1-30-2012

Emotion Detection In Suicide Notes Using Maximum Entropy Classification

Richard H. Wicentowski
Swarthmore College, richardw@cs.swarthmore.edu

M. R. Sydes

Biomedical Informatics Insights

ORIGINAL RESEARCH

Emotion Detection in Suicide Notes using Maximum Entropy Classification

Richard Wicentowski[1] and Matthew R. Sydes[2]

[1]Swarthmore College, Computer Science Department, Swarthmore, PA, USA. [2]Medical Research Council Clinical Trials Unit, London, UK. Corresponding author email: richardw@cs.swarthmore.edu

Abstract: An ensemble of supervised maximum entropy classifiers can accurately detect and identify sentiments expressed in suicide notes. Using lexical and syntactic features extracted from a training set of externally annotated suicide notes, we trained separate classifiers for each of fifteen pre-specified emotions. This formed part of the 2011 i2b2 NLP Shared Task, Track 2. The precision and recall of these classifiers related strongly with the number of occurrences of each emotion in the training data. Evaluating on previously unseen test data, our best system achieved an F_1 score of 0.534.

Keywords: natural language processing, text analysis, emotion classification, suicide notes

Biomedical Informatics Insights 2012:5 (Suppl. 1) 51–60

doi: 10.4137/BII.S8972

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

Introduction

The goal of this study was to identify emotional content expressed in suicide notes. This was undertaken as part of the 2011 i2b2 NLP Shared Task,[1] Track 2. The organizers of the shared task created a fixed inventory of fifteen emotions (Table 1).

An external team first collated the notes and subsequently anonymized them so that no personal or identifiable data remained; names and addresses were substituted with a small selection of alternatives. Subsequently, volunteers who had a strong emotional connection to someone who had committed suicide provided a sentence-level annotation of the suicide notes such that every sentence was either assigned one or more emotions or, more often, was left unlabeled. Sentences were labeled with a particular emotion only when a sufficient number of annotators agreed upon the annotation; hence, an unlabeled sentence does not necessarily indicate a complete agreement that these emotions were absent from the sentence nor does a labeled sentence necessarily indicate incontrovertible evidence that an emotion was present in the sentence.

Participants in the shared task were provided with 600 annotated notes as training data. The task organizers asked participants to optimize their systems according to F_1 score: the harmonic mean of precision and recall.

Two of our systems attempted to achieve maximal F_1 score, as instructed, by balancing precision and recall. Our third system attempted to achieve higher precision at the expense of lower recall. In the sections below, we describe our activities with the annotated training data before focusing on our performance with the test data.

Methods

Overview

In the task presented here, each sentence of a suicide note was classified as exhibiting zero or more emotions, annotated from a pre-defined set of 15 "emotions". The emotions are shown in Table 1 and include two non-emotional labels: "information" and "instructions". In addition, the label we refer to as "happiness" included both "happiness" and "peacefulness". Many well-studied natural language processing (NLP) classification tasks (part-of-speech tagging, word-sense disambiguation, spelling correction, named entity detection, sentiment analysis and spam

Table 1. List of the annotated emotions in the dataset, as well as the frequency of occurrence of each of the emotions in the training set and the test set and the annotation guidelines. The log of the test/train ratio illustrates whether the training set represented the test set. Positive numbers indicate an overrepresentation of the emotion in the test set, zero indicates equal representation and negative indicates underrepresentation.

Emotion	Training		Test		$\log_2$ ratio Test:Train	Annotator guidelines
	Freq	Pct	Freq	Pct		
Instructions	820	17.7%	382	18.3%	0.05	Giving directions on what to do next ...
Hopelessness	455	9.8%	229	12.2%	0.16	Feels hopeless ...
Love	296	6.4%	201	9.6%	0.59	Feels love for someone ...
Information	295	6.4%	104	5.0%	–0.35	Giving practical information where things stand ...
Guilt	208	4.5%	117	5.6%	0.32	Feels guilt ...
Blame	107	2.3%	45	2.2%	–0.10	Is blaming someone ...
Thankfulness	94	2.0%	45	2.2%	0.09	Is thanking someone ...
Anger	69	1.5%	26	1.3%	–0.26	Is angry with someone ...
Sorrow	51	1.1%	34	1.6%	0.57	Feels sorrow ...
Hopefulness	47	1.0%	38	1.8%	0.84	Has hope for future ...
Happiness	25	0.5%	16	0.8%	0.51	Is feeling happy or peaceful ...
Fear	25	0.5%	13	0.6%	0.21	Is afraid of something ...
Pride	15	0.3%	9	0.4%	0.41	Feels pride ...
Abuse	9	0.2%	5	0.2%	0.30	Was abused verbally, physically, mentally ...
Forgiveness	6	0.1%	8	0.4%	1.57	Is forgiving someone ...
Total sentences	4633	–	2086	–	–	
Labeled sentences	2173	46.9%	1098	52.6%	0.17	
Labels assigned	2522	3.6%	1272	4.0%	0.16	
Unannotated notes	5	0.8%	1	0.3%	–1.32	

filtering, to name a few) also draw their labels from a pre-defined label inventory. However, the classifier in those tasks must assign exactly one label to each information unit whereas here we must provide zero or more labels. While this difference is worth noting, it does not preclude us from investigating off-the-shelf implementations of classifiers used for other NLP tasks. It simply requires recasting into the more familiar problem where the classifier assigns exactly one label to each sentence. We accomplish this by training 15 different classifiers, one for each emotion.

Each classifier performs a binary labeling for that emotion (estimating it as present or absent). We then transform the output of each classifier so that the labeling from classifier *e* is either the set $\{e\}$, if the emotion is present, or the null set, if the emotion is not present. Doing so allows us to form the final classification for each sentence by taking the union of the sets assigned by each individual classifier. If the union of these sets for a particular sentence is the null set, we say the sentence is *unlabeled*.

In Table 1, we show the number and percentage of sentences that were annotated with each label ("Labeled sentences"). Since each sentence could have been labeled with up to 15 labels, we may wish to consider the number of total labels assigned ("Labels assigned"). This value exceeds the number of labeled sentences since 14% of labeled sentences had multiple labels. There were 4,633 sentences in the training set, each of which could have received up to 15 different labels. Of the potential 69,495 labels that could have been assigned, 2,522 (3.63%) were assigned.

Maximum entropy classifier and features

For each emotion, we trained a maximum entropy classifier[2] using only the training data supplied as part of the shared task, then we applied the trained classifier to the test data. Maximum entropy classifiers have been widely used in NLP classification tasks, for example in part-of-speech tagging[3] and in named-entity recognition.[4] In this work, we made use of the freely available Stanford Classifier.[5] In addition to using words (unigrams) as features, we experimented with a wide variety of additional classifier options, preprocessing options, and feature types:

Classifier Options

- **Sigma**: The classifier uses sigma as a parameter to specify the strength of the prior. By default, this value is 1.0. Values less than 1.0 indicate a stronger prior. We experimented with values between 0.25 and 2.0. All results presented below use a sigma value of 0.5, which was set based on evidence from cross-validation.
- **Word shape**: One of the features that can be automatically generated by the Stanford Classifier is *word shape*. This feature is used to conflate words that "look" the same. For example, all 4-letter words beginning with a capital letter could both be represented as "Xxxx" and all two-digit monetary amounts could be represented as "$dd". Based on the recommendation found in the classifier's documentation, we used the "chris4" shape algorithm in our experiments. Details of the "chris4" algorithm can be found in the source code for the classifier.

Preprocessing Options

- **Contraction normalization**: A large number of common contractions were present in the dataset with different spellings. For example, the word "can't" was present as "can't", "cant", "can ' t", "ca n't" and "cann*t". The asterisk was a common substitute for an apostrophe in many cases found in the training data (eg, "doesn*t", "don*t", "who*s", etc). We developed rules to cover a range of possible misspellings for common contractions following the misspelled examples we found in the training data in order to accommodate misspellings which might be expected in the test data.
- **Spelling correction:** For any word not found in a large, clean wordlist, our spelling correction algorithm chose the most frequently occurring alternative as ranked by the unigram counts derived from Google's index.[6] Alternatives were generated by applying single character insertions, deletions, substitutions and transpositions. We included "space" as a valid insertion character given the large number of tokens in the dataset that were the result of two words joined together without a space (eg, "thepapers", "knowit"). We added special rules to handle a few common, and uncommon, errors (eg, "ys" → "ies", "x" → "ct") after examining the results of this algorithm on the training data.

Feature Types

- **Morphological features:** We experimented with using both character n-grams and word-stemming algorithms to complement our feature set. Character n-grams included only prefix and suffix n-grams ranging from 2-grams to 6-grams. These character n-grams were used to capture simple morphological processes. We also used the NLTK[7] implementation of the Snowball stemmer[*] to conflate morphologically related words.
- **Word bigrams and trigrams:** In addition to unigram features, we also used word bigrams and word trigrams as features. More specifically, token bigrams and trigrams, as we did not distinguish between words and punctuation in creating these n-grams.
- **Dependency relations:** Although the use of dependency relations is not new in this field,[8] their use is far less common than the other features described above. We were motivated to include dependency relations based on our understanding of the guidelines provided to annotators (Table 1). For example, the guidelines for "forgiveness" state that the author of the note should be forgiving someone, not asking for forgiveness. Therefore, it is relevant to identify dependencies such as "nsubj(i, forgive)" as positive examples of the forgiveness label, whereas "dobj(forgive, me)" would be a negative example. We used the Stanford Parser[9] to generate these dependencies, extracting "collapsed dependencies". Collapsed dependencies combine dependency relations so that the resulting relations only contain content words, omitting prepositions and conjunctions, for example.[†] Note that when spelling correction and contraction normalization are used as features, the parser receives this corrected and normalized text as input; otherwise, the original notes are used.
- **Variable dependency relations:** We also experimented with substituting individual words with variables in these dependency relations, expecting to conflate uncommon patterns in order to decrease data sparseness. For example, this method would conflate "dobj(love, him)", "dobj(love, her)" and "dobj(love, Jane") into "dobj(love, *x*)". We did not perform any entity detection on the arguments in the dependency relations, so relations such as "dobj(love, life)" were also conflated into "dobj(love, *x*)".

The Results section presents the performance of these features using five-fold cross-validation on the training data. In explaining the features used in this presentation, we use the key presented in Table 2.

Heuristics

- **Confidence tuning**: In a binary classification setting, the Stanford Classifier yields results ranging from 0.0 (very strong likelihood sentence is unlabeled) to 1.0 (very strong likelihood sentence is labeled), using 0.5 as the dichotomous split point on whether or not to label a sentence. Using cross-validation, we were able to fine-tune this split point to optimize F_1 (always by increasing recall at the expense of precision). In our first submitted system, we fine-tuned a single split point value for use with all emotion classifiers. In the other two systems, we individualized the split point for each emotion classifier.
- **Minimal annotation**: Inspection of the training data revealed that there were very few notes that had no annotations at all, although most contained sentences that had no annotations. Therefore, we ensured that in each of our submitted systems, every note had at least one annotated sentence. If the classifiers yielded a note with no annotations, we found the sentence-emotion pair with the highest confidence and then labeled that sentence with the emotion.
- **Skipping hard-to-predict emotions**: On cross-validation (see Results for details), our F_1 performance was hurt by low precision and low recall on the emotions which occurred infrequently in the training data (see Discussion for comment). Therefore, in each of our submitted systems we did not include the results of the classifiers associated with the nine lower frequency emotions. For these systems, we estimated only the higher frequency emotions which had performed well in cross-validation experiments: "instructions", "hopelessness", "love", "information", "guilt" and "thankfulness".

[*] See `http://goo.gl/Zrybe`.

[†] See the Dependencies Manual accompanying the Stanford Parser for more details.

Table 2. Key to the features used in the classifiers. For example, the notation "W1–1/C0–0/L-/M-/D-/O-" indicates a feature set using only unigrams as features and "W1–2/C0–0/L+/M+/Dd/Oe" indicates a feature set using word unigrams and bigrams, spelling correction, stemming, dependency relations and a split point optimized for each emotion. See the Methods section for a complete description of these features.

Key	Description
W*x*-*y*	Word *n*-grams, $n = x \ldots y$. eg, W1-1 = Unigrams only; W1-2 = Unigrams and bigrams; $1 < x \leq y$
C*x*-*y*	Prefix/suffix character *n*-grams, $n = x \ldots y$; n.b. C0-0 indicates omission of this feature; $0 < x \leq$ y
L+/L–	Uses spelling correction and contraction normalization (L+) or not (L–)
M+/M–	Uses stemming (M+) or does not (M–)
Dd/Dv/D–	Uses dependency relations only (Dd), with variable dependency relations (Dv) or neither (D–)
Oa/Oe/O–	Optimizes a split point for all classifiers (Oa), for each emotion (Oe) or not at all (O–)

- **Emotionless classification**: Utilizing the same feature set described above, we trained a binary classifier to identify sentences that were labeled with any emotion, regardless of the actual emotion with which they were annotated. We call this the *emotionless* classifier, since it differentiates between sentences "with emotion" and "emotionless" sentences. We combined the output of the emotionless classifier with each of the individual emotion classifiers described above. Given a sentence that was identified as having some emotion by the emotionless classifier, we slightly increased the confidence of each of the individual emotion classifiers for that sentence. We used cross-validation to fine tune the increase amount.
- **Memorize labels**: The training data contained sentences that were identical to sentences in the test data. When using this heuristic, we copied the label from the identical training data sentence onto the matching test sentence.
- **Classifier combinations**: We trained individual classifiers based on each of the 48 combinations of the features listed above. We then attempted to use logistic regression to identify small sets of combination-classifiers that were most orthogonal; we combined these classifiers to try to improve on the performance of the standalone classifiers. The combination was done by linearly combining the confidence values of each classifier on a sentence-by-sentence basis. Each classifier was equally weighted in this linear combination, so the final confidence value was always equal to the mean of the confidence scores of the individual classifiers. This method failed to achieve adequate results to be included in any of our submissions.

Systems used for submission

As part of the evaluation exercise, participants were allowed to submit the results from up to three classifiers, with the understanding that the single best classifier, as determined by F_1 score, would be the system used to provide a ranking of all participants. Below is a description of the three systems that we developed and submitted:

- **System 1**: Our first system, S_1, used `W1-2/C0-0/L+/M+/Dd`, the feature set that performed best on cross-validation, with sigma set to 0.5, confidence tuned to 0.198, skipping hard-to-predict emotions.
- **System 2**: Our second system, S_2, used the same feature set as S_1, with sigma set to 0.5 and skipping hard-to-predict emotions. But this system used split points tuned specifically for each emotion and ensured at least one annotation per note. Sentences that were labeled by an emotionless classifier (same features, sigma = 0.5) had the confidence of their labelings increased by 0.05 for all emotions. We then applied the *Memorize labels* heuristic.
- **System 3**: Our final system, S_3, was identical to S_2 except that the split point was the untuned value of 0.5.

Results

In this section, we present two sets of results. We detail the performance of each feature used by our classifiers using 5-fold cross-validation methods on the training data. Then, we present the performance of our three submitted systems on the test data and compare that to system performance on cross-validated training data.

Cross-validated results and feature selection

Confidence tuning has a pronounced effect on the results regardless of the feature set used; therefore, we begin by presenting results using our default 0.5 split point and the tuned split point found using cross-validation on two representative feature sets. Table 3 sets out the true and false positives (TP and FP) and false negatives (FN) for these models, together with the precision (P), recall (R) and F_1 score. True negatives are not shown in any of the models as they do not impact the F_1 score. Comparing the first and third rows, we can see how introducing bigrams, spelling correction, stemming and dependencies greatly reduced the false positives but at the cost of fewer true positives and more false negatives. The impact on F_1 was still beneficial. The second and fourth rows show how fine-tuning the split point could further improve the functioning of these classifiers.

Given the across-the-board improvement due to split point optimization, Table 4 presents the results of using each of the features individually (relative to using only using word unigrams) using only this optimization. Table 5 shows the performance of the single best classifier using cross-validation. In Table 6, we present the mean change in precision, recall and F_1 observed when each feature was added to our classifier, averaging across all permutations of the other features. Since the overall F_1 score is the mean across all possible other sets of features, one can view the values in Table 6 as an approximation of the additive amount of increase in F_1 that can be obtained by adding each feature to the standard classifier. Table 7 presents the performance for each of our three submitted systems on both the training and test data.

Discussion

All of the systems we submitted achieved results in line with the performance we obtained using cross-validation on the training set. Since, for testing, we could train on the full data set (instead of 80% as in the 5-fold cross-validation setup), we expected a small improvement in our F_1 score on the test data; this was realized for our first (S_1) and third (S_3) systems, but not in our second (S_2) system. One possibility is that the individualized optimization of split points on a per-emotion basis used in S_2 led to overtraining.

We trained our classifiers to function well specifically in terms of the F_1 score as this was the scoring goal of the i2b2 competition. The F_1 score is the harmonic mean of precision and recall. In other settings, precision is known as sensitivity; it captures the proportion of annotated emotions that the classifier detects. Recall also uses the correct positive annotations, focusing on the proportion of positive estimates which were correct; this is known in many other settings, notably health research, as positive predictive value. In those same settings, precision (sensitivity) is supplemented, not by recall but by specificity which focuses on the non-annotation of a label. Here, specificity would be the proportion of sentences that should not be annotated with a given emotional label correctly not having that label applied. The use of F_1 score as a summary in this setting completely ignores the true negatives (ie, those that were correctly not labeled with a given emotion). We believe that these true negatives were actually important. With this exercise we were effectively running a series of yes/no annotation exercises on the same dataset. The clear majority of sentences were not labelled with each given emotion: annotated sentences for each given emotion were quite rare in both the training and test datasets. The correct answer was to not label with each particular emotion in most instances. It is interesting to note that the specificity was extremely high (and, therefore, very encouraging) for all of the classifiers that we developed.

Table 3. For every feature set, fine tuning a single split point for all 15 emotion classifiers yielded a decrease in precision, an increase in recall and a marked increase in F_1 in the cross-validated training set. Two representative feature sets are illustrated above without optimization (split point = 0.500) and with optimization.

Features	Split point	TP	FP	FN	P	R	F_1
W1-1/C0-0/L-/M-/D-	0.500	935	731	1587	56.12	37.07	44.65
	0.342	1113	1229	1409	47.52	44.13	45.76
W1-2/C0-0/L+/M+/Dd	0.500	892	408	1630	68.62	35.37	46.68
	0.170	1228	980	1294	55.62	48.69	51.92

Table 4. Relative performance difference in the cross-validated training set of including a single new feature relative to the baseline system that used only unigrams, evaluated using the "Oa" optimization, sorted by F_1. See Table 2 for an explanation of the notation used to describe the features.

Comment	Features	TP	FP	FN	P	R	F_1
D- to Dv	W1–1/C0–0/L–/M–/**Dv**	1175	1590	1347	42.50	46.59	44.45
L– to L+	W1–1/C0–0/**L+**/M–/D–	1054	1059	1468	49.88	41.79	45.48
Baseline	W1–1/C0–0/L–/M–/D–	1113	1229	1409	47.52	44.13	45.76
C0–0 to C2–6	W1–1/**C2–6**/L–/M–/D–	1176	1378	1346	46.05	46.63	46.34
D- to Dd	W1–1/C0–0/L–/M–/**Dd**	1170	1264	1352	48.07	46.39	47.22
M– to M+	W1–1/C0–0/L–/**M+**/D–	1126	1121	1396	50.11	44.65	47.22
W1–1 to W1–2	**W1–2**/C0–0/L–/M–/D–	1168	1175	1354	49.85	46.31	48.02

The annotated datasets were the gold standard in these exercises and reflected the time and effort of many people. Each sentence was annotated by at least three annotators, assigning annotations to sentences only when two or more annotators agreed. However, there were many instances in both the training and test dataset where we found that emotions were inappropriately applied or omitted by the annotators, as detected by our classifiers. For example, the guidelines state that a sentence should be annotated with "forgiveness" if the author is forgiving someone, not if the author is asking for forgiveness. Yet, the sentence "Forgive me for this rash act but I alone did it." was wrongly annotated with "forgiveness" in the training set by the annotators.

In addition, there were instances where exactly same sentence did not attract the same emotion from the annotators. In the training data, some sentences appeared multiple times across notes. For example, the sentence "I love you." appeared in 7 training sentences: 5 times annotated with "love" and 2 left unannotated. Our *Memorize Labels* heuristic relied on the fact that sentences appearing multiple times across the test and training data would be labeled consistently. However, there were at least 15 sentences in the test data that appeared in the training data with different annotations. Although the particular context of the sentence could affect the labeling, there were two pairs of notes appearing in both the test and training data that were nearly

Table 5. Performance of the best single classifier, W1–2/C0–0/L+/M+/Dd, using 5-fold cross-validation on the training data, with a single optimized split point = 0.198 and sigma = 0.5, sorted by F_1 compared with the performance of the same classifier when not providing labels for emotions with small amounts of training data. The latter is equivalent to system S_1.

Emotion	Classify all labels						Skipping hard-to-predict labels					
	TP	FP	FN	P	R	F_1	TP	FP	FN	P	R	F_1
Love	203	125	93	61.89	68.58	65.06	203	125	93	61.89	68.58	65.06
Instructions	571	384	249	59.79	69.63	64.34	571	384	249	59.79	69.63	64.34
Thankfulness	44	16	50	73.33	46.81	57.14	44	16	50	73.33	46.81	57.14
Hopelessness	244	212	211	53.51	53.63	53.57	244	212	211	53.51	53.63	53.57
Guilt	87	98	121	47.03	41.83	44.27	87	98	121	47.03	41.83	44.27
Information	103	134	192	43.46	34.92	38.72	103	134	192	43.46	34.92	38.72
Blame	5	18	102	21.74	4.67	7.69	0	0	107	–	0.00	–
Hopefulness	1	6	46	14.29	2.13	3.70	0	0	47	–	0.00	–
Abuse	0	1	9	0.00	0.00	0.00	0	0	9	–	0.00	–
Fear	0	0	25	–	0.00	–	0	0	25	–	0.00	–
Forgiveness	0	0	6	–	0.00	–	0	0	6	–	0.00	–
Anger	0	4	69	0.00	0.00	0.00	0	0	69	–	0.00	–
Pride	0	0	15	–	0.00	–	0	0	15	–	0.00	–
Happiness	0	2	25	0.00	0.00	0.00	0	0	25	0.00	0.00	0.00
Sorrow	0	6	51	0.00	0.00	0.00	0	0	51	–	0.00	–
Total	1258	1006	1264	55.57	49.88	52.57	1252	969	1270	56.37	49.64	52.79

Table 6. Average relative performance difference on cross-validated training data between classifiers trained with and without the feature listed, sorted from least effective to most effective. Positive values indicate that, on average, the classifier improved with the addition of that feature.

Feature added	Change in precision		Change in recall		Change in F_1 score	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
Spelling correction (L+)	–0.042	1.652	0.164	1.704	0.069	0.405
Character n-grams (C2–6)	–0.511	2.664	1.313	2.142	0.461	1.495
Stemming (M+)	0.628	1.663	0.861	1.936	0.749	0.710
Variable dependencies (Dv)	–1.521	2.699	2.887	1.837	0.751	1.196
Dependencies (Dd)	1.310	1.801	2.434	1.578	1.920	0.480
Bigrams (W1–2)	4.199	2.573	0.844	2.078	2.473	1.069

Note: All classifiers were evaluated using the "Oa" optimization.

identical. These pairs of notes were the result of a single author writing two notes to different people. One note was in the training data, one in the test data. Even here we find inconsistencies. In one pair, 17 annotations were made to the note found in the training data, yet only 7 annotations were made to the note in the test data. For example, "I want to wear my red and black dress [at my funeral]" was annotated

Table 7. Performance of the three classifiers used for official submissions. Sorted by F_1 on the test data, results are presented for both using 5-fold cross-validation on the training data and on the test data. Note that the Memorize labels heuristic, used in S_2 and S_3 was not applied on cross-validation.

Emotion	Cross-validation on training data						Test data					
	TP	FP	FN	P	R	F_1	TP	FP	FN	P	R	F_1
Classifier S_1												
Love	203	125	93	61.89	68.58	65.06	135	63	66	68.18	67.16	67.67
Thankfulness	44	16	50	73.33	46.81	57.14	30	17	15	63.83	66.67	65.22
Instructions	571	384	249	59.79	69.63	64.34	253	158	129	61.56	66.23	63.81
Hopelessness	244	212	211	53.51	53.63	53.37	138	114	91	54.76	60.26	57.38
Guilt	87	98	121	47.03	41.83	44.27	39	36	78	52.00	33.33	40.62
Information	103	134	192	43.46	34.92	38.72	36	71	68	33.64	34.62	34.12
Skipped	0	0	354	–	0.00	–	0	0	194	–	0.00	–
Total	1252	969	1270	56.37	49.64	**52.79**	631	459	641	57.89	49.61	**53.43**
Classifier S_2												
Love	198	111	98	64.08	66.89	65.45	130	56	71	69.89	64.68	67.18
Instructions	523	281	297	65.05	63.78	64.41	224	112	158	66.67	58.64	62.40
Thankfulness	50	28	44	64.10	53.19	58.14	33	28	12	54.10	73.33	62.26
Hopelessness	250	216	205	53.65	54.95	54.29	137	117	92	53.94	59.83	56.73
Guilt	98	136	110	41.88	47.12	44.34	36	34	81	51.43	30.77	38.50
Information	109	149	186	42.25	36.95	39.42	38	74	66	33.93	36.54	35.19
Skipped	0	0	354	–	0.00	–	1	2	193	0.33	0.01	0.01
Total	1228	921	1294	57.14	48.69	**52.58**	599	423	673	58.61	47.09	**52.22**
Classifier S_3												
Love	164	63	132	72.25	55.41	62.72	110	27	91	80.29	54.73	65.09
Instructions	441	178	379	71.24	53.78	61.29	182	68	200	72.80	47.64	57.59
Thankfulness	29	9	65	76.32	30.85	43.94	22	12	23	64.71	48.89	55.70
Hopelessness	186	92	269	66.91	40.88	50.75	100	49	129	67.11	43.67	52.91
Information	70	52	225	57.38	23.73	33.57	24	29	80	45.28	23.08	30.57
Guilt	63	51	145	55.26	30.29	39.13	21	18	96	53.85	17.95	26.92
Skipped	0	0	354	0.00	0.00	0.00	1	2	193	0.33	0.01	0.01
Total	953	445	1569	68.17	37.79	**48.62**	460	205	812	69.17	36.16	**47.50**

as "instructions" in the training data and was left unannotated in the test data.

The suicide notes provided in the data set were transcriptions of hand-written notes. These notes contained many spelling errors and tokenization inconsistencies. It was unclear where these errors originate, but we suspect some are genuine errors from the author and others were transcription errors in preparing the data sets eg, "3333 Burnet Ave" is sometimes "3333 Burent Ave"; other such errors could have been introduced. Our spelling correction algorithm fixed minor errors (eg, "sufering" → "suffering"; "attemp"→ "attempt"; "beond" → "beyond") but failed to correct more complex errors that involved more than a single substitution, transposition, insertion or deletion. For example, "capsuls" was amended by our system as "capsule" instead of "capsules". Additionally, many spelling errors involved words (often spelled somewhat phonetically) that could not be corrected at all by this simple method: "hemorige" ("hemorrhage"), "disponded" ("despondent), "rearenge" ("rearrange"). Some mispelled words were "corrected" erroneously. There were also instances where the algorithm corrected words that weren't incorrect. For example, changing the abbreviations "appt" (appointment) → "apt" and "tel" (telephone) → "tell". Some of these errors may have been addressed more accurately by using an n-gram language model to estimate the best possible correction.[10] For example, the phrase "get *bettery* charged" should have been corrected as "get *battery* charged" but was actually changed to "get *better* charged".

Introducing dependency relations (Dd) into the model provided a large boost to the overall system performance: the second largest increase in precision, the second largest increase in recall, and the second largest increase in overall F_1. The variable dependencies feature (Dv) conflated dependencies such as "dobj(blame, John)" and "dobj(blame, Mary)" into "dobj(blame, *x*)". We expected that this could help with data sparsity issues and we demonstrated large gains in recall when using this feature. Unfortunately, this also introduced a large drop in precision. These variable dependencies introduced much noise, possibly because we were not differentiating between the arguments of the dependencies we were conflating. For example, the annotation guidelines for the *blame* label state that the author of the note should have been blaming *someone*. However, conflating "dobj(blame, money)" and "dobj(blame, weight)" with "dobj(blame, Mary)" is unhelpful given these guidelines. Had we used entity detection to determine that both *Mary* and *John* were people, we could have constructed "dobj(blame, PERSON)", separating those examples from "dobj(blame, THING)" and potentially improving on our performance.

As one might expect, each classifier did particularly poorly on the emotions that occurred infrequently in the training data. Indeed, the performance of the classifiers for these emotions was so poor that we had better results simply ignoring these emotions rather than include them in our final labeling. Improving our performance on these emotions should be the focus of the continuing development of this work; we suspect that additional training data would have aided.

Our attempts to use classifier combinations were only partially successful. We have demonstrated that introducing the emotionless classifier to boost the confidence of our labelings provided a large increase in recall; however, this yielded a large decrease in precision, with the F_1 score remaining largely unchanged. We also explored using logistic regression to select a panel of orthogonal classifiers with combinations of features that might better balance precision and recall. The efforts to select small panels from 48 combinations of features using regression models were not feasible in the time available to us but may warrant further investigation. An exhaustive evaluation of all pairs and triples of classifier combinations found that no combination of two or three classifiers outperformed the best standalone classifier.

In developing our classifiers, we tried to consider the practical applications of the findings from this exercise.[11] The loss of any life is sad, and the early termination of one's own life particularly so. There is no doubt in our minds that the sentiments expressed in the suicide notes must have been present prior to the actual time of suicide. We consider that there may have been previous efforts to express these emotions to other people. We anticipate that one might consider employing an automatic detection algorithm on social networking platforms. This could review posts and activate access to support networks. However, only systems with very high precision would be of any practical value: high precision is more important than high recall because we would not wish to propose interventions unless we were extremely confident in our predictions.

Towards that goal, our final system, S_3, attempted to achieve high precision at the expense of recall, while minimizing the impact on F_1 score. Potentially, we could have looked to further improve precision with detrimental effects on recall and F_1 score, though it is encouraging that we achieved high precision while maintaining an F_1 score similar to the mean of all systems submitted to this shared task. Further research might be required to consider whether the sentiments expressed on suicide notes are truly expressed previously. Further information on the age, gender and physical and psychiatric health of these people may be of value.

Previous work in suicide note authorship detection used structural and grammatical features such as the number of paragraphs in the note, the number of misspellings, and the depth of parse tree.[12] Our intuition was that these features would not have been useful here, though we corrected spelling errors and used dependency relations from a parser. Structural and grammatical features should be investigated further.

Conclusion

We have shown that it is possible to construct classifiers that can perform well on this task and we have shown that dependency relations can be used effectively as features in these classifiers. We presented a classifier that performs at high levels of precision in order to be useful as a mechanism to propose intervention. We are confident that further improvements could be achieved using off-the-shelf components as done here.

Acknowledgements

We acknowledge the efforts of the i2b2 organizers and all of the brave efforts of the volunteers who kindly annotated the extensive and sensitive dataset.

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 JP, Matykiewicz P, Linn-Gust M, et al. Sentiment analysis of suicide notes: A shared task. In *Biomedical Informatics Insights*. 2012;5 (Suppl. 1):3–16.
2. Nigam K, Lafferty J, McCallum A. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*. 1999;61–7.
3. Ratnaparkhi A. A maximum entropy model for part-of-speech tagging. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing*. 1996.
4. Chieu HL, Ng HT. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL*-2003. 2003;160–3.
5. Manning CD, Klein D. Optimization, maxent models, and conditional estimation without magic. In *HLT-NAACL Tutorial*. 2003.
6. Brants T, Franz A. Web 1T 5-gram, ver. 1. LDC2006T13, Linguistic Data Consortium, Philadelphia, 2006.
7. Bird S, Loper E, Klein E. O'Reilly Media Inc. *Natural Language Processing with Python*. 2009.
8. Nastase V. Unsupervised all-words word sense disambiguation with grammatical dependenices. In *IJCNLP*. 2008;757–62.
9. de Marneffe MC, MacCartney B, Manning CD. Generating typed dependency parses from phrase structure parses. In *LREC* 2006. 2006.
10. Brill E, Moore RC. An improved error model for noisy channel spelling correction. In *Proceedings of the* 38*th Annual Meeting of the Association for Computational Linguistics*. 2000;286–93.
11. Foster T. Suicide note themes and suicide prevention. *International journal of psychiatry in medicine*. 2003;33(4):323–31.
12. Pestian J, Nasrallah H, Matykiewicz P, Bennett A, Leenaars A. Suicide note classification using natural language processing: A content analysis. *Biomedical Informatics Insights*. 2010;2010(3):19–28.