Biomedical Informatics Insights

ORIGINAL RESEARCH

A Naïve Bayes Approach to Classifying Topics in Suicide Notes

Irena Spasić[1], Pete Burnap[1], Mark Greenwood[1] and Michael Arribas-Ayllon[2]

[1]School of Computer Science and Informatics, Cardiff University, Cardiff, UK. [2]School of Social Sciences, Cardiff University, Cardiff, UK. Corresponding author email: i.spasic@cs.cardiff.ac.uk

Abstract: The authors present a system developed for the 2011 i2b2 Challenge on Sentiment Classification, whose aim was to automatically classify sentences in suicide notes using a scheme of 15 topics, mostly emotions. The system combines machine learning with a rule-based methodology. The features used to represent a problem were based on lexico–semantic properties of individual words in addition to regular expressions used to represent patterns of word usage across different topics. A naïve Bayes classifier was trained using the features extracted from the training data consisting of 600 manually annotated suicide notes. Classification was then performed using the naïve Bayes classifier as well as a set of pattern–matching rules. The classification performance was evaluated against a manually prepared gold standard consisting of 300 suicide notes, in which 1,091 out of a total of 2,037 sentences were associated with a total of 1,272 annotations. The competing systems were ranked using the micro-averaged F-measure as the primary evaluation metric. Our system achieved the F-measure of 53% (with 55% precision and 52% recall), which was significantly better than the average performance of 48.75% achieved by the 26 participating teams.

Keywords: natural language processing, sentiment analysis, topic classification, naïve Bayes classifier

Biomedical Informatics Insights 2012:5 (Suppl. 1) 87–97

doi: 10.4137/BII.S8945

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

Introduction

The objective of the 2011 i2b2 Challenge in Natural Language Processing for Clinical Data—Track 2 on Sentiment Classification[1] was to evaluate the performance of natural language processing systems in classifying sentences in suicide notes using a scheme of 15 topics, mostly emotions. The organizers provided the classification scheme and a manually annotated training dataset consisting of 600 suicide notes, where each sentence was associated with zero or more topics from the classification scheme. We assembled a set of lexicons and pattern–matching rules to support feature extraction, which complemented a set of public resources including WordNet[2] and four emotive lexicons[3–6] used for the same purpose. The features extracted from the training dataset were used to train a naïve Bayes classifier. The machine learning module of the system was integrated with a rule–based component, which used a set of pattern–matching rules to annotate sentences with associated topics.

System Overview

Figure 1 illustrates the conceptual architecture of the proposed topic classification approach, consisting of five basic steps: linguistic pre-processing, feature extraction based on lexicon and pattern matching, dimensionality reduction based on principal component analysis, naïve Bayes classification and rule-based classification based on pattern matching. Input document were supplied as plain ASCII text files with one sentence per line and each token separated by blank spaces. Each sentence was linguistically pre-processed, including part-of-speech (POS) tagging,[7] lemmatization[8] and spelling correction.[9] The results of such processing were stored in a relational database[10] for easy access and further semantic reasoning. Prior to its distribution, each sentence in the training set was manually annotated with the associated topics. These annotations were stored together with the text data to be used later to train a machine learning classifier. Lexico-semantic properties of individual tokens as well as phrases matching pre-specified lexico-syntactic patterns were aggregated for each sentence in order to map it to its feature vector. All feature vectors were exported into an ARFF (Attribute-Relation File Format) file for use with the Weka machine learning software.[11]

Using Weka, the training ARFF file with feature vectors mapped to topics first underwent the principal component analysis (PCA) to transform the original features into combinations of possibly correlated ones (called principal components), which account for most of the variance in the training data.[12] As a result, the dimensionality of the data space was reduced from 353 original features to 245 principal components. The transformed data was again saved as an ARFF file and used to train a naïve Bayes classifier. In particular, a discriminative multinominal naïve Bayes classifier was chosen from other varieties offered in Weka as it provided the best results during cross-validation. The naïve Bayes model produced was used in the testing

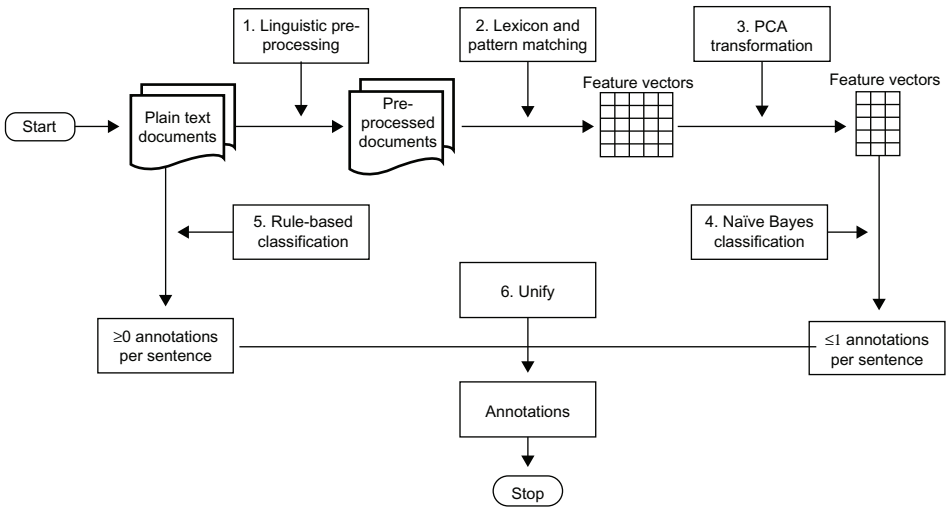

Figure 1. Conceptual architecture of the proposed topic classification approach.

phase to classify sentences from the test dataset. Each test sentence was classified using a single topic with the highest probability (according to the given model) over a set threshold. Different threshold values were used to produce the three test runs submitted, with lower threshold values used to increase the recall and conversely higher threshold values used to increase the precision. A single most probable class was suggested in an attempt to maintain the precision of the method.

Additional classifications were produced using a set of pattern–matching rules extracted manually from the training data. The key criterion in defining these rules was to match multiple true positives while minimizing the number of false positives. The key role of these rules was to improve the recall following the naïve Bayes classification without compromising the precision.

Lexico-Semantic Resources

The majority of features we used combined lexico-semantic properties of individual tokens. These properties were obtained by mapping the tokens to a set of public lexico-semantic resources as well as the internally assembled lexicons. The most general lexical resource used was the WordNet,[2] a large lexical database with content words (ie, nouns, verbs, adjectives and adverbs) grouped into synsets (ie, sets of synonyms), which are mapped to 44 lexical domains and interlinked with 13 sense relations.

The rest of the public resources used were all emotive lexicons, in which words (or phrases) are associated with their positive/negative polarity or mapped to specific emotions. WordNet–Affect[3] maps >900 WordNet synsets to affective concepts they explicitly express. The emotional concepts represented in WordNet-Affect are organized into a hierarchy of 306 nodes, including different polarities represented by positive, negative and neutral emotion classifications within the hierarchy. Similarly, SentiWordNet[4] maps >200K WordNet synsets to three numerical scores, describing how objective, positive and negative the meaning associated with a synset is. The remaining two emotive lexicons used in our approach simply classify words or multi-word expressions as positive or negative, without any further structural or quantitative properties attached to them. We used a lexicon of >6K positive and negative opinion or sentiment words.[5] Similarly, the Macquarie Semantic Orientation Lexicon (MSOL)[6] contains >70K expressions that explicitly convey positive or negative meaning or are implicitly associated with such meaning based on the polarity of words found in their thesaurus paragraph.

Apart from emotive lexicons, which are used to identify explicit mentions of certain emotions[3] or determine the positive/negative polarity of expressions used,[4–6] other types of information may be useful to classify an emotion implied by a specific sentence. For instance, we observed that *Hopelessness* is often associated with health conditions, *Love* typically involves family members, *Anger* is sometimes expressed using the swear words, etc. We modeled semantic categories that may be relevant for emotion classification by assembling a lexicon for each of them. Table 1 lists the types of semantic categories covered with examples taken from the corresponding lexicons. In general, all lexicons were initially seeded by manually analyzing content words frequently found in the corpus. We used the fact that the training data were anonymized using a limited number of personal names to create their lexicon. Similarly, we initially identified references to the family members in the training data and then manually added other related words, eg, if *aunt* was in the lexicon, we would add *uncle* to it as well. Other lexicons were supplemented with data retrieved from the Internet, eg, there are publicly available lists of pejorative words as part of

Table 1. Semantic categories with examples taken from the corresponding lexicons.

Semantic category	Lexicon entries	Total
Family	*baby, child, dad, daughter, husband, sister, uncle, wife, …*	67
Personal names	*Alice, Daniel, Helene, James, Jane, Joseph, Peter, Sandra, …*	57
People	*blonde, boyfriend, guy, lady, mistress, teenager, woman, …*	54
Occupation	*accountant, attorney, consultant, doctor, nurse, officer …*	112
Health	*cancer, disease, incurable, nausea, recovery, arthritis, dizzy, …*	282
Religion	*Antichrist, hell, inferno, Jesus, Lucifer, perdition, Satan, sin, …*	31
Pejorative words	*###, ###, ###, ###, ###, ###, ###, ###, ###, ###, ###, ###, …*	349

swear filters used to remove words deemed offensive from the online content.[13] As for the health terms, we focused on the ones related to suicidal tendencies.[14] The lexicons were further expanded with their synonyms retrieved from the WordNet, eg, the word *grandmother* would be complemented with words such as *grandma, grannie, granny*, etc.

We created no explicit links between these semantic categories and topics from the classification scheme. Instead, any correlations present in the training data were automatically identified as part of the naïve Bayes model created in the training phase. Apart from lexically modeling specific semantic categories, we created 15 other lexicons, one for each topic from the classification scheme, which contained useful lexical clues for the given topic, ie, words describing various semantic types, but all of them related to specific topic (see Table 2). These words were chosen based on the human judgment of their relevance.

Table 2. Topics and examples of their lexical clues.

Topic	Lexical clues	Total
Abuse	*beat, awful, curse, slap, swear, torture, threat,…*	13
Anger	*mean, cruel, hate, lousy, hypocrite, divorce, …*	21
Blame	*blame, cause, drive, fault, kill, responsible, …*	10
Fear	*afraid, brave, coward, dread, fear, panic, terror, …*	33
Forgiveness	*forgive, forgiveness*	2
Guilt	*fail, forgive, guilt, please, regret, understand, …*	26
Happiness_ peacefulness	*God, happy, hope, Lord, heaven, peace, ready, …*	16
Hopefulness	*best, believe, God, good, hope, happy, luck, …*	10
Hopelessness	*agony, numb, pain, sick, sorry, suffer, struggle, …*	79
Information	*account, compartment, debt, payment, receipt, …*	60
Instructions	*ashes, arrange, casket, funeral, phone, service, …*	102
Love	*adore, beloved, darling, deeply, heart, love, kiss, …*	49
Pride	*admire, admirable, pride, proud, prideful, …*	5
Sorrow	*apologize, hurt, heart, please, sad, sorry, …*	10
Thankfulness	*appreciate, grateful, kindness, thank, supportive, …*	45

In addition to manually curated lexicons described previously with words not necessarily originating from the training dataset, we created a lexicon of potentially useful words based on the mutual information between them and the classes considered. Mutual information (MI) is a measure of mutual dependence between two variables, and is calculated as follows given a word w and a class c:

$$\mathrm{MI}(w,c) = \log \frac{\mathrm{P}(w,c)}{\mathrm{P}(w)\mathrm{P}(c)}$$

The class-specific MI values were aggregated as $\mathrm{MI}_{\max}(w) = \Sigma_i \mathrm{P}(c_i)\mathrm{MI}(w,c_i)$ to estimate the global usefulness of each word w for classification purposes.[15] We selected the top 20% of words as potentially most useful classification clues. As a result, a total of 153 words were incorporated into a lexicon of informative words, examples of which include: *friend, tell, spirit, letter, fail, son,* etc. All lexico-semantic resources described thus far were stored in a relational database together with the data to facilitate feature extraction.

Finally, similarly to lexical clues linked to specific topics, we explored pattern–matching rules implemented as regular expressions in Java. To facilitate rule extraction, we used lexical clues with a simple concordance program[16] to analyze false negatives and add new rules iteratively. All rules were defined and selected manually by observing the ratio between true positives and false positives retrieved. Table 3 shows a sample of the pattern-matching rules associated with topics from the classification scheme.

Feature Selection

The given topic classification task is a supervised learning problem, where each sentence is assigned to one or more predefined topics. The first problem associated with this task, or with text categorization in general, is the high dimensionality of feature space, since most often the features chosen are the frequencies of individual words. Such representation suffers from the curse of dimensionality (or Hughes effect),[17] where given a fixed size of the training dataset, the predictive power of a machine learning algorithm reduces as the dimensionality increases. In order to reduce the number of features we applied two strategies: generalization and mutual information.

Table 3. Topics and examples of pattern-matching rules.

Topic	Rules	Examples	Total
Abuse	(drive|drove).{0,1} me.{0,10} (mad|crazy|insane|mind)	Your indifference, …, just plain *drove me out of my mind*.	7
Anger	see .* hell	I will *see you in the hell.*	53
Blame	^(?!.*(never|not|n't|to)) let me down	It looks like John *let me down.*	47
Fear	(not|n't|no).{0,10} courage	I haven*'t the courage* to go on.	10
Forgiveness	forgive (you|him|her|anyone)	Tell him I *forgive him* for all my heart aches and tell him to pray to God for forgiveness.	3
Guilt	I (ask|beg).{0,10} forgiveness	*I beg you forgiveness* for what I have done to myself, to you, the children and all the family.	75
Happiness_ peacefulness	(not|n't) mourn	Please do *not mourn* me, for where I am going, life is beautiful, peaceful; where contentment and happiness reigns.	12
Hopefulness	meet in heaven	I hope someday we will *meet in heaven.*	22
Hopelessness	I ca(n't|not).* any longer	*I cannot stand this constant agony any longer.*	206
Information	I have.{0,20} (\\$|dollar)	I believe *I have around* $200.00 in account.	71
Instructions	I leave.{0,50} to my	*I leave all my earthly possessions to my* wife.	209
Love	(loads|lots) of love	*Loads of love*, Your son, John.	57
Pride	proud of	I'm *proud of* them.	2
Sorrow	my heart.{0,20} br	*My heart has been broken* I thought I was strong enough to go through with what your father started.	11
Thankfulness	I.{0,10} appreciate	*I deeply appreciate* your kindness and affection.	26

As part of generalization, individual words were mapped to different types of categories. At the most general level, words were mapped to their POS, ie, syntactic information attached to words during linguistic pre-processing. A total of 21 POS classes were used. While, this certainly overgeneralizes individual words, some types of POS information do provide useful clues for topic classification purposes. For example, tokens containing numerical information, and thus tagged as CD (ie, cardinal number), vary widely in terms of their content, which itself does not provide information that can improve the classification. However, their general type alone (ie, CD) provides a useful classification clue, as it was mainly associated with the *Information* and *Instructions* classes where it usually represents information such as addresses, phone numbers, quantity, dates, etc. Here are some representative examples to illustrate its use in sentences classified as *Information*:

- All my things are at <CD>*3333*</CD> Burnet Ave.
- Can be reached by phone State—<CD>*636–2051* </CD>.
- $ <CD>*147.00*</CD> in purse.
- I am paid up there till <CD>*01–01–01*</CD>, there is $ <CD>*145.00*</CD> in cash in bank book.
- My Social Security number is <CD>*333–33–3333* </CD>.

At a slightly lower level, we generalized individual words into their lexical domains. Namely, WordNet synsets are organized into 45 lexical domains based on syntactic category and logical groupings.[18] Here we provide some examples of words from the *verb.possession* domain (ie, verbs of buying, selling, owning, etc.) used in sentences classified as *Instructions*:

- <possession>*Buy*</possession> John some clothes or what he needs most.
- I give my sister power of attorney to <possession> *cash*</possession> checks.
- Don't <possession>*pay*</possession> any more rent.
- Don't <possession>*sell*</possession> the new house, please.
- I also told Mr. J. Johnson not to <possession> *spend*</possession> any more than $300.00 on my burial.

Where we needed more fine-grained groupings of words not provided by WordNet, we assembled our own lexicons to support lexical representation of relevant semantic categories (see Table 1) in addition to lexicons of words related to topics considered (see Table 2).

- That <health>*arthritis*</health> and hardening of the <health>*arteries*</health> are too much for me.
- Have him notify my <occupation>*lawyer*<occupation>.
- May <religious>*God*</religious>, <family>*family*</family>, <people>*friends*</people>, and <name>*John*</name> forgive me.
- <instruction>*Bury*</instruction> me at least of <instruction>*expense*</instruction>.
- You have been <anger>*mean*</anger> and also <anger>*cruel*</anger>.
- <love>*Dearest*</love> <love>*darling*</love> I <love>*love*</love> you.

Having generalized individual words in the ways described above, we counted the frequencies of their general categories and used them as features instead of counting the frequencies of individual words. This was done for all generic feature types, apart from the occupation type, where we differentiated between the words as they were shown to be associated with different topics, eg,

- Fear: I know I should see a <occupation>*doctor*</occupation> but I've been afraid to ask you.
- Instructions: Have him notify my <occupation>*lawyer*</occupation> John J. Johnson.
- Anger: They are gang of <occupation>*politicians*</occupation> and grafters.

We also used individual words as features for those selected using mutual information,[15] an additional strategy we used to reduce the dimensionality of the feature space, to address a potential loss of important information due to overgeneralization. Mutual information, as one of the most effective feature selection mechanisms, was used to reduce the number of words considered by identifying the most informative ones. In this case, individual words were still used as features and their frequencies were counted. This reduced the number of words considered from 4,506 to 153 most informative ones.

In order to identify explicit mentions of a range of different emotions, we used the WordNet-Affect[3] lexicon described in the previous section, eg,

- I've tasted the last bitter dregs of <despair>*despair*</despair>, disillusion, <forlornness>*loneliness*</forlornness>, <misery>*misery*</misery>, poverty, strife, confinement, <positive-concern><negative-concern><distress>*worry*</distress></negative-concern></positive-concern>, <grief>*grief*</grief>, failure, and everything else that could contribute to an ignominious end.

Each word found in the lexicon was mapped to the associated emotion, often expressed by the given word. Where a word was mapped to multiple emotions (eg, *worry* was mapped to three emotions: *positive-concern, negative-concern* and *distress*), all mappings were used, that is—no disambiguation was performed. This type of noise was left to be dealt with or resolved by the machine learning algorithm at a later stage. Each occurrence of an emotion word was used to increase the corresponding feature value. In addition, the hierarchy of emotions was used to generalize emotions at all levels. When a word was mapped to an emotion, we also used its ancestors as features and increased their values too, eg, the word *despair* was mapped directly to the *despair* emotion, and indirectly to its ancestors: *negative-emotion, emotion, affective-state, mental-state*, and *root* respectively. This allowed the machine algorithm to decide on the optimal subset of emotional features to use as well as the optimal level of granularity used to differentiate between emotions. Finally, only those emotions identified in the training dataset were used as features, which reduced the number of these features from 306 to 58.

To identify the emotional tone of a sentence (ie, its positive or negative polarity) we used SentiWordNet,[4] which maps words to their positive and negative scores. The polarity of a sentence was calculated by aggregating the polarity scores of individual words in three ways: finding the maximum score, summing up the scores and averaging them. We used all of the aggregated scores as features, again leaving it to the machine learning algorithm to decide on the most useful features in terms of classification performance. Two other emotive lexicons were used, which simply classify words as being positive

or negative.[5,6] The positive or negative polarity of a sentence was quantified as the percentage of positive and negative words found in the sentence. We also counted the occurrence of negation words (eg, *no, never, hardly*, etc.) to help identify negative tone. In addition, as an emotion represents a psycho-physiological experience of an individual's state of mind, it is often expressed subjectively from the first person's point of view and often involves other people, who are the cause or the object of the emotion. We already used four lexicons to support the recognition of different groups of people or their roles (see Table 1), but since people are most generally referred by pronouns, we also counted the occurrence of personal and possessive pronouns to help identify such references. We differentiated between the first person as the potential subject of an emotion and all other persons (except for the gender-neutral ones) as its object.

So far, we used the bag-of-words model in which each sentence is represented as an unordered collection of individual words normalized to their lemmas, ignoring the relationships between them. However, such information, eg, represented through the use of bigrams, may substantially improve the quality of features, thus increasing the overall classification performance.[19] Still, matching longer phrases may lead to a decrease in performance due to high dimensionality and low frequency.[20] Therefore, it is essential to optimize the choice of more complex features. Instead of matching exact phrases, we opted for regular expressions as a more flexible way of representing relationships between individual words. Such flexibility results in both lower dimensionality and higher frequency of the features, thus avoiding the problem of degrading the performance associated with the introduction of longer phrases into the feature space. We also conflated the rules by associating them with specific topics and aggregating their frequencies to further address the dimensionality and frequency issues. A separate feature was introduced for each topic and its values were calculated as the number of regular expressions matched from the corresponding set. Table 3 provides examples of regular expressions used to introduce more complex features on top of those based on individual words.

To summarize, each sentence was mapped to its feature vector consisting of different feature types described in Table 4.

Principal Component Analysis

Compared to a brute-force bag-of-words approach, we managed to significantly reduce the number of features from over 4,500 to only 353. The selected features combined a set of specific words chosen using mutual information as well as more general lexico-semantic properties of individual words. This addressed the high dimensionality and data sparsity issues. The next problem to be resolved was to identify dependencies between the selected

Table 4. Features used to represent a sentence.

Feature type	Total	Feature value as "the total number of …"
Sentence length	1	…tokens in a sentence.
POS	21	…tokens with a given POS tag.
WordNet lexical domains	45	…tokens mapped to a given lexical domain.
Lexicons (semantic categories)	7	…tokens found in a given lexicon.
Lexicons (topic clues)	15	…tokens found in a given lexicon.
Occupation words	24	…occurrences of a given word in a sentence.
Informative words (MI)	153	…occurrences of a given word in a sentence.
WordNet–Affect lexicon	58	…occurrences of words mapped to a given emotion directly or indirectly through inheritance.
SentiWordNet lexicon	6	Positive/negative polarity scores of individual words in a sentence aggregated as their maximum, average or sum.
Positive/negative polarity	4	…positive/negative words found in a given lexicon normalized by sentence length.
Negation	1	…occurrences of negation words in a sentence.
Pronouns	2	…occurrences of personal and possessive pronouns (1st person vs. all other persons).
Pattern–matching rules	16	…patterns successfully matched to a sentence.

features, as they may decrease the performance of a naïve Bayes classifier, whose underlying probability model is based on an assumption of independence between the features. Therefore, we applied principal component analysis (PCA),[12] a mathematical procedure that transforms a complex feature space into a simplified structure that underlies it, by identifying the underlying, uncorrelated variables (called principal components) that account for most of the variation in the training data. We used Weka,[11] a popular suite of machine learning software, to perform PCA and retain enough principal components to account for 95% of the variance in the training data. As a result, the dimensionality of the data space was reduced from 353 original features to 245 principal components. Here are the five top-ranked principle components:[a]

1. –0.17VB–0.169SWN.possum-0.16SWN.negsum-0.158VERB.social–0.157VERB.stative…
2. –0.199CLUE.information–0.184NN–0.179NOUN.artifact–0.174CD–0.172NOUN.communication…
3. –0.215CLUE.forgiveness–0.208WNA.devotion-0.205WNA.affection–0.189CLUE.guilt–0.189NOUN.person…
4. –0.299WNA.distress–0.299WNA.anxiety-0.28WNA.joy–0.24CLUE.happiness_peacefulness–0.235PATTERN.hopefulness…
5. 0.261WNA.defeatism+0.259MI.love+0.232CLUE.love–0.227WNA.devotion–0.225WNA.affection…

Obviously, PCA helped identify new meaningful underlying variables. It is interesting to notice how different types of features were grouped together reflecting their related meanings. For instance, in the second principal component manually identified lexical clues related to the *Information* class were associated with nouns classified as *artifact* or *communication* in WordNet. Similarly, in the fifth principal component we can observe how the actual word *love* chosen as a separate feature using mutual information was associated with lexical clues for the *Love* class, as well as the related emotions from WordNet–Affect, *devotion* and *affection*.

[a]Features written in capital letters only refer to POS tags used as features, eg, VB, NN and CD refer to verb, noun and cardinal number respectively. Qualifiers written in capital letters refer to the type feature, eg, VERB and NOUN refer to lexical domains of verbs and nouns in WordNet, WNA and SWN refer to emotive lexicons WordNet–Affect and SentiWordNet, whereas CLUE, PATTERN and MI correspond to features based on lexical clues, pattern matching and mutual information.

Naïve-Bayesian Classification

Having performed PCA to minimize feature dependence, we applied a naïve Bayes classifier,[21] a probabilistic learning method based on Bayes' theorem, which combines evidence e from multiple sources of data to estimate the probability of a hypothesis h:

$$P(h|e) = \frac{P(h)P(e|h)}{P(e)}$$

When multiple hypotheses are considered, Bayes' theorem provides the probability of each hypothesis being true given the evidence. During a training step, a naïve Bayes classifier uses the training data to estimate the parameters of a probability distribution (on the right-hand side of the equation). During a testing phase, a naïve Bayes classifier uses these estimations to compute the posterior probability (the left-hand side of the equation) given a test sample. Each test sample is then classified using the hypothesis with the largest posterior probability.

We again used Weka[11] to train a naïve Bayes classifier using the PCA-transformed dataset. In particular a discriminative multinominal naïve Bayes classifier was chosen from other varieties offered in Weka as it provided the best results during cross-validation. To classify sentences in the previously unseen test dataset, we applied the naïve Bayes model produced in the training phase to obtain the posterior probability for each class. Each sentence was classified using a single topic with the largest posterior probability. Given the heterogeneous nature of sentences not labeled with any topic, and therefore a difficulty in correctly predicting a topic as *none* when the probability model is used, we introduced a threshold for accepting the largest posterior probability in order to improve the classification precision. A sentence was classified using the topic suggested by the largest posterior probability only if its value exceeded a set threshold, otherwise it was classified as *none* motivated by the fact that it was the largest and the most heterogeneous class. We used three threshold values (0, 0.30 and 0.50) to produce the results for the three system runs submitted. The higher threshold

values should improve precision based on the fact that only sufficiently high probability values are used to make predictions. Conversely, lower threshold values should improve recall by allowing less probable classes to be suggested as well. An optimal threshold should balance the tradeoffs between precision and recall and maximize the F–measure. We used the precision and recall values obtained on the training set using different thresholds to choose three different values.

Rule-Based Classification

The given task allowed multiple topics to be associated with a single sentence. To support multiple classifications, we could have used more than one highly probable class suggested by the naïve Bayes classifier. However, given the relatively small size of the training dataset, we believed that accepting classes with lower probabilities would significantly reduce the precision of the method, and consequently the overall classification performance estimated by the F–measure. Alternatively, we chose to apply rule–based classification following the predictions suggested by the naïve Bayes classifier. We relied on a subset of the pattern–matching rules illustrated in Table 3. By using only those rules that exhibited high precision on the training data, we aimed to maximize the precision. On the other hand, by supplementing the existing predictions with other correct classifications, we aimed to increase the recall. By increasing the recall while at least maintaining the precision, we aimed to improve the overall classification performance.

Evaluation

We received two datasets: 600 notes to be used for training and additional 300 notes to be used for testing. All sentences were annotated with topics shown in Table 2, with multiple annotations allowed where appropriate. The training dataset consisted of a total of distinct 4,315 sentences, out of which 2,145 sentences were associated with a total of 2,494 annotations. Note that we removed duplicated sentences from the dataset. We also manually corrected some annotations in the training dataset to remove some types of noise. For example, the same sentence *"Thanks."* was annotated as *Thankfulness* in most cases, but such annotation was missing in one case and it was annotated as *Instructions*, which we believed to be incorrect. This may account for a slight difference in the numbers reported for the training data we used and the originally distributed training data. Further, the testing dataset consisted of a total of 2,037 sentences, out of which 1,091 sentences were associated with a total of 1,272 annotations.

We submitted three system runs with three threshold values used to reject less probable predictions suggested by the naïve Bayes classifier described earlier. The performance was evaluated using recall and precision, as well as their combination into the F-measure, whose values were micro-averaged.[1] Table 5 describes the results provided by the organizers for the three runs, where T, N, P, R, and F denote the threshold, number of annotations, precision, recall and F-measure respectively. As expected, the lower the threshold the better the recall, and vice versa, the higher the threshold the better the precision. The best overall performance was achieved in Run 2, with the F-measure of 53.34%. The results achieved by the 26 teams that took part in the challenge were summarized by the organizers as follows: mean 48.75, standard deviation 0.0742, minimum 29.67, maximum 61.39, and median 50.27.

To gain more insight into the results, we ran our own evaluation script to obtain the performance measures across all classes for Run 2. As before, Table 6 provides the values for the three evaluation measures, in addition to the numbers of true positives (TP), false positives (FP) and false negatives (FN), for both training and testing datasets. The threshold used to obtain the results on the training data was also 0.30. Our evaluation script does not differentiate between documents, as we considered sentences as such and we removed the duplicate ones, which is accounted for a slight difference between these values and the values provided by the organizer. As expected, poor results were achieved for small-size classes such *Abuse* and *Pride* with only 9 and 15 annotations respectively in the training dataset, which was insufficient to successfully train the naïve Bayes

Table 5. The three test run results.

Run	T	N	P	R	F
Run 1	0.00	1,250	53.68	52.75	53.21
Run 2	0.30	1,199	54.96	51.81	53.34
Run 3	0.50	1,095	55.71	47.96	51.54

Table 6. The evaluation results achieved during training and testing phases.

Topic	Training dataset						Testing dataset					
	TP	FP	FN	P	R	F	TP	FP	FN	P	R	F
Abuse	6	23	3	20.69	66.67	**31.58**	0	11	5	0.00	0.00	**0.00**
Anger	25	28	44	47.17	36.23	**40.98**	7	10	19	41.18	26.92	**32.56**
Blame	56	32	51	63.64	52.34	**57.44**	4	28	41	12.50	8.89	**10.39**
Fear	11	8	13	57.89	45.83	**51.16**	4	8	9	33.33	30.77	**32.00**
Forgiveness	5	3	1	62.50	83.33	**71.43**	2	4	6	33.33	25.00	**28.57**
Guilt	110	92	96	54.46	53.40	**53.92**	54	45	63	54.55	46.16	**50.00**
Happiness_peacefulness	15	6	10	71.43	60.00	**65.22**	6	4	10	60.00	37.50	**46.15**
Hopefulness	17	12	30	58.62	36.17	**44.74**	2	9	36	18.18	5.26	**8.16**
Hopelessness	286	107	168	72.77	63.00	**67.53**	121	74	108	62.05	52.84	**57.08**
Information	143	91	149	61.11	48.97	**54.37**	44	57	60	43.56	42.31	**42.93**
Instructions	580	263	224	68.80	72.14	**70.43**	224	165	156	57.58	58.95	**58.26**
Love	245	96	50	71.85	83.05	**77.04**	139	65	57	68.14	70.92	**69.50**
Pride	11	6	4	64.71	73.33	**68.75**	1	0	8	100.00	11.11	**20.00**
Sorrow	19	16	31	54.29	38.00	**44.71**	4	7	30	36.36	11.76	**17.78**
Thankfulness	82	39	8	67.77	91.11	**77.73**	41	51	4	44.57	91.11	**59.85**
Overall	**1611**	**822**	**882**	**66.21**	**64.62**	**65.41**	**653**	**538**	**612**	**54.83**	**51.62**	**53.18**

classifier or generalize them into appropriate pattern–matching rules. Other classes such as *Hopefulness* and *Sorrow* proved to be too noisy to properly train the naïve Bayes classifier and poor results achieved during the training phase were further deteriorated during the testing phase. There appeared to be some overfitting in classes such as *Blame*, where the F–measure dropped from 57.44% to 10.39%. The smallest decrease in performance was noticed for *Guilt*, whose F–measure of 50.00% was comparable to that of 53.92% achieved during training. Relatively good results with F–measure over 50% were achieved for *Hopelessness*, *Instructions, Thankfulness* and *Love*. *Hopelessness* and *Instructions* were relatively large-size classes and more precisely defined compared to others (eg, *Sorrow*), which provided enough training data with less noise for the naïve Bayes classifier to properly model these classes. *Thankfulness* and *Love* varied less in their content compared to other classes, and they were therefore relatively easy to model with lexical clues and pattern–matching rules. Overall, the result achieved in both training and testing phases were well balanced between precision and recall. The decrease in F–measure from 65.41% to 53.18% was expected given a relatively small size of the training dataset for the large number of annotations distributed over 15 classes. The final result was still significantly better than the average performance of 48.75% achieved by the 26 participating teams.

Conclusion

The given topic classification task is a supervised learning problem, which can be approached with machine learning (ML) or a rule-based methodology. Given the subjective nature of the given classification task and the variety of sentences in terms of their lexical content and syntactic structure, which would lead to proliferation of suitable classification rules if such rules could be identified, we opted for ML as the primary methodology used in our approach. Still, we incorporated pattern–matching rules as part of feature extraction, as well as for rule–based classification for a subset of sentences, whose lexico-syntactic properties exhibited high correlation with the associated topics. Another motivation for favoring the ML approach was the provision of annotated data of a decent size to support the training of different ML algorithms. We conducted experiments with a range of different algorithms supported by Weka,[11] a popular suite of ML software. We took advantage of the naïve Bayes classifier, as it does not necessarily require a lot of training data to perform well,[21] which was confirmed by our experiments. Given the F–measure of 53% achieved, our system significantly outperformed the majority of other competing systems, based on the comparison with the median F–measure of 50%. Given a large amount of noise noticed by manually inspecting the training data, we believe that the performance can be significantly increased with the quality and size of

the training data available. More importantly, the core of the system is portable between different domains with the changes restricted to the lexicons assembled internally to model relevant semantic types and the set of pattern–matching rules associated with the classification scheme.

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. Biomedical Informatics Insights 2012;5 (Suppl. 1):3–16.
2. Fellbaum C, editor. WordNet—An electronic lexical database. MIT Press; 1998:423.
3. Strapparava C, Valitutti A. WordNet-Affect: an affective extension of WordNet. Proc 4th Int Conf on Language Resources and Evaluation, Lisbon, Portugal; 2004:1083–86.
4. Esuli A, Sebastiani F. SentiWord Net: A publicly available lexical resource for opinion mining. Proc 5th Int Conf on Language Resources and Evaluation, Genoa, Italy; 2006:417–22.
5. Hu M, Liu B. Mining and summarizing customer reviews. Proc the ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, Seattle, Washington, USA; 2004:168–77.
6. Mohammad S, Dorr B, Dunne C. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. Proc Conf on Empirical Methods in Natural Language Processing, Singapore; 2009: 599–608.
7. Toutanova K, Klein D, Manning C, Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. Proc North American Chapter of the ACL—Human Language Technologies, Edmonton, Canada; 2003: 252–59.
8. JWI (the MIT Java WordNet Interface): http://projects.csail.mit.edu/jwi/.
9. Jazzy (Java spelling checker API): http://www.ibm.com/developerworks/java/library/j-jazzy/.
10. SQLite: http://www.sqlite.org/.
11. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The Weka data mining software: An update. *SIGKDD Explorations*. 2009; 11(1):10–8.
12. Jolliffe IT. Principal Component Analysis. Springer-Verlag; 1986:487.
13. Bad word list and swear filter: http://www.noswearing.com/dictionary.
14. Causes of suicidal tendencies: http://www.wrongdiagnosis.com/symptoms/suicidal_tendencies/causes.htm.
15. Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. Proc 14th Int Conf on Machine Learning, Nashville, USA; 1997:412–20.
16. Simple Concordance Program: http://www.textworld.com/.
17. Hughes GF. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*. 1968;14(1):55–63.
18. WordNet lexicographer files: http://wordnet.princeton.edu/man/lexnames.5WN.html.
19. Tana CM, Wanga YF, Leeb CD. The use of bigrams to enhance text categorization. *Information Processing and Management*. 2002;38(4):529–46.
20. Lewis D. Feature selection and feature extraction for text categorization. Proc Workshop on Speech and Natural Language, San Mateo, USA; 1992: 212–17.
21. Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*. 1997;29:103–37.