Biomedical Informatics Insights

Libertas Academica
FREEDOM TO RESEARCH

ORIGINAL RESEARCH

LASSA: Emotion Detection via Information Fusion

Ning Yu[1], Sandra Kübler[2], Joshua Herring[3], Yu-Yin Hsu[3], Ross Israel[4] and Charese Smiley[4]

[1]Assistant Professor, University of Kentucky, Lexington, KY, USA. [2]Assistant Professor, Indiana University, Bloomington, IN, USA. [3]PhD Candidate, Indiana University, Bloomington, IN, USA. [4]PhD Student, Indiana University, Bloomington, IN, USA. Corresponding author email: ning.yu@uky.edu

Abstract: Due to the complexity of emotions in suicide notes and the subtle nature of sentiments, this study proposes a fusion approach to tackle the challenge of sentiment classification in suicide notes: leveraging WordNet-based lexicons, manually created rules, character-based *n*-grams, and other linguistic features. Although our results are not satisfying, some valuable lessons are learned and promising future directions are identified.

Keywords: fusion, dependency parsing, character *n*-grams

Biomedical Informatics Insights 2012:5 (Suppl. 1) 71–76

doi: 10.4137/BII.S8949

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

Introduction

Suicide is a major public health issue: In 2008, among the leading causes of death in the US, suicide ranked tenth over all and second and third for the age groups between 25–35 and 15–24 respectively.[1] In order to propose effective suicide prevention strategies, one needs to understand the complex suicidal intension and behavior. Suicide notes provide first-hand materials to support such studies. The traditional suicide notes analysis relies heavily on manual observations and expert knowledge, which is time consuming, difficult to conduct, and unable to handle large amounts of information. Fortunately, the development of health informatics and the advance of Natural Language Processing (NLP) techniques offer rich methods and tools for analyzing suicide notes systematically and computationally. For example, Pestian et al found that machine learning algorithms can aid in distinguishing suicide notes from simulated notes[2] and in classifying suicide notes into classes such as emotional states[3] and that robust machine learning algorithms performed more consistently and accurately than mental health professionals.[2,3]

One computational approach to studying suicide notes is sentiment analysis, an NLP task that originated in the late 1990s and has attracted the attention of researchers and practitioners from different communities. Most studies have focused on determining what people are thinking about certain topics, products, or services by analyzing user-generated content on the Web, such as online reviews, blog posts, or twitter messages. Specific tasks include monitoring mood and emotion on Twitter;[4] differentiating opinions from facts;[5] detecting positive or negative opinion polarity;[6] determining opinion strength;[7] and identifying other opinion properties.[8]

The 2011 i2b2 NLP task was organized by the Informatics for Integrating Biology & the Bedside (i2b2) Center, a national center for biomedical computing. Track II of the i2b2 NLP challenge[9] is a sentiment classification task, but differs from other sentiment analysis tasks in terms of both the fine level of sentiment classes and the unique nature of the dataset: suicide notes. More specifically, track II is a sentence-level multi-label classification task. By this we mean that the target text unit for this challenge is a sentence, and each sentence can be labeled by zero or more classes. There are a total of 15 classes, seven of which carry negative sentiment (eg, ABUSE, FEAR), six carry positive sentiment (eg, FORGIVENESS, PRIDE), and two are neutral (ie, INFORMATION and INSTRUCTION). A long-term goal of this task is for a computer to suggest if a patient might attempt or die by suicide, by automatically finding emotions that are highly associated with suicide notes in text generated by or associated with the patient, for example, blog posts or clinical records.

Due to the complexity of the task and the subtle nature of sentiments, we implemented both machine learning and ad-hoc rule-based classifiers and explored various features including WordNet-based lexicons, manually created rules, character-based *n*-grams, and other linguistic features.

Dataset and Preprocessing

There are a total of 600 suicide notes in the training set and 300 in the test set. When distributed to the participants, each suicide note has automatically been split into sentences, with name, address, and date information anonymized. The length of each note ranges from two words to more than seven hundreds words. All suicide notes have been annotated manually by volunteers. For each sentence, annotators were asked to identify passages that belong to the predefined 15 classes. As the result, zero or more classes were assigned to each sentence. More details about data annotation can be found in the overview paper for i2b2 NLP challenge.[9] Since there is no explicit class defined for sentences that fall into none of those classes, we defined a class OTHER for our machine learning experiments.

An initial investigation of the training data indicates that the number of notes varies dramatically across classes. Table 1 shows the distribution of training data over the 16 classes, with the first column recording the original number of sentences belonging to each class. This skewed training dataset causes a bias for machine learning classifiers. One possibility to avoid a classifier bias would be to conduct a two-level classification: first classify sentences as POSITIVE, NEGATIVE and NEUTRAL, then further classify each group into specific classes. However, we decided not to use a two-stage classification after further examining the training data because those

Table 1. Training data distribution over classes (number of sentences).

Class	Original no.	Re-segmented no.
Abuse	9	14
Fear	25	25
Sorrow	51	60
Anger	69	84
Blame	107	117
Guilt	206	223
Hopelessness	455	478
Forgiveness	6	6
Pride	15	18
Happiness-peacefulness	25	28
Hopefulness	47	48
Thankfulness	91	105
Love	290	311
Information	294	312
Instructions	813	863
Other	2460	2460

upper classes are not mutually exclusive. For example, one sentence "Sorry to my son with all a mother's love" can be labeled as both GUILT (negative) and LOVE (positive).

Because of some untypical punctuation in the original text (eg, using "-" instead of ".") and segmentation errors during pre-processing (eg, treating the period in "Mr." as sentence-final punctuation, not as part of an abbreviation), there are training examples that consist of multiple or partial sentences. In order to properly part-of-speech (POS) tag and syntactically parse the training data as well as to provide relatively clean training data to the machine learning classifiers, we re-segmented the training data. To do so, a script as a finite state machine that goes through the input one character at a time and makes decisions about what to do based on previous context. For example, on seeing a period, the script checks whether it is part of an abbreviation, of a number, or whether it is sentence-final. The second column in Table 1 shows the number of sentences after re-segmentation.

Lexicons and Ad-hoc Rules

For each emotion class, we created a list of related words based on WordNet in two steps: First, we manually selected a small number of seed words specified by POS type and word sense (eg, the word "fear" as a noun under the first word sense on WordNet for the class FEAR); Then, we automatically retrieved related words including synonyms and hyponyms for each seed from WordNet. As this is a multi-label classification task, one word can appear in more than one lexicon.

By inspecting terms with high frequencies in training data, we created a second list of words and patterns for each class. For example, verbs such as "tell" or "notify" tend to occur in sentences labeled as INSTRUCTIONS, and the phrase "best/only way out" is associated with the class HOPELESSNESS.

The two lists of terms, the list extracted from WordNet and the list of manually identified terms, were then merged in the form of regular expressions and used as rules for a simple ad-hoc rule-based classifier. An average of 20 words/patterns were created for each class. If rules for more than one class were applicable for the target sentence, this sentence was labeled with all these class labels.

Tagging and Parsing

As described in the section on the dataset and on preprocessing, we re-segmented the training data, with more specific rules for splitting sentences. We then performed POS tagging and dependency parsing on the re-segmented data. We used the Markov model tagger TnT[10] for its state-of-the-art handling of noisy, informal data with a high percentage of unknown words.[11] TnT was trained using a model generated from a combination of data from the Penn Treebank[12] and from CReST.[13] CReST is a small corpus with dialogues in a collaborative search scenario, which was used for its colloquial speech patterns that cannot be found in the Penn Treebank. In order to annotate the spontaneous speech data, CReST uses an extended tagset, including VBI, for imperative verbs, and DDT, for substituting demonstrative pronouns such as in "that is correct." The POS tagged training data were then hand-corrected before we created the training model. We removed the XY tag introduced by the CReST data for non-words as it only appeared on incorrectly tokenized words. We then used the model trained on in-domain data to POS tag the test data.

The dependency parser MaltParser 1.3[14] was used for parsing the training data with a model trained on

Penn Treebank data. Then, we created a model based on the training data, incorporating the modified POS tag set for parsing the test set. Due to time restriction, no hand corrections were made.

Experiments

Since our previous research[15] shows that the character-based language models worked well for opinion detection in various data domains, we used them for our machine learning experiments. We modified the sentiment analysis model in LingPipe[16] for this specific challenge. Both the default 8-gram character language model and token-based language model were trained. For the token-based language model, we used word trigrams, POS tag trigrams (eg, "PRP VBP JJ","VBP JJ" or "PRP VBP"), and dependency word pairs (ie, head-dependent).

Single character placeholders were used to replace the already anonymized addresses, names, and dates in order to avoid highly frequent *n*-gram features generated from them.

Besides experiments using the single ad-hoc rule-based and the machine learning classifiers alone, we also conducted fusion runs that combined both machine learning and ad-hoc rule-based classification.

Results and Discussion

Table 2 shows the test performance of different classifiers trained on the cleaned sentences. Overall, the machine leaning approach using both character 8-grams and dependency relations produced higher F-score than either the ad-hoc rule-based approach or a fusion of the ad-hoc rule-based approach and a machine learning approach. Character-based *n*-grams outperformed word-token *n*-grams, POS features, and dependency relations. Replacing names, addresses, and dates with single character placeholder only slightly improved performance.

Table 2. Performance of different classifiers trained on re-segmented training data.

Feature type	F-score	Precision	Recall
Ad-hoc (1)	0.31	0.26	0.37
Word trigrams (2)	0.38	0.49	0.31
Word trigrams w/placeholder (3)	0.38	0.50	0.31
POS tag trigrams (4)	0.32	0.40	0.26
Dependency relations (5)	0.40	0.53	0.33
Character 8-grams (6)	0.41	**0.57**	0.32
Character 8-grams w/placeholder (7)	0.41	**0.57**	0.32
Fusion: 1 + 5	0.38	0.32	0.46
Fusion: 1 + 7	0.40	0.29	**0.62**
Fusion: 5 + 7	**0.42**	0.45	0.40
Fusion: 1 + 5 + 7	0.40	0.32	0.53

The results show that the single ad-hoc rule-based classifier was the most inefficient one, with an F-score as low as 0.31. The reason for the low F-score is that emotions have to be classified at a very fine level; thus the classifications requires context information, which is not given in matching words or patterns within the target sentence. For example, if "regret" occurs in the context of "I regret that", it is an indicator for class GUILTY, but not if it appears in "I have no regrets"; "forgive" in "I forgive what you did" is an indicator for class FORGIVENESS, but is an indicator for class GUILTY in "Please forgive me."

The machine leaning classifiers reported in Table 2 were all trained with 16 classes including OTHER using re-segmented training data. For each example to be classified, a conditional probability score (*P*(*Class/Input*)) is returned for each class. Different thresholds for this score were examined, with higher thresholds producing better precision and lower thresholds higher recall. The best F-score was achieved by setting the threshold to 0.55. As shown in Table 2, character-based *n*-grams outperformed the ad-hoc rule-based classifier by 32% in F-score.

Character-based *n*-grams and the dependency relation pairs achieved similar performance and a fusion of both result sets yield the best performance, 0.42 in F-score in our experiments. The highest precision, 0.57, was reached by character-based *n*-grams (with and without placeholders). The highest recall, 0.62, was reached by a fusion of ad-hoc rules and character-based *n*-grams.

We also trained the machine learning classifiers with the original training data, which has fewer training examples but more words per example than

Table 3. Performance of 8-gram character-based classifiers trained on original and re-segmented training data.

Training data type	F-score	Precision	Recall
Re-segmented	0.41	0.57	0.32
Original	0.42	0.55	0.34

Table 4. Performance of 8-gram character-based classifiers with and without the OTHER class.

Training classes	F-score	Precision	Recall
Without OTHER	0.44	0.40	0.48
With OTHER	0.42	0.55	0.34

the re-segmented training data. Surprisingly, the performance is slightly better, as shown in Table 3. Because of time constraints, we could not re-segment the test data. This probably resulted in a discrepancy between training and test data in the experiment using the re-segmented training data.

We also investigated the influence on using the OTHER label. Since there are 2,460 examples in the class of OTHER, it is the majority label, which, in the machine learning approaches, caused a bias to label an unknown example as OTHER. To avoid this bias, we trained the classifiers using only the 15 classes, without OTHER. We then changed the label to OTHER if the best prediction for an unknown example has a probability score lower than 0.9. The results in Table 4 show that the approach that uses the OTHER label results in higher precision but also in lower recall so that the overall F-score is lower than the one for the experiment introduces OTHER after classification.

Conclusion and Future Work

From the experiments presented here, we conclude that character *n*-grams and dependency pairs are two robust sources of information for classifying emotions in suicide notes. Word *n*-grams as well as POS *n*-grams are less robust. We assume that the linguistic annotation contains more important information, for example, about the scope of negations. For this reason, we are planning on extracting more specific information from the linguistic annotation. For this challenge, we simply combined the resulting sets. In the future, more sophisticated fusion strategies will be investigated. We will also investigate using user-generated content similar to suicide note from the WWW for a semi-supervised approach.

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), 2011. URL http://www.cdc.gov/injury/wisqars/.
2. Pestian JP, Matykiewicz P, Grupp-Phelan J. Using natural language processing to classify suicide notes. In: *BioNLP* 2008*: Current Trends in Biomedical Natural Language Processing*, Columbus, Ohio, 2008:96–7.
3. Pestian JP, Nasrallah H, Matykiewicz P, Bennett A, Leenaars A. Suicide note classification using natural language processing: A content analysis. *Biomedical Informatics Insights*. 2010;3:19–28.
4. Bollen J, Mao H, Zeng XJ. Twitter mood predicts the stock market. *Journal of Computational Science*. 2011;2(1):1–8.
5. Wiebe J, Wilson T, Bruce R, Bell M, Martin M. Learning subjective language. *Computational Linguistics*. 2004;30(3):277–308.
6. Abbasi A, Chen H, Salem A. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Transactions on Information Systems*. 2008;26(3).
7. Tsou BKY, Yuen RWM, Kwong OY, Lai TBY, Wong WL. Polarity classification of celebrity coverage in the Chinese press. In: *Proceedings of the International Conference on Intelligence Analysis*, McLean, VA, 2005.
8. Kim SM, Hovy E. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*. Sydney, Australia, 2006:1–8.
9. Pestian JP, Matykiewicz P, Linn-Gust M, Wiebe J, Cohen K, Brew C, et al. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*. 2012;5 (Suppl. 1):3–16.
10. Brants T. TnT—a statistical part-of-speech tagger. In: *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics and the 6th Conference on Applied Natural Language Processing (ANLP/NAACL)*. Seattle, WA, 2000:224–31.
11. Kübler S, Scheutz M, Baucom E, Israel R. Adding context information to part of speech tagging for dialogues. In: *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT)*. Tartu, Estonia, 2010:115–26.
12. Marcus M, Santorini B, Marcinkiewicz MA. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*. 1993;19(2):313–30.
13. Eberhard K, Nicholson H, Kübler S, Gunderson S, Scheutz M. The Indiana “Cooperative Remote Search Task” (CReST) Corpus. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta, 2010.
14. Nivre J. *Inductive Dependency Parsing*. Springer Verlag, 2006.
15. Yu N, Kübler S. Filling the gap: Semi-supervised learning for opinion detection across domains. In: *Proceeding of the Fifteenth Conference on Computational Natural Language Learning*, Portland, OR, 2011:200–9.
16. Alias-i. LingPipe (Version 4.0.1). Available from http://alias-i.com/lingpipe., 2008. URL http://alias-i.com/lingpipe.