Biomedical Informatics Insights

ORIGINAL RESEARCH

Using Ensemble Models to Classify the Sentiment Expressed in Suicide Notes

James A. McCart[1,2], Dezon K. Finch[1,2], Jay Jarman[1,2], Edward Hickling[2], Jason D. Lind[2], Matthew R. Richardson[2], Donald J. Berndt[1–3] and Stephen L. Luther[1,2]

[1]Consortium for Healthcare Informatics Research, [2]HSR&D/RR&D Center of Excellence, James A. Haley Veterans' Hospital, Tampa, FL. [3]University of South Florida, Tampa, FL. Corresponding author email: james.mccart@va.gov

Abstract: In 2007, suicide was the tenth leading cause of death in the U.S. Given the significance of this problem, suicide was the focus of the 2011 Informatics for Integrating Biology and the Bedside (i2b2) Natural Language Processing (NLP) shared task competition (track two). Specifically, the challenge concentrated on sentiment analysis, predicting the presence or absence of 15 emotions (labels) simultaneously in a collection of suicide notes spanning over 70 years. Our team explored multiple approaches combining regular expression-based rules, statistical text mining (STM), and an approach that applies weights to text while accounting for multiple labels. Our best submission used an ensemble of both rules and STM models to achieve a micro-averaged F_1 score of 0.5023, slightly above the mean from the 26 teams that competed (0.4875).

Keywords: sentiment analysis, machine learning, text analysis, i2b2 competition

Biomedical Informatics Insights 2012:5 (Suppl. 1) 77–85

doi: 10.4137/BII.S8931

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

Introduction

Suicide is a major public health problem. In 2007, suicide was the tenth leading cause of death in the U.S., accounting for 34,598 deaths, with an overall rate of 11.3 suicide deaths per 100,000 people.[1] The suicide rate for men is four times that of women, with an estimated 11 attempted suicides occurring per every suicide death.[1] Suicidal behavior is complex, with biological, psychological, social, and environmental risks and triggers.[2] Some risk factors vary with age, gender, or ethnic group and may occur in combinations or change over time. Risk factors for suicide include depression, prior suicide attempts, a family history of mental disorder or substance abuse, family history of suicide, firearms in the home, and incarceration.[2–8] Men and the elderly are more likely to have fatal attempts than are women and youth.[5]

Suicide notes have long been studied as ways to understand the motives and thoughts of those who attempt or complete a suicide effort.[9] Given the impact of suicide and other mental health disorders, the broad goal of organizers from the 2011 Informatics for Integrating Biology and the Bedside (i2b2) Natural Language Processing (NLP) shared task (track two) was to develop methods to analyze subjective neuropsychiatric free-text. To further that goal, this challenge focused on sentiment analysis, predicting the presence or absence of 15 emotions in suicide notes. Our team explored multiple approaches combining regular expression-based rules, statistical text mining (STM), and an approach that applies weights to text while accounting for multiple labels. Overall, our best system achieved a micro-averaged F_1 score of 0.5023, slightly above the mean from the 26 teams that competed (0.4875). The remainder of this paper includes an abbreviated literature review on sentiment analysis and then a discussion of the methods and results of our challenge submissions.

Background

Sentiment analysis is concerned with identifying emotions, opinions, evaluations, etc. within subjective material.[10] A sizable portion of research in sentiment analysis has focused on business-related tasks such as analyzing product and company reviews,[11–14] which are typically coherent and well written.[15] These analyses commonly focus on the polarity of words to classify whether a review is positive or negative. Unfavorable reviews can then be examined to identify and address negative mentions of products or services through a customer support function.

Correctly determining sentiment can be difficult for a number of reasons. First, the polarity of a word from a lexicon may not match when taken in context.[16] For instance, the word "reasonable" in a lexicon is positive, but the word takes a negative meaning in the sentence "It's reasonable to assume the crowd was going to become violent". Second, words may have multiple senses which change the meaning of a statement. For instance, the word "sad" can mean an experience of sorrow (eg, "I feel sad all the time"), but it can also indicate being in a bad situation (eg, "I'm in such a sad state"). Finally, multiple emotions, opinions, etc. may be contained in a single document, making interpretation at the document level more difficult. Thus, classification may be done at the word[17] or sentence[14] level of analysis, instead of the document[18] level of analysis.

Methods

The subsections below provide a description of the dataset, preprocessing done to the data, modeling techniques used, and finally how the techniques were combined together to create ensemble models.

Dataset

The entire dataset consisted of 900 suicide notes collected over a 70-year period (1940–2010) from people who committed suicide.[a] 600 of the notes were made available for training, with the remaining 300 held-out for testing submitted systems. All names, dates, and locations were changed in the notes. Everything else in the notes were typed as written, retaining all errors in spelling and grammar. The notes were split on sentences and tokenized.

For the competition, each sentence was reviewed by three annotators and assigned zero to many labels representing emotions/concepts (eg, ABUSE, INFORMATION, LOVE). The sentence-level inter-annotator agreement for the training and test dataset was 0.546. In both datasets, roughly half the sentences were assigned a label, with relatively few of those having multiple labels.

[a]A more in-depth description of the dataset is available in Pestian et al.[19]

Preprocessing

Both the training and test datasets were preprocessed before training or applying any models. A summary of changes made to the data are provided below.

- Contractions were separated at the apostrophe during the original tokenization process. Thus, the added white space was removed (eg, *ca n't* → *can't*).
- A number of contractions used asterisks in place of apostrophes. To standardize, all asterisks were replaced with apostrophes.
- A large number of misspellings were encountered while reading through the training notes. A two-step automated approach was used to help correct these errors. First, a custom dictionary was used to ignore and/or correct a small subset of words not present in the standard dictionary used in the second step. For instance, contractions without apostrophes (eg, *dont* → *donut* using the standard dictionary) and alternate spellings (eg, *tonite*, *thru*) were added to the custom dictionary. Second, HunSpell, an open-source spell-checker, was used with a standard United States English dictionary.

Modeling

In the data, a small but not insubstantial number of sentences had more than one label assigned (302 sentences or 6.51% of all sentences). To allow the use of a wide array of machine learning algorithms and toolkits the data were transformed from a multi-label to a single-label classification problem, where each label was converted into an independent single-label binary classification. The data were then formatted with each sentence as a row of data, along with the note ID, sentence number, and binary variables representing each of the 15 labels.

The following subsections describe the three different modeling techniques used with the newly formatted dataset. The purpose of investigating multiple techniques was to create ensemble models of complimentary methods. First, rules using regular expressions were created to find generalizable patterns—especially within labels with little data. Second, STM was used to discover more complex patterns of word usage and because classifiers based on machine learning generally perform better than rules on sentiment classification tasks.[13] Finally, a unique method of applying weighting schemes to text while accounting for multiple labels was investigated.

Rules

Rule-based systems have commonly been used for categorization of textual documents.[20] For this competition, rules were an attractive method due to the small sample size for many labels. Relying on machine learning algorithms alone for such labels would have likely resulted in unstable models. Thus, rules were used as a complimentary method. The purpose of the rule-based system was to discover phrases (rules) that made intuitive sense, were generalizable to the test data, and limited false positives. The semi-automated process used to generate rules for each label is described below.

1. Sentences were categorized as either being *positive* or *negative* for a label.
2. Each sentence in the *positive* set was parsed into *n*-gram candidate phrases, where *n* ranged from one to five.
3. Any phrases found in the *negative* set were discarded. In addition, duplicate phrases and one-word phrases subsumed by multi-word phrases were also removed. Removing one-word phrases was done to limit false positives because a single word may apply equally well in many contexts, whereas multi-word phrases were expected to be constrained in their usage.
4. The list of remaining phrases were then examined manually. Phrases without intuitive meaning for the label were discarded. For instance, the phrase "my oldest boy" was discarded for the ABUSE label, but "abusive behavior" was kept. Variations and expansions of the remaining phrases were created as necessary.

After the entire process, over 4,000 phrases/rules were retained (more than one rule may exist per sentence). Table 1 shows the breakdown of rules by label.

Statistical text mining

Although rules were created for each label, the patterns being matched in the rules were fairly simplistic and prone to overfitting—ie, looking for the exact same word usage. Therefore, STM was used as a complimentary method in hopes of discovering more

Table 1. Number of rules by label.

Label	No. of rules
Abuse	24
Anger	227
Blame	261
Fear	33
Forgiveness	18
Guilt	287
Happiness	51
Hopefulness	49
Hopelessness	37
Information	590
Instructions	2,093
Love	507
Pride	29
Sorrow	80
Thankfulness	158
All	4,444

robust models that have increased generalizability to the test set—especially among labels with larger sample sizes (eg, INSTRUCTIONS).

For the first step of the STM process, the data (ie, sentences) were transformed into a term-by-document matrix by converting all text to lowercase; tokenizing; removing stopwords and tokens with fewer than three characters; stemming; and finally removing terms that only occurred once in the data. The result was a term-by-document matrix with 1,895 terms and 4,633 documents (sentences).

Next, models using three distinctly different machine learning algorithms were trained: Decision Trees (DTs), k-Nearest Neighbor (kNN), and Support Vector Machines (SVMs). Table 2 summarizes the parameters used with each algorithm. Greater detail of the process and parameters used are given in the list below.

Table 2. Statistical text mining modeling parameters.

Algorithm	Parameters
Decision trees	
Term weighting	GR, LOR, X^2
Top *n* terms	10, 25, 50, 100, 250, 500, 1000, All
Split criterion	GI, GR
k-Nearest neighbor	
Term weighting	GR, LOR, X^2
Top *n* terms	25, 50, 100, 250, 500, 1000, All
k	1, 2, 5, 10
Support vector machines	
Term weighting	GR, LOR, X^2
Top n terms	0, 25, 50, 100, 250, 500, 1000, All
SVD dimensions	0, 25, 50, 100, 250

Abbreviations: GI, Gini index; GR, gain ratio; LOR, log odds ratio; X^2, Chi-square.

Decision Trees—The top *n* terms were selected as features based on their weight within the term-by-document matrix. Three term weighting formulas were used: gain ratio, log odds ratio, and chi-square.[21] Decision tree models based on C4.5[22] then used the presence or absence of the selected terms and split nodes using the Gini index or gain ratio.

k-Nearest Neighbor—Three factors were used in weighting the term-by-document matrix: (1) term frequency, (2) collection frequency, and (3) normalization factor.[23] Term frequency and cosine normalization were used for the first and third weighting factors, respectively. The same three term weighting formulas used in DT were used for the second weighting factor. Like in DT, the top *n* terms were selected; however, the weighted values of those features were used as inputs for the kNN models instead of the simple presence or absence of a term. Cosine similarity was used to evaluate sentences to one another with the number of neighbors (*k*) varying between 1, 2, 5, and 10.

Support Vector Machines—The same weighting procedure from kNN was used. In addition, Latent Semantic Analysis (LSA)[24] employing Singular Value Decomposition (SVD) was used as a dimension reduction technique. The top *n* terms and/or the top *m* SVD dimensions were used as features in a linear SVM classifier.[25]

Finally, the performance for each combination of parameters were compared using 10-fold stratified cross-validation,[26] where the weighting methods, selection of top *n* terms, and generation of SVD dimensions were all performed on the training folds and then applied to the validation fold. For each machine learning algorithm, the model with the highest F_1 score from the various combination of parameters was selected for each label. If no models for a label correctly predicted a single true positive, then no model was selected for that label—ie, all actual positive sentences would be false negatives.

Weights

In addition to STM, we also explored a method of applying weights to text while accounting for multiple labels. A total of four formulas based on chi-square[27] and a modified version of the Gini index[28] were used

Table 3. Weight formulas.

Formula	Description
X^2	Sum of chi-square values for each term in a sentence.
$\pm X^2$	Sum of chi-square values for each term in a sentence, with the sign of a term's weight determined by whether it exists predominately in the *positive* or *negative* set.
GI_{mod}	Sum of modified Gini index (Equation 1) for each term in a sentence.
$GI_{mod} * X^2$	Sum of modified Gini index multiplied by chi-square for each term in a sentence.

to generate weights. Equation 1 provides the formula for the modified version of the Gini index (GI_{mod}). Given a sentence with m terms, GI_{mod} adds up each term's proportion of existence between all *positive* and *negative* sentences for the specified label. For instance, a sentence with term A and B which exist 25% and 65% in the *positive* group of sentences would have a GI_{mod} value of −0.2. The modified Gini index used here differs from the traditional calculation in two ways: (1) absolute value is not used for differences in proportion and (2) the final sum value is not multiplied by 1/2. These modifications were made to retain the overall sign of a sentence to a label and to not artificially compress the final value.

$$GI_{mod} = \sum_{i=1}^{m} \frac{term_{i,pos}}{n_{pos}} - \frac{term_{i,neg}}{n_{neg}} \quad (1)$$

Table 4 summarizes the four formulas used to calculate weights along with a short description of their calculation. The formulas were used to create sets of features for input into data mining models—ie, for each formula used, a feature would be created for each label. The set notation { } (used in Table 4 below) represents which groups of formulas were used to create features. For instance, $\{GI_{mod}, \pm X^2\}$ indicates features calculated using the GI_{mod} and $\pm X^2$ formulas were included, whereas $\{All\}$ means features calculated from all four formulas were included.

Table 4. Weight-based modeling parameters.

Algorithm	Parameters
Decision trees	
Split criterion	ACC, GI, GR, IG
Feature sets	$\{\pm X^2\}$, $\{GI_{mod}\}$, $\{GI_{mod} * X^2\}$, $\{GI_{mod}, \pm X^2\}$, {All}
Logistic regression	
Feature sets	$\{\pm X^2\}$, $\{GI_{mod}\}$, $\{GI_{mod} * X^2\}$, $\{GI_{mod}, \pm X^2\}$, {All}
Support vector machines	
Kernel	Linear, Poly, Sigmoid, RBF
Feature sets	$\{\pm X^2\}$, $\{GI_{mod}\}$, $\{GI_{mod} * X^2\}$, $\{GI_{mod}, \pm X^2\}$, {All}

Abbreviations: ACC, Accuracy; GI, Gini index; GR, gain ratio; IG, information gain.

In addition to the weight-based measures of the text, features representing structural elements of the text were also included in all models. A description of the structural features are described in more detail below.

- **Note length**—Length of note this sentence came from in characters. We hypothesized that longer notes may be more associated with some emotions than others.
- **Sentence length**—Like note length, but at the sentence level.
- **Line position**—Normalized value between 1 and 100 representing the relative position of a sentence within a note. We thought there may be some common pattern in the order one might use when writing a note.

The weight and structural features described above were calculated for all sentences using distinct terms (after removing stop words). Three different machine learning algorithms were used: Decision Trees (DT), Logistic Regression (LR), and Support Vector Machines (SVM). Table 4 summarizes the parameters used with each algorithm. Greater detail of the process and parameters used are given in the list below.

Decision Trees—C4.5-based decision trees[22] were used. However, unlike the process used in STM, the numeric value of each feature was used instead of the simple presence or absence of a feature. In addition, two additional criteria were examined for splitting nodes: accuracy and information gain. As shown in Table 4, five different feature sets were included as inputs to the decision tree.

Logistic Regression—The same feature sets used in DT were also used. Models were created with logistic model trees, a method that builds trees with logistic regression models in their leaves.[29]

Support Vector Machines—The same feature sets used in DT and LR were also used. The performance

using four different kernels was investigated: linear, poly, sigmoid, and RBF.[30]

Finally, similar to the STM process, the performance for each combination of parameters were compared using 10-fold stratified cross-validation,[26] and the best performing models for each label and algorithm were selected.

Ensemble models

Ensemble models were used to capitalize on the strengths of different modeling techniques and methods (algorithms). Each method within an ensemble was given an equal vote. A sentence meeting or exceeding a set number of votes was predicted as "positive" for the specified label. A two-stage process determined the makeup of the ensembles.

The first stage focused on methods within a technique. All method combinations from the same technique were evaluated, allowing one, two, or three votes to decide on a positive classification. (Requiring only a single vote would increase recall at the expense of precision, whereas two or three votes would do the opposite.) For instance, STM had three methods for a total of seven combinations: {DT}, {kNN}, {SVM}, {DT, kNN}, …, {DT, kNN, SVM}. All seven combinations were evaluated using one vote, four combinations with two votes, and one combination with three votes; resulting in 12 evaluations. In addition, individual model performance within a method was also investigated. Poor model performance can hurt the micro-averaged F_1 score if there are far more false positives than true positives. Thus, three cutpoints based on the F_1 score of individual models were investigated: $\geq$0.00 (all), $\geq$0.10, and $\geq$0.20. Models not meeting a cutpoint were not included for that method. For instance, if a model predicting PRIDE for kNN got an F_1 score of 0.0454, it may be removed. Overall, a total of 36 evaluations were done for each technique (STM and weights).

The second stage combined methods from different techniques. The best two ensembles from each technique from the previous stage were selected. All combinations were done again (excluding combinations of only methods from the same technique), allowing one, two, or three votes using the same three cutpoints. For instance, assume R = rules; T_1 and T_2 = text mining ensemble 1 and 2; and W_1 and W_2 = weight ensembles 1 and 2. Example combinations include $\{R\}$, $\{R, T_1\}$, …, $\{R, T_2, W_2\}$. A total of 72 evaluations were done (24 per cutpoint).

For submission, the best ensembles from four categories were compared and the top three were submitted. The categories include (1) rules only; (2) rules and STM; (3) rules and weights; and (4) rules, STM, and weights. Rules were included in each category because of the likelihood of doing better with small sized labels.

Results and Discussion

Table 5 lists the F_1 score of the best models from the training data by technique, method, and label. In addition, the micro-averaged F_1 score is also provided for each method. Each of the techniques had one of the top three performing methods: rules (0.8396), STM using SVM (0.4630), and weights using DT (0.4206).

Noteable is the large discrepancy in performance between the rules and the other two techniques. Due to small sample sizes and time constraints, the rules were built using the entire training dataset. Thus, the performance on the training dataset was expected to be overly optimistic to what would be seen on the test dataset. However, the other two techniques both used stratified cross-validation to train and test models. Thus, the training results of STM and weight-based models were assumed to be more in-line with what performance could be expected with test data.

After finding the best models per technique and method, a variety of ensemble models were created and tested. The ensemble models selected from the training set and submitted for the test set are shown in Tables 6 and 7. Table 6 shows what methods were included in the ensemble, the cutpoint used, and overall performance measures, whereas Table 7 breaks down F_1 score by label. The first submission used only rules, while the other two submissions used a combination of rules with weights or STM.[b] (All ensembles required only a single vote to classify an instance as positive.)

The first submission for the test set demonstrated the rules were overfit, dropping almost 0.50 in F_1 score

[b]Since the rules were known to be overfit, the last two ensemble models were also calculated without including rules to get a more realistic performance estimate on the test set. Without rules, the submissions had F_1 scores on the training set of 0.4791 and 0.4821, respectively.

Table 5. Training set F_1 score by label and method.

Label	Rules	STM			Weights		
		DT	kNN	SVM	DT	LR	SVM
Abuse	0.8235	0.0000	0.0000	0.0000	0.5882	0.5714	0.5714
Anger	0.9466	0.0000	0.1622	0.1980	0.5758	0.6486	0.4138
Blame	0.9353	0.0484	0.1635	0.1569	0.3837	0.4487	0.2835
Fear	0.8889	0.1333	0.1923	0.1429	0.4681	0.2500	0.0000
Forgiveness	0.9091	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Guilt	0.8125	0.3158	0.4278	0.3601	0.4091	0.3368	0.2335
Happiness	0.8636	0.0741	0.0588	0.1395	0.0000	0.0000	0.0000
Hopefulness	0.7949	0.0000	0.0702	0.0345	0.0732	0.0000	0.0000
Hopelessness	0.7665	0.3280	0.3782	0.4149	0.3494	0.3569	0.3117
Information	0.8500	0.2481	0.3028	0.3410	0.2857	0.2310	0.0946
Instructions	0.8739	0.4554	0.4379	0.5449	0.4796	0.4435	0.4309
Love	0.8056	0.6497	0.6022	0.6497	0.6188	0.6122	0.5055
Pride	0.7500	0.0000	0.1481	0.1111	0.0000	0.0000	0.0000
Sorrow	0.8182	0.0000	0.0513	0.0339	0.0345	0.0000	0.0385
Thankfulness	0.8101	0.6869	0.6696	0.6559	0.0885	0.0000	0.0000
All	0.8396	0.4110	0.3972	0.4630	0.4206	0.4010	0.3326

(0.8396 to 0.3408). Many of the rules did not capture the same pattern of word usage in the test set (no true positives were found in eight of the 15 labels), leading to a substantial drop in recall. In addition, many of the patterns found in the training set also applied to sentences without the same specified label, generating a large number of false positives.

The second and third submissions fared better than rules alone, increasing the F_1 score by 0.1362 and 0.1615, respectively. Combining rules with weighting methods resulted in two more labels finding true positives (BLAME and FEAR). In addition, five of the seven labels with true positives found by the rules had an increased F_1 score.

While the third submission did not result in any additional labels finding true positives, it did perform the best over-all. Submission 3 had the highest F_1 score (0.5023) and recall (0.5055) and the second highest precision (0.4992). On an individual level, the third submission outperformed the first submission on six of the seven labels with true positives and the second submission on six of the eight labels with true positives.

The results of the third submission were analyzed for errors. A random sample of up to 50 false positives and 50 false negatives were examined for each label. Overall, a few common themes emerged.

- A clear delineation between various labels was difficult to discern. For instance, sentences incorrectly classified as INFORMATION instead of INSTRUCTIONS and vice versa.
- Complex language usage was not accounted for because our techniques employed shallow

Table 6. Training and testing performance by submission.

Submission	Rules	STM			Weights			Cutpoint	Results			
		DT	kNN	SVM	DT	LR	SVM		F_1	Precision	Recall	N
Training												
1	✓							20%	0.8396	0.9908	0.7284	1,854
2	✓				✓	✓		20%	0.7420	0.6795	0.8172	3,033
3	✓	✓		✓				20%	0.7228	0.6457	0.8208	3,206
Testing												
1	✓							20%	0.3408	0.5667	0.2437	547
2	✓				✓	✓		20%	0.4770	0.4865	0.4678	1,223
3	✓	✓		✓				20%	0.5023	0.4992	0.5055	1,288

Table 7. F_1 Score by submission and label.

Label	Submission 1		Submission 2		Submission 3	
	Train	Test	Train	Test	Train	Test
Abuse	0.8235	0.0000	0.7368	0.0000	0.8235	0.0000
Anger	0.9466	0.1290	0.7950	0.1111	0.9466	0.1290
Blame	0.9353	0.0000	0.7854	0.1842	0.9353	0.0000
Fear	0.8889	0.0000	0.7419	0.2222	0.8889	0.0000
Forgiveness	0.9091	0.0000	0.9091	0.0000	0.9091	0.0000
Guilt	0.8125	0.1791	0.7494	0.4233	0.6856	0.4677
Happiness	0.8636	0.0000	0.8444	0.0000	0.8636	0.0000
Hopefulness	0.7949	0.0000	0.7949	0.0000	0.7949	0.0000
Hopelessness	0.7665	0.1931	0.7157	0.4531	0.6680	0.5081
Information	0.8500	0.2119	0.7176	0.3519	0.6723	0.3793
Instructions	0.8739	0.4808	0.7335	0.5664	0.7313	0.5562
Love	0.8056	0.4952	0.7518	0.6437	0.6841	0.6541
Pride	0.7500	0.0000	0.7500	0.0000	0.7500	0.0000
Sorrow	0.8182	0.0000	0.8182	0.0000	0.8182	0.0000
Thankfulness	0.8101	0.4286	0.8101	0.4286	0.8018	0.6500
All	0.8396	0.3408	0.7420	0.4770	0.7228	0.5023

text analysis. For instance, errors were found in sentences with sarcasm (eg, "also am sorry you never cared" → ANGER), negation (eg, "... she doesn't love me ..." → not LOVE), and emotions stated in a general sense rather than expressed by the writer (eg, "... all us good men expect from the woman we love ..." → not LOVE).

- Wide variability in word usage and meaning made uncovering robust and generalizable patterns challenging, especially for rules. Having a document collection that spanned a 70-year period and included writers of heterogeneous backgrounds contributed to the variation.
- Finally, it was unclear why some sentences were or were not assigned to certain labels in the gold standard. It appeared some assignments were based on context from surrounding sentences, but others were not as apparent.

Conclusion

This paper described our team's submissions to the 2011 i2b2 NLP shared task competition (track two). Our submissions used individual and ensemble systems consisting of regular expression-based rules, STM models, and weight-based models. Our three submissions obtained micro-averaged F_1 scores of 0.3408, 0.4770, and 0.5023, with the best submission using a combination of rules and STM models. A review of incorrectly classified sentences highlighted four common themes: (1) fuzzy delineation between various labels, (2) complex language usage, (3) wide variability in word usage and meaning and (4) questionable label assignments. In the future, better results may be obtained by focusing on a smaller set of clearly distinct labels; incorporating a Natural Language Processing (NLP) pipeline to perform deeper text analysis; and employing thesauri or fuzzy-matching mechanisms to account for word variability.

Acknowledgement

This study was undertaken as part of the James A. Haley Veterans Hospital. Views expressed are those of the authors and not necessarily those of the Department of Veterans Affairs.

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. Centers for Disease Control and Prevention National Center for Injury Prevention and Control (NCIPC). Injury prevention and control: Data and statistics (WISQARS), 2011. URL http://www.cdc.gov/injury/wisqars/.
2. Shekelle P, Bagley S, Munjas B. Strategies for suicide prevention in veterans. Technical report, Department of Veterans Affairs, Health Services Research and Development Service. 2009.
3. Moscicki EK. Epidemiology of completed and attempted suicide: Toward a framework for prevention. *Clinical Neuroscience Research*. 2001;1(5): 310–23.
4. Miller M, Azrael D, Hepburn L, Hemenway D, Lippmann SJ. The association between changes in household firearm ownership and rates of suicide in the United States, 1981–2002. *Injury Prevention*. 2006;12(3):178–82.
5. Arango V, Huang YY, Underwood MD, Mann JJ. Genetics of the serotonergic system in suicidal behavior. *Journal of Psychiatric Research*. 2003; 37(5):375–86.
6. National Institute of Mental Health (NIMH). Suicide in the U.S.: Statistics and prevention. Technical Report 06-4594, National Institute of Health. Available from: http://www.nimh.nih.gov/health/publications/suicide-in-the-us-statistics-and-prevention/index.shtml.
7. Kessler RC, Borges G, Walters EE. Prevalence of and risk factors for lifetime suicide attempts in the national comorbidity survey. *Archives of General Psychiatry*. 1999;56(7):617–26.
8. Petronis KR, Samuels JF, Moscicki EK, Anthony JC. An epidemiologic investigation of potential risk factors for suicide attempts. *Social Psychiatry and Psychiatric Epidemiology*. 1990;25(4):193–9.
9. Shneidman ES, Farberow NL. Clues to suicide. *Public Health Reports*. 1956;71(2):109–14.
10. Wiebe JM. Tracking point of view in narrative. *Computational Linguistics*. 1994;20(2):233–87.
11. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: *Conference on Empirical Methods in Natural Language Processing*. 2002:79–86.
12. Cui H, Mittal V. Comparative experiments on sentiment classification for online product reviews. In: *21st National Conference on Artificial Intelligence*. 2006:1265–70.
13. Matsumoto S, Takamura H, Okumura M. Sentiment classification using work sub-sequence and dependency sub-trees. In: *Advances in Knowledge Discovery and Data Mining*. Springer Berlin/Heidelberg, Berlin, Heidelberg; 2005:301–11.
14. Kudo T, Matsumoto Y. A boosting algorithm for classification of semi-structured text. *Conference on Empirical Methods in Natural Language Processing*. 2004:1–8.
15. Gamon M. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In: *20th International Conference on Computational Linguistics*. Association for Computational Linguistics; 2004.
16. Wilson T, Wiebe J, Hoffmann P. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*. 2009;35(3):399–433.
17. Turney P. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *40th Annual Meeting of the Association for Computational Linguistics*. 2002:417–24.
18. Dave K, Lawrence S, Pennock DM. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. 2003:519–28.
19. Pestian JP, Matykiewicz P, Linn-Gust M, Wiebe J, Bretonnel Cohen K, Brew C, et al. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*. 2012;5 (Suppl. 1):3–16.
20. Hayes P, Weinstein S. Construe-TIS: A system for content-based indexing of a database of news stories. In: *Second Conference on Innovative Applications of Artificial Intelligence*, Washington DC; May 1990.
21. Lan M, Tan CJ, Su J, Lu Y. Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2009;31(4):721–35.
22. Quinlan JR. Induction of decision trees. *Machine Learning*. 1986;1(1): 81–106.
23. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *Communications of the ACM*. 1975;18(11):613–20.
24. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*. 1990;41:391–407.
25. Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995; 20(3):273–97.
26. Witten IH, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc, San Francisco, CA, 2nd ed. 2005.
27. Mantel N. Chi-square tests with one degree of freedom: Extensions of the Mantel-Haenszel procedure. *Journal of the American Statistical Association*. 1963;58(303):690–700.
28. Sakoda JM. A generalized index of dissimilarity. *Demography*. 1981;18(2): 245–50.
29. Landwehr N, Hall M, Frank E. Logistic model trees. *Machine Learning*. 2005;59(1–2):161–205.
30. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2011:2(3).