ending his/her own life.[1–3] The suicide note provides us with the first-hand information about that person's particular mind status and mind logic.[4] Analyzing the internal emotions revealed in the suicide note might help us to identify people who potentially have suicide ideation, and thus prevent the misery from happening.

The task of 2011 i2b2 Challenge Track 2 is designed to identify/recognize/categorize sentence-level sentiments in the suicide notes.[5,6] The basic problem structure is divided into fifteen categories, including thirteen subjective categories (*abuse, anger, blame,fear,forgiveness,guilt,happiness_peacefulness, hopefulness, hopelessness, love, pride, sorrow, and thankfulness*) and two objective categories (*information, and instructions*). Each sentence may belong to none or multiple categories at the same time. 600 annotated notes were provided by the challenge organizers for training; another 300 notes were reserved for evaluating the competing systems.

The main difficulty in this task comes from the underlying ambiguity of the classification task and the limited amount of training data in each category. The entire task covers thirteen emotion categories and two additional factual categories, which are fine-grained, ie, to distinguish *anger* from *blame*, or *sorrow* from *hopelessness*. Since real-world suicide notes are very difficult to obtain (valuable too), the training data volume is relatively small with a highly skewed distribution. Out of the 4,633 sentences in the 600 training notes, only 2,173 sentences (46.9%) have one or more class labels. The top two categories having the most notes are *instructions* (820 sentences, 17.7%) and *hopelessness* (455, 9.8%); the smallest two categories are *forgiveness* (6, 0.1%) and *abuse* (9, 0.2%).

A wide variety of existing methods in sentiment analysis are not immediately applicable to the tasks in this challenge. A single classification system might not be sufficient. Existing feature selection algorithms, such as document frequency thresholding, information gain, mutual information, χ^2 statistic, and term strength, are also ineffective in dealing with the limited amount of training data and accidental features[7] in this case. Existing sentiment lexicons only consider limited number of categories and thus cannot be directly applied to this task; examples include positive/negative dictionaries from General Inquirer and six emotional categories from WordNet-Affect.

Our system aims to deal with the above difficulties by utilizing several web data resources in a multi-layer classification system.

- We divide the fifteen categories into three groups and handle each group differently. This is due to: (1) the expressions adopted by objective and subjective categories exhibit different properties; and (2) the effectiveness of machine learning method depends on the amount of available training data. For two objective categories (*information* and *instructions*) and eight subjective ones with relatively sufficient training data (*fear*, *guilt*, *hopelessness*, *love*, *sorrow*, *hopefulness*, *thankfulness*, and *happiness_peacefulness*), we use one-versus-the-all binary SVM classifiers.[8] For the remaining five subjective categories (*abuse*, *anger*, *blame*, *pride*, and *forgiveness*), which are either too infrequent or too subtle to be systematically learned, we adopt a pattern matching approach. Let us give an example of "My son has always been a clean, honest boy and man and all who meets him loves him."
- The candidate features for classifying the *two objective c*ategories and eight subjective ones are prepared differently. For the subjective ones, a feature *called spanning n-gram* is designed to better capture emotional expressions than traditional n-grams. Since *spanning n-gram* results in a high dimensional feature space, feature selection is performed with the help of semantically matched weblog corpora (LiveJournal[9] blogs with "moods"). For the objective ones, noun phrases containing daily items are identified by shallow parsing and with help of external knowledge (item-lists from eBay),[10] and then sentences are normalized before learning the bag-of-n-gram features.
- To alleviate the situation of having very limited amount of training data, we make an effort to collect additional data (posts from SuicideProject website)[11] with similar emotional contents as those in this challenge. A total number of 1,814 sentences are manually annotated according to the

classification schema. Using our trained model as a classifier, we further choose the sentences with high SVM classifier confidence score as the labeled data. Another 268 *information* and *instructions* sentences are then obtained in this way.

In this paper, we describe our system for the i2b2 challenge 2011sentiment classification task. Section 2 reviews previous research in sentiment classification and suicide note analysis. Section 3 gives detailed explanations for each component of our system. In Section 4, experiment results are shown to compare alternative approaches and validate our system. Section 5 discusses the submitted results and Section 6 performs error analysis. Finally, conclusions are drawn in Section 7.

Related Work

Sentiment analysis is an increasingly important topic in NLP. There are some related work in blog data, user reviews and customer feedback. Farah et al[12] focused on the strength of subjective expressions within a sentence or a document using specific 200 news articles. The experiments demonstrated that a combination of adjectives and adverbs plays a more effective role than pure adjectives. Chesley et al[13] used blog corpus to identify each post as being objective, positive, or negative; their experiments demonstrated that verbs and adjectives are strong features in the blog corpus. Yang[14] used blog corpus such as LiveJournal, Windows Live Spaces and Yahoo! Kimo blog to classify emotions using Conditional Random Fields (CRF) and SVM. This work is similar to our task in using LiveJournal. Mihalcea[15] exploited LiveJournal to find words related to happiness or sadness; their method can be used to create a "happiness" dictionary. Mishne[16] utilized blog information to predict the levels of various moods within a certain time. Aman[17] classified emotions using some dictionaries such as General Inquirer, WordNet-Affect and other features.

There is some related work in feature design for sentiment analysis. N-grams[18] are the most widely used features. N-grams may ignore contextual information such as negations, and valence shifters.[18] A prior sentiment lexicon[19] is useful for sentiment analysis, but it is difficult to accurately build a dictionary for each category in our task. Syntax information is treated as features to classify sentiment. Nakagawa[20] described a dependency tree-based method for subjective sentiment analysis using CRFs with hidden variables. Jiang[21] applied syntax information such as verb, adjective, noun, and adverb to confirm target information to classify positive, negative and neutral.

Research on classification for suicide note corpus has not attracted much attention. Pestian et al[1,4] used a feature selection strategy to tell genuine from elicited suicide notes by combining decision trees, classification rules, function models, lazy learners or instance-based learner, and meta learners. The accuracy of results by a learning-based algorithm, trainees, and mental health professionals are 78%, 49% and 63% respectively. Surprisingly, a learning-based approach achieves the best results. Matykiewicz et al[22] presented an unsupervised learning algorithm to distinguish actual suicide notes from newsgroups. The mean F-measure is 0.973 when the number of clusters varied from two to six. The experiment demonstrated that machine learning can be used to successfully detect suicide notes or newsgroups.

Methods

Figure 1 gives an overview of our system. Each sentence in a suicide note is first preprocessed (explained below) and then passed into three groups of one-versus-all binary classifiers, each deciding whether the sentence belongs to the corresponding category or not. The first group uses a linear SVM trained with spanning n-gram features (see a description below), accounting for eight subjective categories. Features are ranked and selected with the help of semantically matched corpora from LiveJournal. The second group accounts for the two objective categories. Items and locations are generalized by eBay product lists before the sentence is represented as 1–4 gram feature vector and learned by a linear SVM. The third group accounts for the remaining five infrequent/subtle subjective categories. Dictionaries of patterns are compiled from various resources and then matched against the sentence to determine its category. Finally, the output labels of each classifier are combined to assign categories for that sentence.

Sentence preprocessing

The preprocessing steps of a sentence include token normalization, spell checking and stemming.

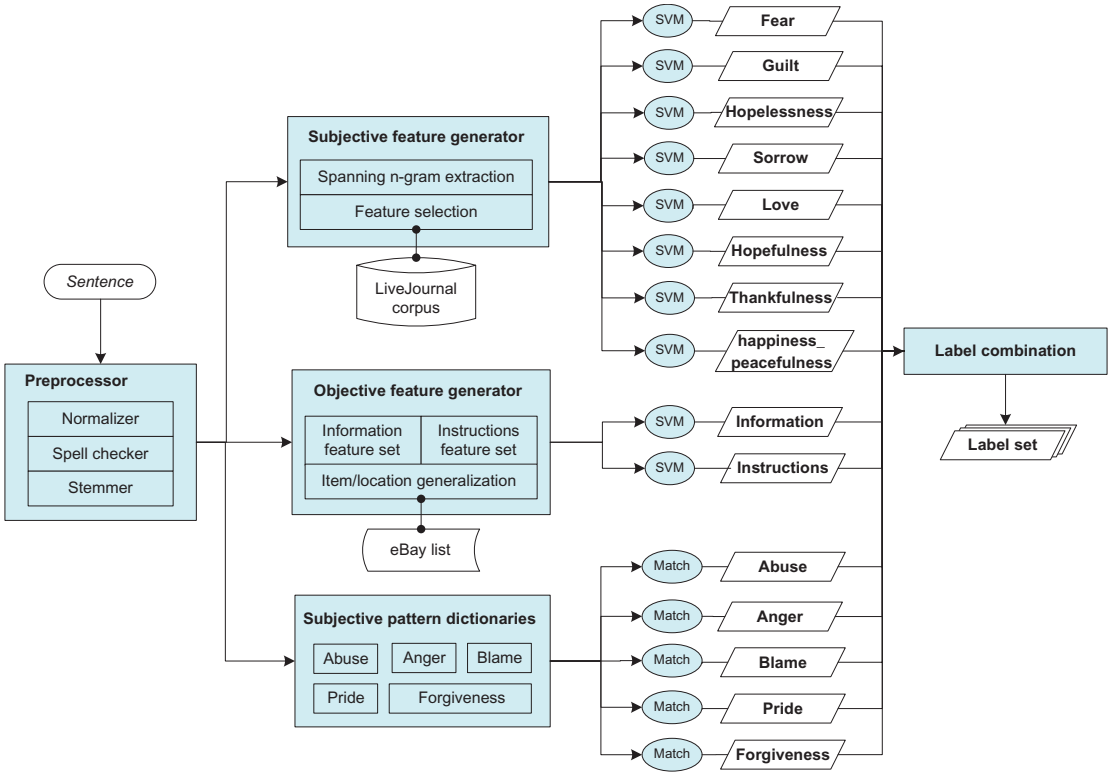

Figure 1. System architecture diagram.

Normalized tokens are represented by numbers, dates, relatives, titles, first and last names, place names. Placeholders are also normalized, such as consecutive underscores "–" and letter x in "I am xxxx from ____". Two spell checkers are used sequentially: the first one is compiled from training data, converting separate tokens into one token, such as "ca n't" to "can't"; the second one is Hunspell,[23] an open source spell checker. The stemmer utilizes a function provided by Hunspell.[23]

Subjective classification using spanning n-grams features

We begin this subsection by introducing our spanning n-gram features; then we illustrate the process of feature computing and ranking, and feature selection by leveraging LiveJournal and POS patterns, and training an SVM model.

Feature description

We design a feature called *spanning n-gram* extracted from a sentence. A spanning n-gram consists of a pair of n-grams which *spans* across several words. For example, <*take*, *any longer*> is a spanning n-gram from the sentence "I can't take it any longer.", and "it" is the word within the spanning n-gram. Specifically, in an m-word text window ($w_1, w_2, \ldots w_m$), we consider four kinds of spanning n-grams:

- *uni-uni* grams: $< w_i, w_j >$, $1 \le i \le j\ m$
- *uni-bi* grams: $< w_i, w_j w_{j+1} >$, $1 \le i \le j \le m - 1$
- *bi-uni* grams: $< w_i w_i{+}1, w_j >$, $1 \le i < i{+}1 < j \le m$
- *bi-bi* grams: $< w_i w_{i+1}, w_j w_{j+1} >$, $1 \le i < i{+}1 < j \le m - 1$

The intuition behind this feature design is to capture the subtle emotions in a sentence. First, some emotions are not expressed in any single word but by colloquial phrases comprised of common words. A spanning n-gram allows for a certain degree of variation within a traditional n-gram, such as the *uni-bi* gram <take, any longer> in *take* (*it*|*the pain*|*my life*) *any longer*. Second, some emotions are only differentiable by considering the subject and the complement together, as in "*I have been such a burden.*" (*guilty*) and "*Life has been a burden to me.*" (*hopelessness*). A spanning n-gram simultaneously considers two (relatively) distant n-grams, which can potentially capture distant dependencies such as *I … a burden* and *life … a burden*.

In our implementation, the window is bounded by punctuations in the sentence, and the maximum window size is set to be 8. Intuitively, a small window can barely capture distant dependencies, while a big window (spanning across several clauses) is too long to characterize a phrasal structure.

Feature computing and ranking

The number of extracted spanning n-grams is greater than their n-gram counterparts, resulting in a very high dimensional feature space. Given the limited size of training data, feature selection is performed in order for the SVM classifier to assign weights properly to most relevant features.

For each emotional category, we select a set of features by their relevance to that category, and use all these features to represent each sentence into a feature vector for statistical learning. The relevance is computed as a variant of odds ratio. In one-versus-rest classification settings, denote e as the positive category, $\bar{e}$ as the collection of all negative categories, $r(e, g)$ as the relevance of a feature gram g to the positive class e, then

$$r(e,g) = \begin{cases} P + f(e,g) & c(\bar{e},g) = 0 \\ f(e,g)/f(\bar{e},g) & c(\bar{e},g) > 0 \end{cases}$$

where

$c(e,g)$ is the total occurrences of gram g in category e;

$f(e,g) = c(e, g)/n(e)$ is the normalized frequency of gram g in category e;

$n(e)$ is the number of sentences in category e;

$f(\bar{e},g) = c(\bar{e}, g)/n(\bar{e})$ is computed similarly from negative categories $\bar{e}$;

P is a constant such that $P > \max_g r(e,g)$ when $c(\bar{e},g) > 0$.

When $r(e, g) < P$, higher $r(e, g)$ indicates that the gram g is more likely to indicate category e; $r(e, g)$ close to 1 means g is neutral to e; $r(e, g) = 0$ means that g is never seen in e. This motivates us to select features with $t_1 < r(e, g) < P$, where t_1 is a manually-set threshold.

When $r(e, g) > P$, g only showed up in category e. A high frequency $c(e, g)$ indicates that g is strongly related to e, which motivates us to select features with $r(e, g) > P$ and $c(e, g) > t_2$ where t_2 is a manually-set threshold.

Whether to select those features with a low $c(e, g)$, eg, $c(e, g) = 1$, is debatable. They could either be useful since they only appear in e, or irrelevant to e but did not appear in $\bar{e}$ due to data sparseness. To further determine the relevance of these infrequent spanning n-grams, we leverage external emotional text collected from LiveJournal.

Feature selection through LiveJournal corpus

The high dimensional feature space produced by spanning n-gram and the limited training data result in a sparse feature matrix. Common feature selection algorithms may not be directly applicable here. LiveJournal is a weblog space where users are able to tag his/her article with an optional "mood". The data from LiveJournal is a rich repository for sentiment analysis. Moods can be chosen from a list of 132 emotions. We select semantically matched moods to construct extended corpus for some of the suicide emotion categories, as shown in Table 1. If multiple LiveJournal moods are selected for one category, sentences from these moods are pooled together to form one LiveJournal mood set. We assume that people tend to use similar phrases to express similar emotions. For example, popular phrases in articles tagged "depressed", "crushed" or "frustrated" would also be more relevant to "hopelessness" category in suicide notes.

We compute the relevance of an infrequent spanning n-gram g to a suicide emotion category e as

Table 1. Suicide note emotions and similar LiveJournal moods.

Suicide note emotion e	LiveJournal mood tag e'
Anger	Annoyed, aggravated, angry, pissed off
Abuse	Embarrassed
Blame	Rejected, annoyed
Fear	Scared
Forgiveness	(No corresponding mood)
Guilt	Guilty
Happiness_peacefulness	Happy, cheerful, peaceful
Hopefulness	Hopeful, optimistic
Hopelessness	Depressed, crushed, frustrated
Love	Loved
Pride	Accomplished
Sorrow	Sad, gloomy
Thankfulness	Thankful, grateful

follows. Let e' be the corresponding LiveJournal mood set of e, $\bar{e}'$ be the collection of LiveJournal moods in Table 1 other than e, P' be the constant when $c(\bar{e}', g) = 0$. Then r(e', g) of a gram g to e' can be computed in a similar way as $r(e, g)$. Note that the portion of every mood set in $\bar{e}'$ should be proportional to every corresponding category in $\bar{e}$.

The infrequent grams with $t_3 < r(e', g) < P'$ or $r(e', g) < P'$ and $c(e',g) > t_4$ are selected based on the same reasoning as mentioned above; t_3 and t_4 are manually-set thresholds. Note that these grams are too sparse to receive proper weights from a statistical classifier. To alleviate this problem, we sort these grams into bins of part-of-speech (POS) patterns. In this task, the POS tagger in OpenNLP is used. For example, if *uni-bi* grams <trouble, cause you> and <misery, cause you> are selected for the category *guilt*, and tagged as "trouble/NN cause/VB you/PRP" and "misery/NN cause/VB you/PRP", then they are placed in *guilt*-NN_VB_PRP bin. In this way, feature sparseness is reduced by equivalence classes (emotion-POS bins). Each emotion-POS bin is a binary feature for SVM classifier. An emotion-POS feature is active when any gram in that POS bin appears in a sentence.

To this end, every emotion has a set of features: selected spanning n-grams and emotion-POS bins. These features are binary valued, representing the presence/absence of the feature. We concatenate these features to form the entire feature vector for subjective category classification.

Objective classification of *information* and *instructions*

The two objective categories, *information* and *instructions*, are mainly about the author's disposition of personal effects and properties. Since there are so many possible items and locations that could be mentioned in the disposition, it is useful to normalize such objective information to capture more similarity in *information* and *instructions* sentences. We normalize four kinds of phrases: daily items ("clothing", "briefcase", etc.), financial terms ("insurance", "cash", etc.), locations ("apartment", "closet", etc.) and location prepositions ("in", "on", "under", etc.). This takes the following steps.

First, four corresponding dictionaries are prepared. The "daily item" dictionary contains items appearing in training data. To cover as many daily items as possible, product category list from www.ebay.com is added. Since the data size is small, dictionaries of "financial term", "location" and "location preposition" are manually compiled from the training data. This process is facilitated by POS-tagging and chunking[24] the sentences of objective categories.

Second, the OpenNLP chunker[24] is used to chunk a sentence based on a shallow parser. The chunk helps to (1) determine the part-of-speech of a matched word, since words like "watch" "check" and "ring" are not items if they are used as verbs; (2) allow different modifiers before the same head noun, so that "my ring" "my diamond ring" and "the engagement ring" are all recognized as long as the dictionary contains the single word "ring". For each smallest unit produced by the chunk, if it contains any word from the above four dictionaries, it is normalized to the name of the dictionary. An example is shown in Figure 2. The sentence "Please find my insurance in the briefcase." is chunked as "[VP Please/VB find/VB] [NP my/PRP$ insurance/NN] [PP in/IN] [NP the/DT briefcase/NN] ./.". It is first normalized as "Please find _financial_term_ _location_prep_ _daily_item_".

Third, if both a preposition and the following noun phrase are normalized, then they are normalized to "_at_somewhere_". In the above example, the sentence is further normalized as "Please find _financial_term_ _at_somewhere_".

1–4 grams of the normalized sentence is used as features for information and instruction. In addition, the count of "_at_somewhere_" is a special feature for information; the presence of a funeral-related term ("bury", "cremation", etc.) is a special feature for instructions. A list of funeral-related terms is compiled from training data.

Subjective classification using pattern matching

Some subjective categories such as *anger*, *pride, abuse, blame,* and *forgiveness* have too few training sentences for statistical classifier to learn generalizable models. To detect emotions such as *anger*, *blame* or *pride* in a sentence, it is often necessary to identify the underlying reason accounting for that emotion. For example, "Today he bet again & lost thirty." may express *blame* because "he" was connected to the writer in some way, and the loss through gambling was an undesirable event. Accurate classification of these categories

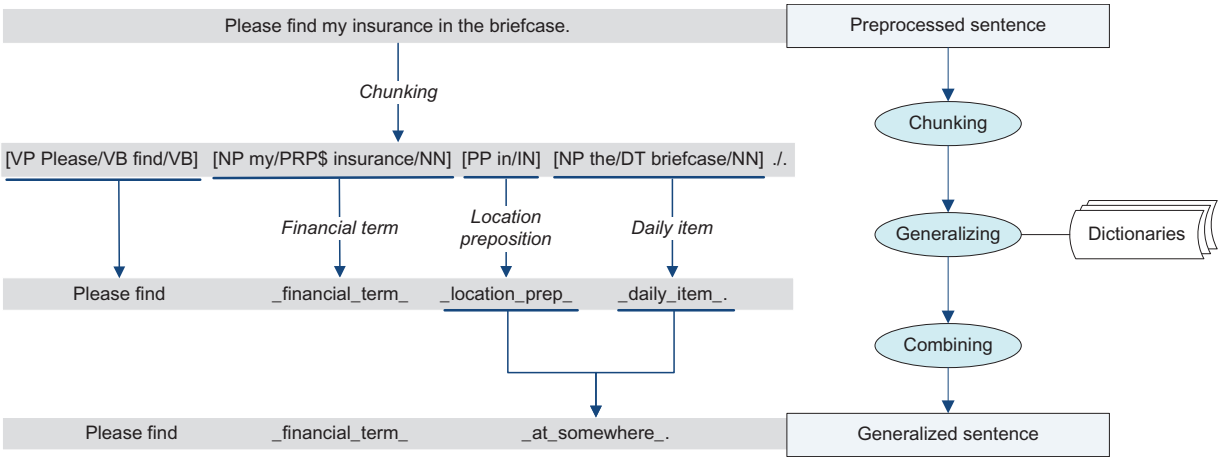

Figure 2. Objective information generalization.

will require deep semantic analysis and common sense inference. Given the limited system development time, we classify these categories by pattern matching method. Pattern dictionaries for each emotion are compiled from (1) highly relevant unigrams in training data; (2) highly relevant unigrams in LiveJournal corpus; (3) strong indication words and phrases from WordNet.[25] An emotion is assigned to a sentence if the sentence matches any pattern of that emotion.

Results

The performance of experiments was measured using three standard measures: precision (P), recall (R) and F-measure (F).

To compare different statistical classifiers, we used bag of 1–4 grams features to train SVM, Naïve Bayes and decision tree boosting classifiers[26] using 600 training data with 10-fold cross validation experiments. Table 2 is the micro-averaged results of the three classifiers. The experiment results demonstrated the results of SVM (tune bias of each category) have the best performance compared with other two classifiers. The results in Table 2 show that SVM is flexible. After tuning threshold parameter of each category, SVM ranks top in the three classifiers. In this task, SVM was chosen as the classifier for its flexibility.

The effect of the framework, ie, dividing fifteen categories into three groups, is shown in Table 3. The first row uses preprocessed sentences and 1–4 grams without feature selection, but categories *abuse*, *anger*, *blame pride* and *forgiveness* are not included to avoid false positives. The second row uses the framework as explained above. The F-measure of the system of dividing categories into three groups is improved by 4.96%.

The effect of spanning n-gram feature for eight subjective categories is shown in Table 4. All systems used bias-tuned SVM as classifiers. The baseline uses 1–4 gram features. When we replace the feature with unigram and four types of spanning n-grams without feature selection, both precision and recall improved by a margin. After feature ranking and selection with the help of LiveJournal corpus, the classification result is further improved.

The influence of item/location generalization for objective categories is shown in Table 5. The baseline uses pure 1–4 gram features. The results show that generalization greatly improves the performance of the two categories. To assess the effectiveness of knowledge from eBay, we conduct experiments with and without item normalization. We can see that the eBay knowledge contributes to a more significant

Table 2. 10-fold cross validation micro-averaged results using bag of 1–4 gram features as baselines (categories *abuse*, *anger*, *blame* and *pride* are not included to avoid False Positive).

	P	R	F
SVM	0.7767	0.2772	0.4085
SVM (tune threshold of each category)	0.4779	0.5325	**0.5038**
Naïve Bayes	0.5481	0.4088	0.4683
Boosting*	0.6497	0.3493	0.4543

Notes: *A multiclass boosting algorithm[26] is used. Weak classifiers: decision trees (depth = 2); number of iterations: 500. The default parameters are used for baseline experiment.

Table 3. Micro-averaged results for fifteen categories on test data (Evaluating the use of framework).

	P	R	F
1–4 grams	0.4911	0.5204	0.5053
Dividing categories into three groups	0.5305	0.5818	**0.5549**

performance gain for *information* than that for *instructions*.

We submitted three systems for the task. System 1 used the 600 notes provided by i2b2 organizers as training data and 300 notes as testing data. In System 2, extra labeled *information* and *instructions* sentences were added for training. In System 3, more sentences from all categories were added on the basis of the system 2. These labeled sentences originated from www.suicideproject.org, a website where people share stories about their painful thoughts and unbearable life. 220,000 unlabeled sentences from the web as testing data were imported into *information* and *instructions* models. We obtained all confidence ranking of the 220,000 sentences. The confidential sentences above the threshold were manually chosen. A total of 268 sentences (158 *information* and 110 *instructions*) were annotated. It is noted that we added sporadic labeled sentences of other categories when we labeled *information* and *instructions*. For system 3, we collected posts expressing similar emotions as in suicide notes, and manually labeled sentences following the same schema of this task. A total of 1,814 sentences were labeled.

The last test results were micro-averaged. Table 6 is the micro-averaged results for sentiment analysis in suicide notes. The experiment results demonstrated that adding extra labeled data can improve the overall performance.

Table 4. Micro-averaged results for eight subjective categories on test data (Evaluating spanning n-gram features and feature selection).

	P	R	F
1–4 grams	0.4993	0.5426	0.5201
unigram + spanning n-grams, not selected	0.5199	0.5469	0.5331
unigram + spanning n-grams, selected	0.5180	0.5815	0.5479

Table 5. Micro-averaged results for objective categories on test data (Evaluating item/location normalization and eBay knowledge).

	P	R	F
Information	0.2798	0.5865	0.3789
Information, normalization w/o eBay	0.3613	0.4135	0.3857
Information, normalization w/ eBay	0.3313	0.5288	**0.4074**
Instructions	0.6241	0.6649	0.6439
Instructions, normalization w/o eBay	0.6675	0.6832	0.6753
Instructions, normalization w/ eBay	0.6530	0.7094	**0.6801**

Discussion

The results show that better performance can be achieved when more data is supplied to machine learning classifiers, as previous work have demonstrated.[27] In System 2, as we added more *information* and *instructions* sentences, the model did improve for these two categories. Since some of these sentences also contained other sentiments, other categories also improved. As labeled sentences from all 15 categories were added for training, System 3 showed improvements on most categories. But there can be inconsistent annotations in these subjective sentences and this brought noise, which was the case of *pride* in System 3.

Some categories such as *abuse* and *pride* have too few instances to train robust classifiers. In these cases, SVM classifiers present a low recall and a high precision. When the bias of linear SVM is tuned to get an optimal single-class F-measure, recall rises and precision drops. This process may introduce significant amount of false positives (FP) since true positives (TP) appear at a small probability. The FPs may harm the overall micro-averaged F-measure. To preserve a higher F-measure, it may be wise to leave these small categories undone to avoid too many FPs, rather than tune the bias.

Error Analysis

Two major sources of errors are shared among all categories.

- No-label sentences are labeled: the system assigns a label to a sentence, but the gold standard assigns no label to it. Since more than half of the sentences bear no label in the gold standard, this type

Table 6. Micro-averaged results for sentiment analysis in suicide notes.

	System 1			System 2			System 3		
	P	R	F	P	R	F	P	R	F
Abuse	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Anger	0.17	0.04	0.06	0.17	0.04	0.06	0.20	0.08	0.11
Blame	0.44	0.18	0.25	0.44	0.18	0.25	0.44	0.18	0.25
Fear	0.30	0.23	0.26	0.30	0.23	0.26	0.33	0.31	0.32
Forgiveness	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Guilt	0.49	0.42	0.45	0.50	0.43	0.46	0.49	0.54	0.51
Happiness_peacefulness	1.00	0.50	0.67	1.00	0.50	0.67	1.00	0.69	0.81
Hopefulness	0.24	0.21	0.23	0.26	0.21	0.23	0.25	0.24	0.24
Hopelessness	0.52	0.66	0.58	0.54	0.66	0.59	0.64	0.66	0.65
Information	0.33	0.53	0.41	0.34	0.66	0.45	0.36	0.66	0.47
Instructions	0.65	0.71	0.68	0.65	0.73	0.69	0.69	0.72	0.71
Love	0.71	0.68	0.70	0.72	0.69	0.70	0.70	0.75	0.72
Pride	0.40	0.22	0.29	0.40	0.22	0.29	0.33	0.22	0.27
Sorrow	0.14	0.24	0.17	0.14	0.24	0.17	0.14	0.26	0.18
Thankfulness	0.45	0.84	0.59	0.45	0.84	0.59	0.46	0.89	0.61
Micro-averaged	0.53	0.58	**0.55**	0.53	0.60	**0.56**	0.56	0.62	**0.59**

of error accounts for a major portion of the false positives (FPs) in all categories. For example, nearly 2/3 FPs of the category *hopelessness* should have no labels according to the gold standard, but most of them are actually expressing despair and resignation. These sentences have no labels not because they belong to none of the categories, but because the human annotators did not reach an agreement.[6]

- Multiple labels compete with each other: for a sentence with k labels $\{L_1, L_2, \ldots L_k\}$, the system assign L_i to the sentence when L_j is expected. On one hand, this may cause false negatives (FNs). For example, if an *instructions* sentence contains detailed information such as personal names, addresses, and telephone numbers, it looks more like *information* and is not classified as *instructions*. On the other hand, label competing may also cause false positives. For example, when the positive category is *hopelessness*, a *blame* sentence in which the writer is blaming someone for causing his life unlivable can be misclassified as *hopelessness*. For example, one of such *blame* sentences is: "What is the use of going on. Life is n't worth living when your family wants everything after all my hard years work."

Besides, errors are also caused by word sense ambiguity. Some words, such as "please" (the beginning of an imperative sentence) and "call" (to contact someone by telephone), are usually strong indicators of *instructions*. But their senses changes with context, as in "nothing I do *pleases* you" and "your own sisters *called* you Mary". Because our approach stems the words back to its original form in preprocessing and adopts bag-of-n-grams features, it fails to disambiguate the word sense.

Lastly, FNs appear where the test data are too bizarre to be generalized by the model learned from training data. Since we adopt a one-versus-rest classification scheme, the size of positive training set is smaller than that of the negative training set. Therefore, the model learned by a linear binary SVM classifier is biased toward the negative side. If a positive test sentence contains no feature with significantly positive weight, the classifier will not be confident enough to assign a positive label. This is especially the case for some of the small emotion categories, where the training set is far from representative of that emotion.

Conclusion

In this paper, we have described a sentiment classification system by utilizing augmented web data and devising effective features. The task categories are systematically divided into three groups. The first type makes use of spanning n-gram features for subjective categories; the second one mainly focuses on bag-of-n-gram features for objective

categories; the third type is based on unigram feature for those subjective categories which need semantic understanding. The external web data are from three sources: the weblog of LiveJournal which assists the feature selection process in spanning n-gram features, the eBay List derived an item dictionary to help special normalization in *information* and *instructions*, and the suicide project web which provides the unlabeled data with the similar properties of suicide notes. The experiment results demonstrated the performance of our system outperforming the one from the average human annotator and those from 1–4 grams using three statistical classifiers. For the five emotion categories involving semantic level understanding, our approach is still preliminary due to limited development time. To classify these categories more accurately, we will focus on the semantic level features to better identify these emotions in a sentence.

Acknowledgement

This work was supported by Microsoft Research Asia (MSR Asia). The work was also supported by MSRA eHealth grant, Grant 61073077 from National Science Foundation of China and Grant SKLSDE-2011ZX-13 from State Key Laboratory of Software Development Environment in Beihang University. We would like to thank the organizers of the i2b2 NLP challenge, especially Dr. Ozlem Uzuner and Dr. Scott Duvall for their tremendous contribution. Z. Tu is supported by ONR N000140910099and NSF CAREER award IIS-0844566.

Disclosures

Author(s) have provided signed confirmations to the publisher of their compliance with all applicable legal and ethical obligations in respect to declaration of conflicts of interest, funding, authorship and contributorship, and compliance with ethical requirements in respect to treatment of human and animal test subjects. If this article contains identifiable human subject(s) author(s) were required to supply signed patient consent prior to publication. Author(s) have confirmed that the published article is unique and not under consideration nor published by any other publication and that they have consent to reproduce any copyrighted material. The peer reviewers declared no conflicts of interest.

References

1. Pestian J, Nasrallah H, Matykiewicz P, Bennett A, Leenaars A. Suicide note classification using natural language processing: a content analysis. *Biomedical Informatics Insights*. 2011;3:19–28.
2. Pestian JP, Matykiewicz P, Grupp-Phelan J, Lavanie SA, Combs J, Kowatch R. Using Natural Language Processing to classify suicide notes. *AMIA Annu Symp Proc*. 2008:1091.
3. Duch W, Matykiewicz P, Pestian J. Neurolinguistic approach to natural language processing with applications to medical text analysis. *Neural Netw*. 2008.
4. Pestian JP, Matykiewicz P. Classification of suicide notes using natural language processing. *Proceedings of ACL Bio NLP*. 2008:96–7.
5. The 2011 i2b2 sentiment analysis challenge. Available at: http://computationalmedicine.org/home-0.
6. Pestian J. Emotional Assignment. Available at: http://i2b2vacincinnati-2011-challenge-i2b2va-track-2@googlegroups.com.
7. Yang YM, Pedersen JO. A comparative study on feature selection in text categorization. *ICML*. 1997:412–20.
8. Light binary SVM. Available at: http://svmlight.joachims.org/.
9. The weblog of LiveJournal. Available at: http://www.livejournal.com/.
10. The eBayList. Available at: http://www.ebay.com.
11. The suicide project. Available at: http://www.suicideproject.org.
12. Farah B, Cesarano C, Reforgiato D. Sentiment analysis: adjectives and adverbs are better than adjectives alone. *ICWSM*. 2006:1–7.
13. Chesley P, Vincent B, Xu L, Srihari RK. Using verbs and adjectives to automatically classify blog sentiment. *AAAI*. 2006: Press.
14. Yang CH, Lin KH, Chen HH. Emotion classification using web blog corpora. *IEEE/WIC*. 2007:275–8.
15. Mihalcea R, Liu H. A corpus-based approach to finding happiness. *AAAI*. 2006:Press.
16. Mishne G, Rijke MD. Capturing global mood levels using blog posts. *AAAI*. 2006:Press.
17. Aman S, Szpakowicz S. Identifying expressions of emotion in text. *TSD*. 2007:Press.
18. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classi cation using machine learning techniques. *EMNLP*. 2002:79–86.
19. Wilson T, Wiebe J, Hoffmann P. Recognizing Contextual Polarity: an exploration of features for phrase-level sentiment analysis. *Computational Linguistics*. 2009;35(3):Press.
20. Nakagawa T, Inui K, Kurohashi S. Dependency tree-based sentiment classi cation using CRFs with hidden variables. *NAACL-HLT*. 2010:786–94.
21. Jiang L, Yu M, Zhou M, Liu XH, Zhao TJ. Target-dependent twitter sentiment classification. *ACL*. 2011:151–60.
22. Matykiewicz P, Duch W, Pestian J. Clustering semantic spaces of suicide notes and newsgroups articles. *ACL*. 2009:179–84.
23. Hunspell. Available at: http://hunspell.sourceforge.net/.
24. The OpenNLP. Available at: http://sharpnlp.codeplex.com/.
25. WordNet. Available at: http://wordnet.princeton.edu/.
26. Multiclass boosting algorithm. Available at http://people.csail.mit.edu/carreras/soft/boostree-0.1/README.
27. Chang E, Lippmann R. Improving wordspotting performance with artificially generated data. *ICASSP*. 1996.