Biomedical Informatics Insights

ORIGINAL RESEARCH

Statistical and Similarity Methods for Classifying Emotion in Suicide Notes

Kirk Roberts and Sanda M. Harabagiu

The University of Texas at Dallas, Richardson, TX.

Corresponding author email: kirk@hlt.utdallas.edu; sanda@hlt.utdallas.edu

Abstract: In this paper we report on the approaches that we developed for the 2011 i2b2 Shared Task on Sentiment Analysis of Suicide Notes. We have cast the problem of detecting emotions in suicide notes as a supervised multi-label classification problem. Our classifiers use a variety of features based on (a) lexical indicators, (b) topic scores, and (c) similarity measures. Our best submission has a precision of 0.551, a recall of 0.485, and a F-measure of 0.516.

Keywords: similarity method, statistical method, sentiment classification, suicide notes

Biomedical Informatics Insights 2012:5 (Suppl. 1) 195–204

doi: 10.4137/BII.S8958

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

Introduction

The 2011 i2b2 Shared Task on Sentiment Analysis of Suicide Notes[1] evaluated automatic methods for detecting emotions in suicide notes from actual suicide patients. The goal of the task is to enable researchers to develop systems that may improve the care of suicide patients by detecting the emotions present in their notes, and thus inform the health care providers about their emotional state. The task considered 15 emotions: ABUSE, ANGER, BLAME, FEAR, FORGIVENESS, GUILT, HAPPINESS_PEACEFULNESS, HOPEFULNESS, HOPELESSNESS, INFORMATION, INSTRUCTIONS, LOVE, PRIDE, SORROW, and THANKFULNESS. These emotions were manually annotated on 900 suicide notes at the sentence level, with 600 being distributed to the shared task participants as training data and the remaining held for evaluation.

We have used this training data to develop a hybrid method that combines two types of approaches. The first approach is based on statistical methods designed to extract words and phrases that are indicative of a particular emotion. Our goal is to build a lexicon of words and phrases for individual emotions similar to Mohammad and Turney.[2] For example, we want to have the ability to recognize that the phrase "forgive me" is indicative of GUILT, while "I hate you" is indicative of ANGER. This lexicon is collected from both the training data as well as a large, unlabeled corpus. The second approach is a series of similarity metrics for finding sentences that evoke similar emotions. The similarity metrics utilize both word overlap as well as more robust topic-based representations. The statistical and similarity methods are both complementary and mutually reinforcing, as they both capture sentences containing common emotional expressions, but are also able to capture rarer and less overt expressions of the author's emotional state.

Previous Work

Constructing a customized lexicon is a common approach in both sentiment analysis and emotion detection. Taboada et al[3] discusses the use of lexicons for general-purpose sentiment detection (ie, positive and negative polarity), while Elliott[4] presents an early lexicon-based method to detect emotion. The WordNet Affect Lexicon[5] consists of several hundred words, annotated with the emotion they evoke and organized into a emotion hierarchy. Mohammad and Turney[2] use Mechanical Turk to annotate over two thousand terms with the eight emotions drawn from Plutchik.[6] We chose not to employ their methodology directly due to both privacy concerns of placing words on Mechanical Turk as well as the difficulty of selecting which terms might evoke an emotion. Inspection of the training data reveals that many sentences do not contain any single emotion-evoking word (eg, "I can't go on."), and extending Mohammad and Turney's method to phrases would require significantly more crowd-sourced annotation. Instead, we generate our lexicon using the training data (manual annotations) and a large, unlabeled corpus (automatic annotations). This allows us the flexibility of determining whether any word or phrase can evoke an emotion.

Similarity-based methods have been utilized as well. For example, Strapparava and Mihalcea[7] use Latent Semantic Analysis (LSA) to perform dimensionality reduction on a large corpus. They compare the latent representation of the query text with the individual emotion word's representation in order to determine a similarity. In contrast, we use a nearest neighbor based approach to find the most similar sentences for the query sentence. This allows us to recognize numerous different ways an emotion can be evoked, as we do not depend on a "centroid" representation.

Approaches

In order to produce a multi-label classification of emotions, we need to extract a variety of features from the suicide notes. Some of them are explicit, others are implicit. We use statistical techniques to detect both explicit indicators of emotions (words and phrases that directly evoke an emotion) as well as implicit indicators (such as words or topics often associated with an emotion). Both types of indicators are important as language allows for both explicit emotion declaration and more subtle implication of the author's emotional state. In addition, we use several similarity metrics to evaluate the "emotional distance" between sentences in the training and the testing sets.

Statistical distillation of emotion-bearing phrases from the training data

The most accessible means of learning phrases associated with a given emotion is to discover phrases associated with that emotion in labeled data. This captures both common explicit references to emotion and

common implicit phrases that frequently occur in a specific emotion's context. For example, in the i2b2 training data, the phrase "can't go on" is highly associated with the emotion HOPELESSNESS, "please forgive me" is associated with GUILT, and "bless you" is associated with LOVE.

We perform a statistical dependence test for each possible phrase/emotion pairing, where the null hypothesis is that the phrase and emotion are unrelated and only co-occur by coincidence. To calculate this we use pointwise mutual information (PMI):

$$\text{pmi}(e;x) = log\,\frac{p(x\,|\,e)}{p(x)}$$

where $p(x|e)$ is the probability of the phrase x occurring in a sentence labeled as emotion e, and $p(x)$ is the probability of seeing phrase x in the training data.

Here we consider a phrase to be any fixed number of tokens (we experimented with 1, 2, and 3-token phrases) as opposed to a syntactic definition of phrase. Examples of top phrases for the most common emotions are shown in Table 1. When classifying new sentences, phrases above a given threshold are extracted and matched to their associated emotion, where the exact threshold is specific to the individual feature. For more details on the actual features used, see the Feature types section. Additionally, we experimented with Fisher's exact test as an alternative to PMI, but while it proved successful in previous work,[8] it did not have a positive impact on this task.

Statistical distillation of emotion-bearing words from unlabeled data

In addition to collecting phrases from a small, labeled emotion corpus, words associated with emotions can be collected from a large, unlabeled corpus. Instead of using the manually labeled sentences, we utilize the emotion-evoking terms drawn from WordNet Affect.[5] We perform no word sense disambiguation. Rather, WordNet Affect is transformed from a sense-based inventory to a surface-form lexical inventory that matches all possible senses of a word. Sentences containing a term from WordNet Affect are assumed to evoke the emotion that term is associated with in the WordNet Affect ontology (eg, "afraid" evokes FEAR, "ecstatic" evokes HAPPINESS). Our source of unlabeled data is the English Gigaword corpus,[9] which contains over 8.5 million newswire articles. Due to the size of the data, only individual words are considered, as phrases of length two or more would require significantly more processing. PMI is again used to determine the most statistically indicative words.

We manually identified 21 emotions from the WordNet Affect ontology that would best correspond to the i2b2 emotions as well as additional high-level emotions that might prove useful. These chosen emotions are: *emotion, mental-state, positive-emotion, negative-emotion, anxiety, liking, dislike, hate*, *joy* (for HAPPINESS), *contentment* (PRIDE), *love* (LOVE), *gratitude* (THANKFULNESS), *calmness* (HAPPINESS_PEACEFULNESS), *positive-fear* (FEAR), *positive-hope* (HOPEFULNESS), *sorrow* (SORROW), *sadness* (SORROW), *regret-sorrow* (GUILT), *anger* (ANGER), *forgiveness* (FORGIVENESS), and *despair* (HOPELESSNESS).

The primary limitations of this approach are (1) the assumption that sentences containing an emotion term actually evoke that emotion, and (2) emotion in newswire is lexically expressed similar to transcribed suicide notes. However, this approach will find emotion-evoking words not present in the small, labeled i2b2 corpus, and features based on these words (see the Feature types section) do prove useful at detecting emotions in suicide notes. Alternatively,

Table 1. Representative phrases chosen from the i2b2 training data by statistical phrase discovery.

Emotion	1 token	2 tokens	3 tokens
INSTRUCTIONS	Notify	my funeral	I would like
HOPELESSNESS	helpless	any more	can't go on
LOVE	Bless	love Jane	you have been
INFORMATION	debts	life insurance	in my purse
GUILT	fault	Please forgive	God have mercy
BLAME	wreck	trouble with	you have done

unsupervised detection of topic can also cluster words indicating the same emotions, thus allowing the discovery of many more emotion bearing words. We therefore use topic modeling as a means of discovering additional features.

Topic modeling

Related sentences can be clustered using *topic modeling* to create clusters based on implicit topical information. Topic modeling techniques, such as latent Dirichlet allocation (LDA),[10] can discover cross-document similarities even when sentences have no words in common. We use the MALLET implementation of LDA[11] and treat every sentence as its own document.

LDA then considers every sentence as a bag-of-words. It assumes each sentence is associated with a probabilistic mixture of topics, and each topic is composed of a probabilistic mixture of words. For example, one topic might deal with family and contain words such as "love", "dear", and "daughter". Another topic might be more financial in nature and contain words such as "money", "debt", and "payment". With LDA, the granularity of the topics may be adjusted by increasing or decreasing the total number of topics. Additionally, LDA is completely unsupervised, so it can operate over a large amount of unlabeled data.

We used LDA for modeling topics because we believe there is a relationship between topics and emotions. For instance, sentences about health are likely to address the reason for the author's suicide and convey an emotion like Hopelessness. A sentence discussing financial issues is likely to contain Information. And a sentence topically related to religion is likely to evoke Forgiveness or Thankfulness. Table 2 contains the results of running LDA on the i2b2 training data with 10 topics (word casing has been removed). As can be seen in the table, common words such as "i", "you", and "have" are present in many topics since they do not add much topical information, while words like "bank", "dollars", "check", and "purse" are co-located in a single topic (Topic 1), suggesting financial information. Other topics (eg, topic 7) do not seem to form cohesive topic clusters. This is likely a result of running LDA over sentences instead of documents, as sentences are less likely than documents to have clearly defined topics.

Given an unlabeled sentence, the results of running a topic model can then easily be used to find similar sentences in the training data (see Similarity metrics) and the sentence's inferred topics can be directly used as features in a classifier (see Feature types). Importantly, LDA's compact topic

Table 2. Top words for each topic determined by LDA from the i2b2 training data.

Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
i	My	i	johnson	my	i	you	they	i	who
me	you	am	john	i	my	i	food	you	your
we	find	so	cincinnati	johnson	have	my	who	have	father
up	box	way	burnet	john	been	love	their	do	god
not	one	not	jane	take	has	john	where	know	pray
out	two	do	my	bill	life	jane	too	what	peter
my	car	sorry	smith	money	no	me	down	me	world
when	which	ca	ohio	pay	death	good	mountain	would	let
go	also	no	please	want	which	your	destroyed	not	lord
would	book	just	hospital	give	own	please	then	can	people
here	check	out	bill	have	one	have	just	could	days
over	dollars	have	children	funeral	time	forgive	many	only	where
after	get	more	mary	insurance	years	dear	business	want	man
should	other	me	ave	body	made	god	carry	one	other
time	purse	my	signed	they	last	so	family	did	soul
last	here	going	call	jane	long	much	up	than	these
years	back	want	january	please	since	very	warm	your	death
again	letter	take	give	not	being	mother	wear	like	light
night	bank	go	oh	can	things	mary	along	say	no
one	keys	know	phone	no	who	always	between	about	before

representation generalizes well to valid semantic spaces, so if two sentences are in similar topics, they likely evoke similar emotions. Additionally, sentences containing similar emotions can be found through the use of similarity metrics.

Similarity metrics

Given the importance of implicit information in emotion detection, it is difficult to devise universal rules for what constitutes an emotional statement. Rather, this is often defined empirically by the task's annotated training data. The relatively low inter-annotator agreement on this data confirms this, reported by the organizers as 0.546. Thus, instead of designing methods that extract information from a sentence so that a classifier may decide what emotion is present, we focus on methods that find similar sentences and their emotions. In this case the classifier's role is merely to weigh the result of multiple similarity metrics, thus simplifying the learning problem. We experimented with numerous similarity metrics but settled on just three: unweighted token overlap, tf-idf weighted token overlap, and topic similarity.

Unweighted token overlap treats both sentences as bags of words and measures the percentage of tokens the two sentences have in common. In set notation:

$$\text{overlap}(S_1, S_2) = \frac{|S_1 \cap S_2|}{\max(|S_1|, |S_2|)}$$

where S_1 and S_2 are the non-unique words in the two sentences, and $|S|$ indicates the number of words in the sentence.

Tf-idf weighted overlap is simply a weighted version of token overlap designed to favor rarer words. The weights are assigned using term frequency-inverse document frequency, the standard means of assigning term importance in the field of information retrieval. Term frequency (tf) is simply the number of times the word appears in the sentence. Inverse document frequency (idf) is the inverse of the number of documents a term appears in a given corpus (we use English Gigaword). We smooth the document frequency by assigning a minimum document count of 10 for rare words. This weighting method therefore gives greater importance to rarer words and almost no weight to stop words and punctuation, as they are present in nearly every document.

Topic-based similarity differs from word overlap similarity metrics in that it can find similar sentences that have few or no words in common. LDA assigns topic distributions to both documents (sentences in our case) and words. Typically, topic-based similarity metrics would use the topic distribution associated with the sentence. However, given the short length of sentences relative to the documents that topic models typically use, the sentence topic distribution can be quite noisy. Instead, we average the topic distributions for each word in the sentence in order to get an overall topic distribution. The topic distributions of two sentences are then compared using the inverse Jensen-Shannon divergence:

$$\text{JS}(D_1, D_2) = \frac{1}{2}\text{KL}(D_1 \,||\, M) + \frac{1}{2}\text{KL}(D_2 \,||\, M)$$

where D_1 and D_2 are the two topic distributions, M is the average of the two distributions, and KL(A||B) is the Kullback-Leibler divergence:

$$\text{KL}(A \,||\, B) = \sum_{k=1}^{K} A_k \log \left| \frac{A_k}{B_k} \right|$$

Jensen-Shannon is simply a symmetric extension to Kullback-Leibler. Jensen-Shannon has proved useful in calculating the similarity of two probability distributions in many NLP applications.[12] These three metrics are used to compute the most similar sentences to a query sentence.

Features based on these similarity metrics can then use k-nearest neighbor (KNN) style classification in order to indicate the emotions in similar sentences from the training data. Since KNN is a computationally expensive $O(n^2)$ operation, we pre-cache all possible sentence distances and the nearest 100 neighbors for each sentence. This caching process takes approximately one hour per similarity metric on a single CPU core. See the Feature types section for more details on these features.

Classification Framework

The approaches described in the previous section are integrated into a supervised classification framework, shown in Figure 1. The exact choice of features is optimized relative to the classifier using an automated feature selection technique.

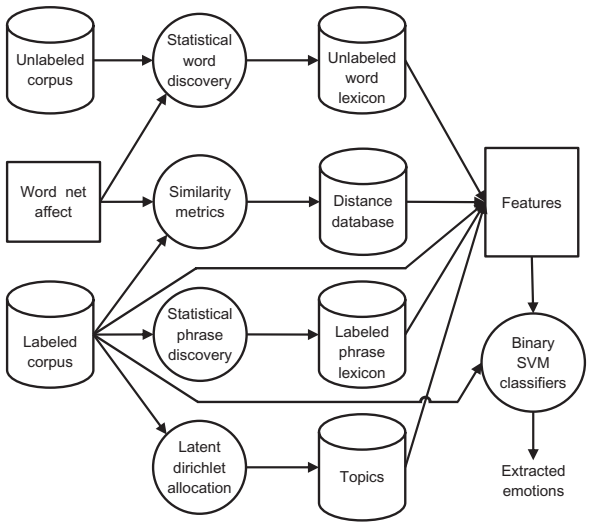

Figure 1. System architecture.

Classifier

We utilize a series of binary SVM classifiers[13] to perform emotion detection. Each classifier performs independently on a single emotion, resulting in 15 separate binary classifiers. The combination of these separate classifiers can be thought of as a single multi-label classifier, allowing for a sentence to be annotated with zero or more emotions. If multiple binary classifiers return a positive result, the sentence may have more than one emotion; if every binary classifier returns a negative result, the sentence has no emotions. While it would be possible to use separate features for each binary classifier, many of the emotions have very few training instances so this might lead to over-fitting. Additionally, SVMs allow for a bias parameter to be set to weight an individual outcome, which is useful when dealing with rare outcomes as a high frequency outcome will always be chosen over a very low frequency outcome if both outcomes are given equal weight. This would lead to good precision but very low recall for many of the emotions in this task. We set the bias parameters for each outcome to the inverse probability of that outcome in the training data (eg, BLAME constitutes 2.1% of all emotions in the training data, so the positive output in the BLAME classifier has weight 0.979 and the negative output has weight 0.021).

Feature types

Based on the methods outlined in the Approaches section as well as a bag-of-words baseline, we created the following feature types (or templates):

- *SentenceUnigrams*: A baseline bag-of-words feature.
- *StatisticalLabeledPhrases*(*phrase_size*, *threshold*): Returns all phrases in the sentence judged to be statistically indicative for any emotion. Parameters specify the phrase size (number of tokens) and the minimum PMI threshold.
- *StatisticalUnlabeledWordSum*(*emotion*): Real-valued feature that calculates the sum of word scores for the given emotion from the unlabeled data based on the WordNet Affect ontology. Unlike the feature based on statistical phrases from labeled data, the words from unlabeled data might not be present in the training data and therefore these features must be directly tied to a specific emotion.
- *StatisticalUnlabeledWordStrongest*(*emotion*): Real-valued feature that indicates the score of the strongest word for the given emotion instead of the sum.
- *TopicScore*(*topic*): The score for a given LDA topic for the sentence.
- *MostCommonEmotion*(*sim_type*,*num_neighbors*): A k-nearest-neighbor feature that indicates the most common emotion from a sentence's nearest neighbors. All sentences from the training data are considered as potential neighbors. Parameters specify which of the three similarity metrics (unweighted token overlap, tf-idf weighted token overlap, topic similarity) to use as well as the number of neighbors to consider.
- *StrongestEmotionScore*(*emotion*, *sim type*, *num neighbors*): A real-valued feature that returns the similarity between the current sentence and the nearest neighbor that contains the given emotion. Parameters include the emotion, the similarity measure, and the maximum number of nearest neighbors to consider before returning a similarity of zero.

Feature selection

The feature types previously discussed (along with many more not discussed here) have far too many parameterizations and combinations to manually select the best subset of parametrized features. Rather, we use an automatic feature selection technique known as floating forward feature selection[14] or greedy forward/backward. This method iteratively improves the set of features using a greedy selection. Each iteration is composed of a 'forward' step, which adds at most one feature, and a 'backward' step, which

removes already added features. In the forward step, all unused features (ie, all possible parameterizations of the feature types above) are individually tested in combination with the current set of chosen features. The single feature that improves the cross-validation performance the most is added to the chosen feature set. If no new feature improves the performance, the algorithm terminates. In the backward step, features in the chosen set that hurt cross-validation performance are removed. Intuitively, over time, some features may become redundant or even harmful after new features are added, so pruning the chosen set can improve performance. The result of running feature selection on a 5-fold cross-validation of the training data is shown in Table 3. These are the features used in our official submission to the i2b2 emotion detection task. All features from Table 3 are used in all 15 classifiers. While features such as "StrongestEmotionScore (ANGER, *token*, 15)" seem to target a specific emotion, they may be useful in other classifiers as well. Additionally, this allowed us to run our feature selection algorithm just once, using the scores for all 15 classifiers in order to guide the feature selection process, as opposed to running feature selection separately for all 15 emotion classifiers.

The feature selector chose features from each of the approaches discussed in the Approaches section, suggesting that they add complementary information. Two features were chosen based on the lexicon built from the labeled data–one uses 2-token phrases and the other uses 3-token phrases. The fact that a higher threshold was chosen for the 3-token phrases suggests that 3-token phrases can be quite noisy, so a higher threshold is necessary to filter all but the most indicative phrases. Two features were also chosen from the lexicon built from the unlabeled data, using the WordNet Affect emotions *forgiveness* and *positive-fear*. It is difficult to determine why these two were chosen instead of others, but given that FORGIVENESS and FEAR were two of the rarest annotations in the training data, it is likely that the classifiers for other emotions were able to effectively use these features as well. The topic score for topic 8 was chosen, which is a topic that deals with the author's actions and thoughts. This likely was useful for distinguishing between the typical emotions and the two command-like emotions INFORMATION and Instructions. Finally, six separate features were chosen based on all three of the described similarity metrics. While most of the similarity features deal with specific emotions (such as the two that use SORROW), they can still be useful for other emotion classifiers as well. For instance, knowing that no similar sentence has a strong SORROW score can help the positive classification of the emotions PRIDE and LOVE. The similarity feature based on NONE (ie, the sentence has no emotion) was probably useful for making negative classifications for each emotion classifier.

Table 3. Features selected through automatic feature selection.

SentenceUnigrams
StatisticalLabeledPhrases(2, 2.0)
StatisticalLabeledPhrases(3, 3.0)
StatisticalUnlabeledWordSum(*forgiveness*)
StatisticalUnlabeledWordStrongest(*positive-fear*)
TopicScore(8)
MostCommonEmotion(*topic*, 5)
StrongestEmotionScore(LOVE, *tfidf*, 50)
StrongestEmotionScore(SORROW, *tfidf*, 10)
StrongestEmotionScore(SORROW, *token*, 5)
StrongestEmotionScore(ANGER, *token*, 15)
StrongestEmotionScore(NONE, *topic*, 50)

Results

We submitted three runs to the 2011 i2b2 Shared Task on Sentiment Analysis of Suicide Notes. The first run ("SVM-Binary") is described in the Classification framework section. The second run ("SVMBinary-Top5") only used the binary classifiers for the five most frequent emotions in the training data: INSTRUCTIONS, HOPELESSNESS, LOVE, INFORMATION, and GUILT. In our experiments, we noticed many of the less common emotions had very poor precision, resulting in a significant number of false positives. In cross-validation experiments over the training data, the overall F1-measure improved if the emotions with fewer than 200 training instances were ignored, leaving only the five most common emotions. The third run ("SVM-Multi") was a single multi-class SVM classifier which chose at most one emotion per sentence. The official results for the three runs are shown in Table 4.

The best performing of the three runs was SVMBinary, with an F1 of 0.51589. Limiting output to only the five most frequent emotions (SVMBinaryTop5) improved precision only slightly compared to the loss of recall, therefore failing to improve performance on the test

Table 4. Official results for our three submissions as well as the mean, median, and bag-of-words result.

Submission	#	Precision	Recall	F1
SVMBinary	1120	0.55089	0.48506	0.51589
SVMBinaryTop5	1048	0.55725	0.45912	0.50345
SVMMulti	1020	0.54020	0.43318	0.48080
Mean submission				0.4875
Median submission				0.5027
Bag-of-words	2108	0.34108	0.56525	0.42544

set. Further post-hoc experiments show there is some merit to this idea, however, as a run which uses the seven most frequent (instead of five) emotions (75 or more instances) would result in an F1 of 0.51824. The single classifier (SVMMulti) performed worse than SVMBinary in both precision and recall. The best performing submission (SVMBinary) was both above the mean submission (0.4875) and the median submission (0.5027).

The bag-of-words result in Table 4 shows how our classifier setup would work if the only features available to it were the words in the sentence (ie, a purely un-ordered lexical approach with no additional knowledge or processing). We consider this to be a baseline approach that shows how our other features are able to generalize emotion from words, as all of our features use the same bag-of-words representation as input. Only the statistically extracted phrase features include any kind of word ordering, and none of our features use any syntactic or semantic processing information. Additionally, the only external knowledge source we use is WordNet Affect. Thus the improvement from the bag-of-words approach to the SVMBinary submission (a different in F1 of 0.09045) can be viewed as the ability of the features in Table 3 to extract emotion from sentences with minimal additional information.

A detailed per-emotion breakdown of the results is shown in Table 5. The results are generally proportional to then umber of annotations in the training data, which was expected. The false negatives are likely caused by an insufficient breadth of data to reveal the many ways in which an emotion can be textually expressed. The false positives are likely caused by a combination of inconsistency in the manual annotations and an insufficient amount of data to properly learn an emotion's textual properties. Our approach performs poorly on low frequency emotions such as Fear and Forgiveness, which contain 25 and 6 train instances, respectively, in the training data, and 13 and 8 instances, respectively, in the test data. This is consistent with many machine-learning techniques, which perform poorly when only a few examples are available to the classifier. The Fear classifier never fired on any sentence in the test data, and the Forgiveness classifier only fired once, resulting in a false positive. Furthermore, reducing the threshold to make a positive guess in order to get more rare emotions results in far more false positives than true positives, suggesting the classifier cannot properly generalize on the small amount of data.

Table 5. Detail per-emotion results for SVMBinary submission.

Submission	TP	FP	FN	Precision	Recall	F1
Abuse	0	0	5	0	0	0
Anger	1	2	25	0.33	0.04	0.07
Blame	3	4	42	0.43	0.07	0.12
Fear	0	0	13	0	0	0
Forgiveness	0	1	8	0	0	0
Guilt	50	64	67	0.44	0.43	0.43
Happiness_Peacefulness	1	1	15	0.50	0.06	0.11
Hopefulness	1	6	37	0.14	0.03	0.04
Hopelessness	122	97	107	0.56	0.53	0.54
Information	40	83	64	0.33	0.38	0.35
Instructions	241	168	141	0.59	0.63	0.61
Love	136	65	65	0.68	0.68	0.68
Pride	0	0	9	0	0	0
Sorrow	0	4	34	0	0	0
Thankfulness	26	14	19	0.65	0.58	0.61

In order to determine the most common mistakes, we compared the gold and system output using Cohen's Kappa, commonly used to calculate inter-annotator agreement. The largest source of confusion was INFORMATION being confused for INSTRUCTIONS (= 0.079) and vice versa (0.075). These sentences do tend to look very similar, especially at a lexical level. For instance, most addresses (which were anonymized in the data) occur in either an INSTRUCTIONS or INFORMATION sentence, as the author is either calling the reader's attention to something happening at a particular address (which would be INFORMATION) or instructing them to do something in regards to that address (INSTRUCTIONS). Other common confusions were HOPELESSNESS for GUILT (0.067), LOVE for GUILT (0.064), SORROW for GUILT (0.059), and GUILT for HOPELESSNESS (0.052). We believe a significant source of confusion was the overlapping nature of these emotional contexts, as GUILT was often found in other emotion sentences.

One important aspect of emotion detection not integrated into our approach is a direct modeling of negation, hedging, and other modalities. The lexical approaches presented in this paper do not directly capture such linguisitic phenomena. Many of the features will capture negation and other modifiers: bag-of-words features capture their raw lexical forms, while topic and similarity features are influenced by the full text of the sentence, including not only modifiers but also word choice, which tends to differ under negation. We did attempt to include some heuristic features to recognize negation and hedging, but these did not have sufficient recall to prove useful. We therefore leave more in-depth modeling of emotion negation and modality to future work.

Conclusion

We have presented our approach to the 2011 i2b2 Shared Task on Sentiment Analysis of Suicide Notes. We have described supervised multi-labeling approaches for detecting emotions from real suicide notes using a hybrid strategy of statistical lexicon extraction and sentence similarity metrics. The submission achieved good results in the task, well out-performing the average entry. The submission comes close to the inter-annotator agreement (0.546), meaning it achieves near-human performance on the task.

The only emotion-specific knowledge used by the system is WordNet Affect, which contains a relatively small number of emotion words. We believe the lack of emotion-specific resources is the primary bottleneck to our performance, and thus plan to incorporate more such resources in future work.

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. John P. Pestian, Pawel Matykiewicz, Michelle Linn-Gust, et al. Sentiment Analysis of Suicide Notes: A Shared Task. *Biomedical Informatics Insights*. 2012;5 (Suppl. 1):3–16.
2. Saif M. Mohammad, Peter D. Turney. Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In: *Proceedings of the NAACL HLT* 2010 *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. 2010:26–34.
3. Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, Manfred Stede. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*. 2011;37(2):267–307.
4. Clark Elliot. *The Affective Reasoner: A process model of emotions in a multi-agent system*. PhD thesis, Institute for the Learning Sciences, Northwestern University, 1992.
5. Carlo Strapparava, Alessandro Valitutti. WordNet-Affect: an affective extension of WordNet. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. 2004:1083–6.
6. Robert Plutchik. A general psychoevolutionary theory of emotion. In: *Emotion: Theory, Research, and Experience*. 1980;1:3–33.
7. Carlo Strapparava, Rada Mihalcea. Learning to Identify Emotions in Text. In *Proceedings of the ACM Conference on Applied Computing*. 2008.
8. Kirk Roberts, Sanda M. Harabagiu. A flexible framework for deriving assertions from electronic medical records. *Journal of the American Medical Informatics Association*. 2011;18(5):568–73.
9. Robert Parker, David Graff, Junbo Kong, Ke Chen, Kazuaki Maeda. English Gigaword Fourth Edition. *The LDC Corpus Catalog*., LDC2009T13, 2009.
10. David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*. 2003;3:993–1022.
11. Andrew Kachites McCallum. Mallet: A machine learning for language toolkit, 2002.
12. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
13. Thorsten Joachims. Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods—Support Vector Learning*. 1999:41–56.
14. Pavel Pudil, Jana Novovicov a, Josef Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*. 1994;15:1119–25.