Biomedical Informatics Insights

ORIGINAL RESEARCH

A Combined Approach to Emotion Detection in Suicide Notes

Alexander Pak, Delphine Bernhard, Patrick Paroubek and Cyril Grouin

LIMSI–CNRS, 91403 Orsay, France. Corresponding author email: cyril.grouin@limsi.fr

Abstract: In this paper, we present the system we have developed for participating in the second task of the i2b2/VA 2011 challenge dedicated to emotion detection in clinical records. On the official evaluation, we ranked 6th out of 26 participants. Our best configuration, based upon a combination of both a machine-learning based approach and manually-defined transducers, obtained a 0.5383 global F-measure, while the distribution of the other 26 participants' results is characterized by mean = 0.4875, stdev = 0.0742, min = 0.2967, max = 0.6139, and median = 0.5027. Combination of machine learning and transducer is achieved by computing the union of results from both approaches, each using a hierarchy of sentiment specific classifiers.

Keywords: emotion detection, machine-learning, SVM classifier, transducers

Biomedical Informatics Insights 2012:5 (Suppl. 1) 105–114

doi: 10.4137/BII.S8969

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

Introduction

In this paper, we present the LIMSI participation in the second track of the i2b2/VA 2011 challenge, whose aim was the detection of emotions expressed in a corpus of suicide notes, provided by the organizers. After a short reminder of the challenge requirements and a description of the corpus, we present our natural language processing pipelines. We then report on the evaluation of the different approaches we have tried and discuss our results on the task.

Related Work

One of the earliest approaches for automatic analysis of suicide notes was described by Stone et al.[1] They have used a system called General Inquirer created at IBM to detect fake suicide notes. The core of the General Inquirer system is a dictionary containing 11,789 senses of 8,641 English words (ie, certain words have several senses), each mapped to one or more of 182 categories, such as "positive", "negative", "self", "family", etc. The authors used the distribution of categories to distinguish between simulated and genuine suicide notes. The evaluation, using 33 simulated notes and 33 real notes, showed that the General Inquirer system was able to correctly identify 17 out of 18 test note pairs, which is a better performance than the one of random classification.

A more recent work by Pestian et al[2] used features extracted from the text of the notes to train different machine-learning classifiers. The features were: number of sentences, word distribution statistics, distribution of part-of-speech tags, readability scores, emotional words and phrases. The performance of machine-learning models were compared against the judgments of psychiatric trainees and mental health professionals. Experimental evaluations showed that the best machine-learning algorithms accurately classified 78% of the notes, while the best accuracy obtained by the human judges was 63%.

To our knowledge, there is no published research on automatic emotion detection in suicide notes or similar topics.

Among the categories that participating systems had to use to tag sentences, there were two categories not related to emotions: instructions and information. For these, previous work on objectivity detection is clearly relevant. In the related domain of sentiment classification, Riloff and Wiebe[3] proposed using lexico-syntactic patterns for classifying sentences as objective or subjective. The patterns contain both words and variables corresponding to part-of-speech tags, eg, `<x> drives <y> up the wall`, in order to deal with different surface forms of the same expressions. The patterns are automatically acquired using a bootstrapping approach. High-precision subjectivity classifiers first classify sentences as subjective or objective. Then, syntactic templates are applied to the sentences in order to generate extraction patterns which instantiate the templates. Finally, the patterns are ranked based on how often they occur in subjective versus objective sentences and the best patterns are selected. Subsequently, the patterns can be used for identifying other subjective sentences.

Pang and Lee[4] found that they could improve opinion detection by removing the sentences they considered as objective, before classifying. Pak and Paroubek[5] used a corpus made of text messages from the Twitter accounts of 44 popular newspapers and magazines, such as *New York Times, Washington Post*, etc, as training material for a Bayesian classifier to build an objectivity detector for Twitter posts.

Challenge Requirements

The second track of the i2b2 2011/VA Challenge consists in identifying the opinion expressed in suicide notes by tagging sentences with one or several of the following fifteen categories:[6] `instructions`, `information`, `hopelessness`, `guilt`, `blame`, `anger`, `sorrow`, `fear`, `abuse`, `love`, `thankfulness`, `hopefulness`, `happiness_peacefulness`, `pride`, `forgiveness`. Note that the first two categories do not describe emotions but objective material. Sentences which do not fall into one of these categories have to be left untagged. The unique source of information provided to the participants is a training corpus, which has been hand-tagged.

Corpus Description

The training corpus consists of 600 suicide notes hand-annotated, while the test corpus is composed of 300 suicide notes. Those documents are of several kinds, mainly last will and testament. The corpus has been fully de-identified* (names, dates, address) and tokenized.

*Each name has been replaced by a generic name (*Jame, John, Mary*) and all addresses by the one of Clincinnati Children's Hospital Medical Center.

Each document from the training corpus is very brief, on average: 7 sentences and 132.5 tokens (mainly words but also punctuation marks) per document. Proportions are similar for the test corpus.

Documents include spelling errors (*conctract – poicies*). There are a few residual processing errors, more particularly the apostrophe in genitives and abbreviations, where spaces have been introduced (*could n't – Mary' s*) or the apostrophe replaced by a star with missing tokenization (*don*t – wasn*t*). Sentence segmentation is noisy (several short sentences are sometimes encoded as one single sentence). In the training corpus, 2,173 different sentences have been hand-annotated, among them 302 sentences received several category labels (see Table 1).

Lines with several annotated emotions are long sentences: the two lines composed of five emotions are between 73 and 82 tokens long. As an example, the longest line (*"My Dearest Son Bill : Please forgive mother for taking this way out of my umbearable trouble with your Dad Smith—Son I 've loved you and Dad beyond words and have suffered the tortures of hell for Smith but his lies and misconduct to me as a wife is more than I can shoulder any more—Son God has been good to you and mother and please be big and just know that God needs me in rest ."*) has been annotated with the five following emotions classes: `abuse`, `blame`, `guilt`, `hopelessness` and `love`. In Table 2, we give the distribution of the annotation among the different categories.

Here is an example of annotation from the test corpus with its reference annotation.

```
INPUT FILE: 20080901735_0621.txt

John : I am going to tell you this
at the last .
You and John and Mother are what I
am thinking—I ca n't go on—my life
is ruined .
I am ill and heart—broken .
Always I have felt alone and never
more alone than now .
John .
Please God forgive me for all my
wrong doing .
I am lost and frightened .
God help me ,
Bless my son and my mother .

OUTPUT FILE: 20080901735_0621.con.txt

c = "You and John and Mother
are what I am thinking—I can't
go on—my life is ruined ." 2:0
2:21||e = "hopelessness"
c = "Always I have felt alone and
never more alone than now ." 4:0
4:11|| e = "sorrow"
c = "I am lost and frightened ." 7:0
7:5||e = "fear"
```

We have found the task to be difficult for the following reasons.

- **Multiple labels per sentence**. In the following example, the two labels `hopelessness` and `instructions`: were provided by the annotators:

  ```
  In case of sudden death , I wish to
  have the City of Cincinnati burn
  my remains with the least public-
  ity as possible as I am just a sick
  old man and rest is what I want .
  ```

Table 1. Number of sentences for each number of annotations per line in both training and test corpora.

#annotations/sentence	0	1	2	3	4	5
Train	2460	1871	266	27	7	2
Test	811	946	134	15	2	1

Table 2. Number of annotations for each category in both training and test corpora.

	Train	Test
Abuse	9	5
Anger	69	26
Blame	107	45
Fear	25	13
Forgiveness	6	8
Guilt	208	117
Happiness/peacefulness	25	16
Hopefulness	47	38
Hopelessness	455	229
Information	295	104
Instructions	820	382
Love	296	201
Pride	15	9
Sorrow	51	34
Thankfulness	94	45

Multiple labeling makes the task more difficult for machine-learning classifiers that normally work with a single label per sample.

- **No annotation**. When no annotation was assigned to a sentence, two interpretations are possible: either there is no emotion expressed, or there was a disagreement between the annotators. Here is an example, where a note could have been annotated with the `love`, but was left without annotation:

```
I love you all, but I can't continue
to be a burden to you.
```

The ambiguous "no annotation" assumption adds noise to the training data.

- **Fine grained labels**. Certain labels have very close meanings and are consequently hard to distinguish from one another. As an example, `information` *vs.* `instructions`, `guilt` vs. `forgiveness`, or `sorrow` *vs.* `hopelessness`.
- **Unbalanced distribution of labels**. Certain labels in the training (and test) set appear much more frequently than others. The most frequent label `instructions` appears 820 times in the training set, while the label `forgiveness` appears only 6 times. This makes it all the more difficult to learn rare classes, due to possible biases during the training.
- **Lack of additional training data**. The task organizers provided the training corpus, however it is extremely difficult to find additional training material. To our knowledge, there is no publicly available text corpora of suicide letters or other similar resources. Construction of such a corpus is also problematic due to the nature of the task and lack of information about the guidelines used by the annotators.

Our Approach

In order to answer the challenge, we created a system that uses both a machine-learning approach and hand-written rules to detect emotions. Our intention was to create a high-precision rule-based system backed up by a machine-learning algorithm to improve recall and to generalize on unknown data.

Machine-learning based approach

In our machine-learning based approach, we trained an SVM classifier using different features extracted from the training set. We used the LIBLINEAR package[7] with a linear kernel and default settings. In order to perform multi-label classification, we employed the one-versus-all strategy, ie, we trained an SVM classifier for each emotion independently. Each classifier provides a decision whether a given sentence contains the emotion it was trained to recognize or not. Such a setting allows us to have multiple labels per line or no labels at all, when all the classifiers returned a negative answer.

Here is a list of features that we have used to build our classification model:

- **N-grams.** N-gram models are widely used as a common approach for representing text in information retrieval, text categorization, and sentiment analysis.[8] We used unigrams and bigrams, with normalized binary weights, such that for a given text T represented as a set of terms:

$$T = \{t_1, t_2, \ldots, t_k\} \quad (1)$$

we define the feature vector of T as

$$TF = \left\{ \frac{1}{\text{avgtf}(t_1)}, \ldots, \frac{1}{\text{avgtf}(t_k)} \right\} \quad (2)$$

where avg.tf(t_i) is a normalization function based on a term average frequency:

$$\text{avg.tf}(t_i) = \frac{\sum_{\forall T, t_i \in T} \text{tf}(t_i)}{|\forall T, t_i \in T|} \quad (3)$$

A procedure of attachment of the negation particle was performed to capture the negations, ie, particles "no" and "not" were attached to a following word when generating n-grams.

- **POS-tags**. We used the TreeTagger[9] to obtain part-of-speech tags for words and also to perform sentence segmentation as some lines contain multiple sentences. To construct a feature vector, we used the frequencies of tags in a sentence. The important information provided by tags features are: the usage of auxiliary verbs, verb properties (tense, person, voice, mood), usage of adjectives and adverbs and their comparative or superlative forms, usage of cardinal numbers (important for distinguishing informative classes), and punctuations (such as the symbol $). It has been shown that the distribution of POS-tags is different in subjective and objective texts, and texts with positive and negative polarities.[5]

- **General Inquirer**. We used the dictionary from the General Inquirer (GI) system to create supplementary features as follows. Each word from a tested sample was lemmatized if possible. The lemma was searched in the GI dictionary and if found, all the associated categories were added to the bag of categories. Next, for each of the 182 GI categories, we counted the occurrences within the sentence. We got a 182-length feature vector. No disambiguation was done at this point. If multiple senses existed in the dictionary for a given lemma, all the categories associated with the senses were added to the bag.
- **ANEW.** In order to capture the mood of a text, we used the Affective Norms of English Words[10] lexicon. The lexicon contains 1,034 English words with associated numerical scores of valence, arousal, and control. To construct a feature vector, we represented each word from ANEW in a 3-dimensional space, where each dimension represents a word's score. Next, we divided this space equally into N^3 buckets and counted the number of words from a sentence that fall into each bucket. The scores in ANEW dataset take a value between 1 and 9, thus all the words may have coordinates starting from (1, 1, 1) to (9, 9, 9). For example, we set $4^3 = 64$ buckets. Then, the first bucket would contain words with coordinates from (1, 1, 1) to (3, 3, 3), the second bucket: from (1, 1, 3) to (3, 3, 5) etc. Thus, we would obtain a 64-length feature vector.
- **Dependency graphs**. Typed dependency based features are considered to be effective when capturing sentiment expressions.[11,12] We extracted subgraphs from the sentence dependency trees produced by the Stanford Lexical Parser[13,14] in order to create patterns of sentiment expressions.
- **Heuristic features**. Finally, we added a number of heuristically produced features: the position of the sentence with respect to the beginning of the note, the presence of the following words in the sentence: "god", "thank", "please", "car", and "Cincinnati".

On different stages of classification, we used different combinations of the listed features. In order to combine features, we simply concatenated the produced feature vectors.

It has been shown that hierarchical classifiers yield better results than flat ones, when classifying emotions.[15] We have organized the labels into a hierarchy as shown in Figure 1.

Our final algorithms is as follows.

1. First, we have trained an annotation detector to distinguish sentences with annotations from unannotated ones. Features used: POS-tags, General Inquirer.
2. Next, the sentences considered to have annotations were fed to a subjectivity detector, to separate subjective sentences from objective ones. Features used: heuristic, POS-tags, General Inquirer.
3. Objective sentences were then classified between: `information` and `instructions`. Features used: uni-grams, bigrams, General Inquirer, dependency graphs.

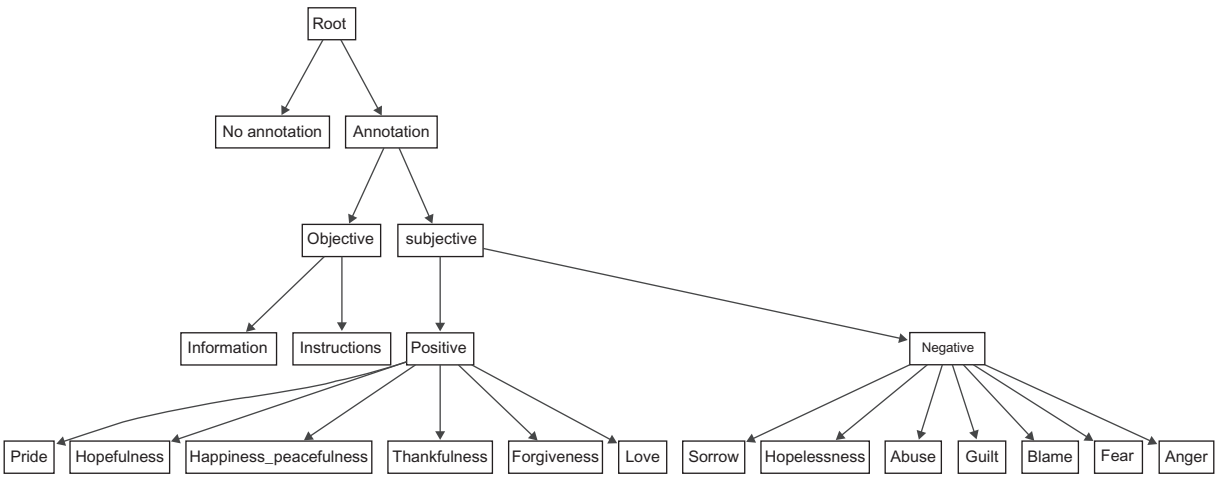

Figure 1. Emotions hierarchy.

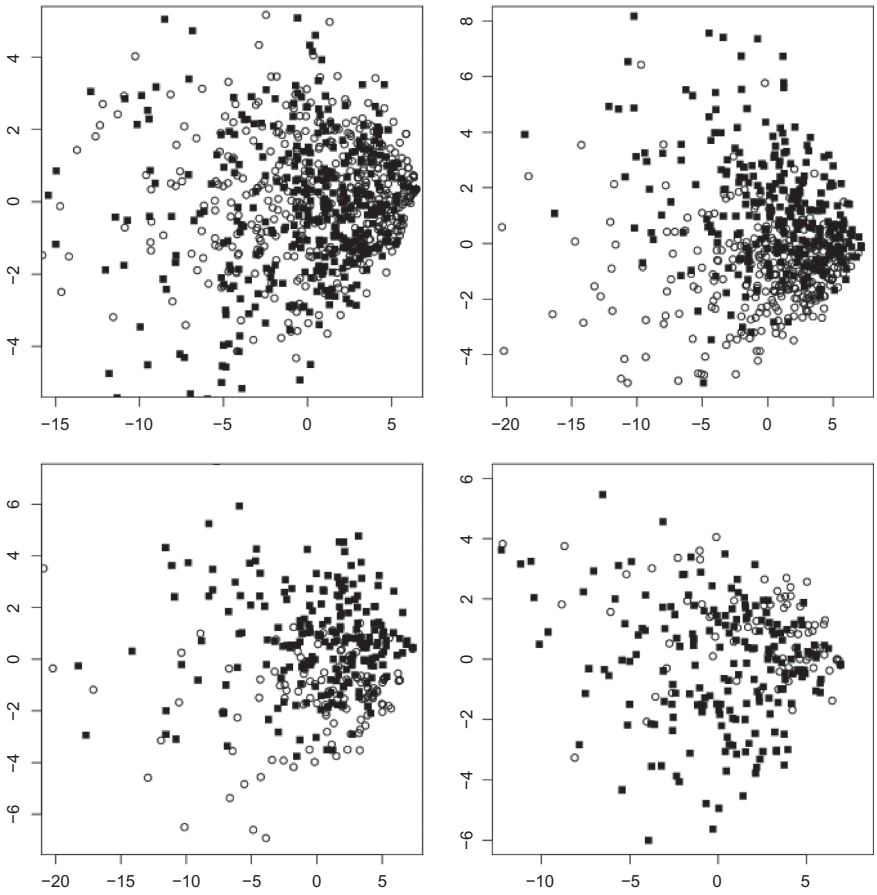

Figure 2. Visualizing samples in 2-dimensions: annotated (black squares) vs. not annotated (white discs), upper left corner (random 20% of total data); subjective (white discs) vs. objective (black squares), upper right corner (random 33% of total data); positive (white discs) vs. negative (black squares), lower left corner (random 33% of total data); information (white discs) vs. instructions (black squares), lower right corner (random 33% of total data).

4. Subjective sentences were divided into emotions with a positive polarity and the ones with a negative polarity, using a polarity classifier. Features used: POS-tags, ANEW.
5. Sentences with a negative polarity were further classified according to 7 classes: `sorrow`, `hopelessness`, `abuse`, `guilt`, `blame`, `fear`, `anger`. Features used: unigrams, bigrams, General Inquirer, dependency graphs.
6. Sentences with a positive polarity were further classified among 6 classes: `pride`, `hopefulness`, `love`, `happiness/peacefulness`,

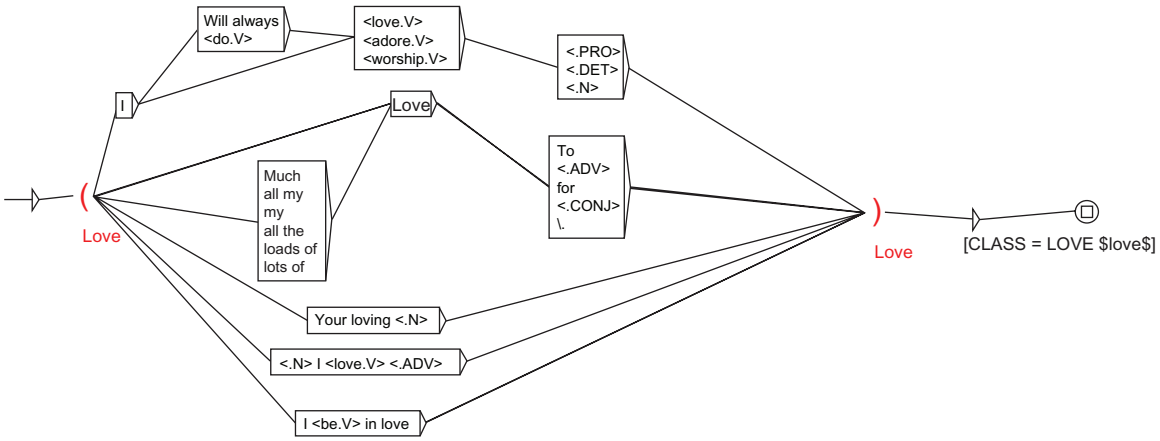

Figure 3. Example transducer for the emotion class `love`.

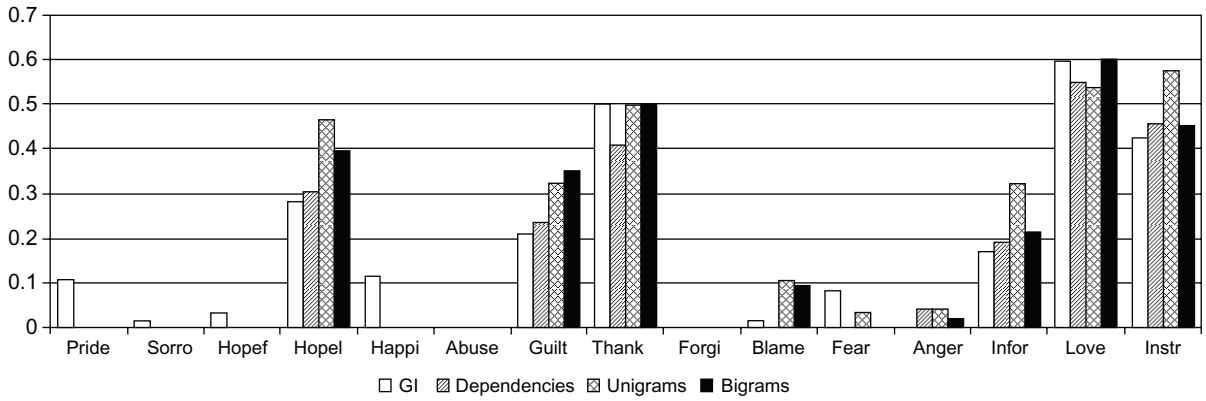

Figure 4. Performance of different features used for emotion detection across the classes.

`thankfulness`, `forgiveness`. Features used: unigrams, bigrams, General Inquirer, dependency graphs.

In order to estimate the task difficulty, we have plotted the data on a 2-dimension graph using PCA for dimension reduction and General Inquirer features as shown in Figure 2. As we can see from the figures, it is very difficult to separate annotated samples from unannotated ones. The distinction between subjective/objective and negative/positive emotions is much easier. Finally, `information` and `instructions` classes are less distinguishable.

Emotion detection using transducers

We also used an approach based on extraction patterns to identify emotions in suicide notes. Given the limited amount of training data and the number of target classes, we chose to define these patterns manually, rather than trying to identify them automatically. These patterns combine surface-level tokens, lemmas and POS (part-of-speech) tags and are detected in texts using finite-state transducers, which automatically tag pattern occurrences in the input text.

We have manually developed one transducer for each class using UNITEX (http://igm.univ-mlv.fr/~unitex/),[16] which provides also with its base configuration a tokenizer, a POS tagger and a lemmatizer. The transducers were created by careful investigation of the training corpus. For instance, the transducer built for the love category is shown in Figure 3. It can identify expressions such as *I will always love you*, or *your loving husband*.

Each valid path in the graph represents an emotion-specific pattern, which is subsequently marked in the input text. Nodes in the transducer may correspond to sequences of surface tokens, lemmas with a given POS (eg, `<love.V>` for the verb "to love" and all its inflected forms) or POS tags (eg, `<.ADV>` for any adverb). As a consequence, the transducer is able to identify surface variants of the same pattern.

For the final classification, we applied all the transducers in a cascade, one after the other, in a specific order (*anger, love, abuse, blame, fear, forgiveness, guilt, happiness, hopefulness, hopelessness, pride, sorrow, thankfulness, information, instruction*). The order used for applying the transducers was determined on the

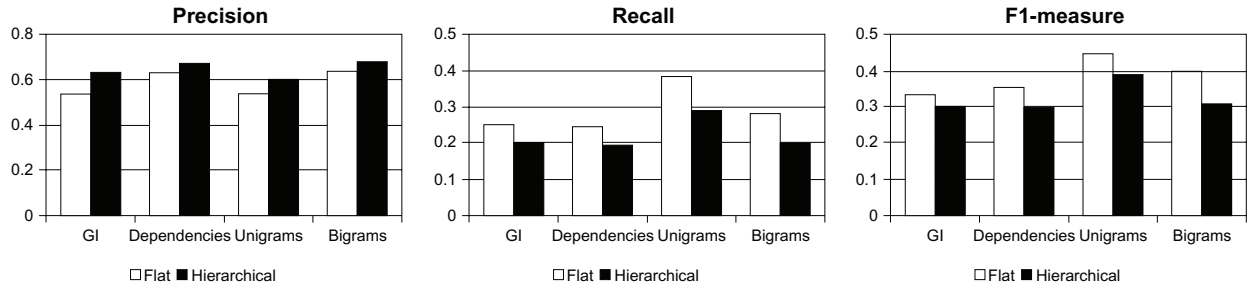

Figure 5. Hierarchical vs. flat classification performance (precision, recall and F1-measure).

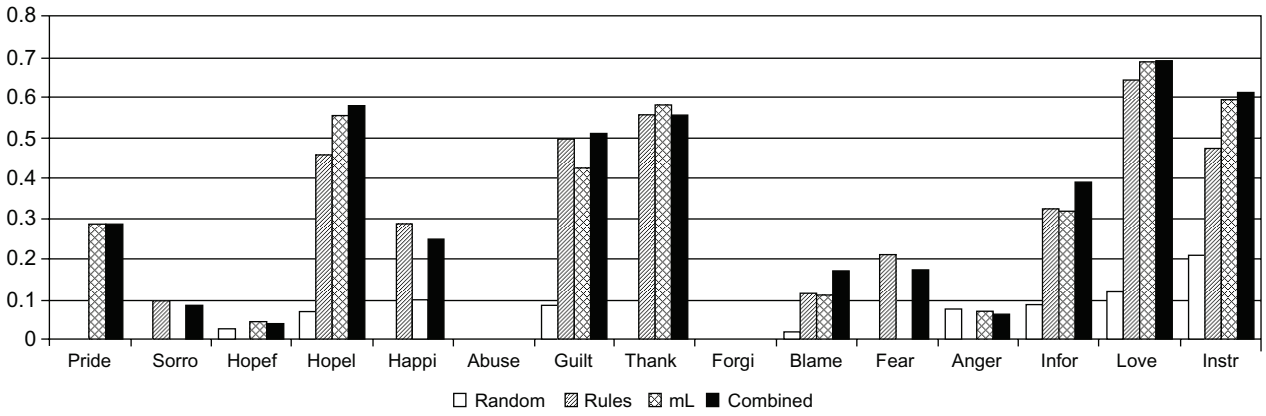

Figure 6. Performance of a random, rule-based, machine-learning, and combined systems across the classes.

training corpus, so as to avoid potential problems due to expressions which might be identified by several transducers. A sentence is labeled with a given category if at least one expression has been recognized by the corresponding transducer.

Experiments and Results

In order to tune the system parameters of the machine-learning component, we performed 10-fold cross validation on the training corpus. The task official performance measures are: micro-average precision/recall/F-measure. For our own purposes, we also calculated precision/recall/F-measure for each emotion category.

First, we analyzed the performance of the features used for emotion detection: GI, dependencies, unigrams, and bigrams. Figure 4 plots the classification F-measure of each emotion category and each feature using a flat classi-fication scheme. The classification performance of more frequent classes is higher than those of rarer ones: `love`, `thankfulness`, `hopelessness`, and `guilt` are much better classified than `blame`, `fear`, and `anger`. Moreover, `Pride`, `sorrow`, `hopefulness`, and `happiness` could be only detected with GI features, yet the performance is good. `Abuse` and `forgiveness`—the most rare classes in the corpus—are not detected by any features. As aforementioned, `information` and `instructions` classes are hardly distinguishable, which explains the low classification performance of the `information` and `instructions` classes, even though the later is the most frequent.

When performing hierarchical classification, we achieved 71% of accuracy on annotation detection, 84% on subjectivity detection, and 85% on polarity classification. The effect of the hierarchical classification is depicted on Figure 5. Micro-average precision/recall/F1-measure are presented for each feature. We can observe that precision augments when using hierarchical classification, but F1-measure drops due to the decrease of recall. To compensate this, we decided to use hierarchical classification with the mentioned features, but we added another classifier based on combination of unigrams and bigrams, which does a flat classification across all classes.

The final classification system consists of the rule-based component and the machine-learning based one. We present the classification performance of rule-based, machine-learning, and the combination of both systems on the evaluation set in Figure 6 (across the classes) and in Figure 7 (micro-average). A baseline random classifier was added for a comparison.

Official evaluations results

On the training corpus, the transducer-based system achieved a precision of 0.6033, a recall of 0.4873 and

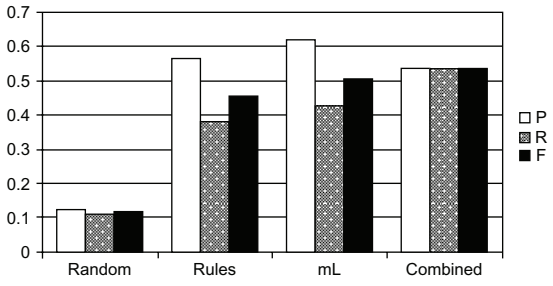

Figure 7. Micro-average performance of a random, rule-based, machine learning, and combined systems.

Table 3. Micro-average performance of a random, rule-based, machine learning, and combined systems

	Prec	Recall	F1
Random	0.123	0.110	0.116
Rules	0.566	0.380	0.455
mL	0.621	0.428	0.507
Combined	0.538	0.539	0.538

an F-measure of 0.5392. The results obtained on the test corpus were very closed from those obtained on the training corpus, with a 0.5383 global F-measure. This decrease in performance is mainly due to lower recall, as it is difficult to manually list all possible emotion-specific expressions. Another problem we have encountered with the data were the numerous spelling mistakes, which also lead to a lower recall, since transducers work with strict string equality. Nevertheless, those equivalent scores reveal the robustness of our system.

Conclusion

The emotion detection track of the i2b2/VA 2011 evaluation campaign is a difficult task due to the nature of the data and the specificity of the annotation schema. The LIMSI team has developed a system combining two approaches for emotion detection and classification: machine learning and rule-based approaches. On the official evaluation, we ranked 6th out of 26 participants with a 0.5383 global F-measure. As a future work, we would like to test our approach on other corpora, such as blogs or movie reviews, to see how well it generalizes on other domains.

Acknowledgements

This work was partially funded by project DoXA under grant number DGE no 08-2-93-0888 supported by the numeric competitiveness center CAP DIGITAL of Ile-de-France region.

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. Stone Philip J, Hunt Earl B. A computer approach to content analysis: studies using the general inquirer system. In: *Proc of the Spring Joint Computer Conference*, pages 241–256, New York, NY, 1963. ACM. doi: 10.1145/1461551.1461583.
2. Pestian John, Nasrallah Henry, Matykiewicz Pawel, Bennett Aurora, Leenaars Antoon. Suicide Note Classification Using Natural Language Processing: A Content Analysis. *Biomed Inform Insights*. 2010;(3):19–28.
3. Riloff Ellen, Wiebe Janyce. Learning extraction patterns for subjective expressions. In: *Proc of Empirical Methods in Natural Language Processing*. pages 105–112, Stroudsburg, PA, 2003.
4. Pang Bo, Lee Lillian. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: *Proc of ACL*. 2004.
5. Pak Alexander, Paroubek Patrick. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proc of LREC*, Valletta, Malta, 2010. European Language Resources Association (ELRA).
6. Pestian John, Matykiewicz Pawel, Linn-Gust Michelle, et al. Sentiment Analysis of Suicide Notes: A Shared Task. *Biomed Inform Insights*, 2012;5 (Suppl. 1):3–16.
7. Fan Rong-En, Chang Kai-Wei, Hsieh Cho-Jui, Wang Xiang-Rui, Lin Chih-Jen. LIBLINEAR: A Library for Large Linear Classification. 2008;9:1871–4.
8. Pang Bo, Lee Lillian, Vaithyanathan Shivakumar. Thumbs up?: sentiment classification using machine learning techniques. In *Proc of Empirical Methods in Natural Language Processing*, pages 79–86. Association for Computational Linguistics, 2002. doi: 10.3115/1118693.1118704.
9. Schmid Helmut. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proc of International Conference on New Methods in Language Processing*. 1994.
10. Bradley Margaret M, Lang Peter J. Affective Norms for English Words (ANEW).*University of Florida*, 1999.
11. Nakagawa Tetsuji, Inui Kentaro, Kurohashi Sadao. Dependency tree-based sentiment classification using CRFs with hidden variables. In: *Proc of NAACL*, HLT'10, pages 786–794, Los Angeles, CA, 2010. Association for Computational Linguistics.
12. Pak Alexander, Paroubek Patrick. Text representation using dependency tree subgraphs for sentiment analysis. In *Proc of the international conference on Database Systems For Advanced Applications*, pages 323–332, Berlin, Heidelberg, 2011. Springer-Verlag.
13. de Marneffe Marie-Catherine, Maccartney Bill, Manning Christopher D. Generating Typed Dependency Parses from Phrase Structure Parses. In: *Proc of LREC*, 2006.
14. de Marneffe Marie-Catherine, Manning Christopher D. *Stanford typed dependencies manual*, 2008. URL http://nlp.stanford.edu/software/dependencies_manual.pdf.
15. Yang Dan and Lee Won-Sook. Music Emotion Identification from Lyrics. In: *Proc of the IEEE International Symposium on Multimedia*, pages 624–629, Washington, DC, 2009. doi: 10.1109/ISM.2009.123.
16. Paumier Sébastien. *Unitex 2.1 user manual*, 2011. URL http://igm.univ-mlv.fr/˜unitex/.