Biomedical Informatics Insights

ORIGINAL RESEARCH

Labeling Emotions in Suicide Notes: Cost-Sensitive Learning with Heterogeneous Features

Jonathon Read, Erik Velldal and Lilja Øvrelid

Department of Informatics, University of Oslo, Norway. Corresponding author email: jread@ifi.uio.no

Abstract: This paper describes a system developed for Track 2 of the 2011 Medical NLP Challenge on identifying emotions in suicide notes. Our approach involves learning a collection of one-versus-all classifiers, each deciding whether or not a particular label should be assigned to a given sentence. We explore a variety of features types—syntactic, semantic and surface-oriented. Cost-sensitive learning is used for dealing with the issue of class imbalance in the data.

Keywords: emotion classification, suicidology, support vector machines, cost-sensitive learning

Biomedical Informatics Insights 2012:5 (Suppl. 1) 99–103

doi: 10.4137/BII.S8930

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

Introduction

This paper presents a survey of the utility of various types of features for supervised training of Support Vector Machine (SVM) classifiers to determine whether sentences from suicide notes bear certain emotions, or if they communicate instructions or information. The work described in this paper was conducted in the context of Track 2 of the 2011 Medical NLP Challenge.[1] The task organizers provided developmental data consisting of 600 suicide notes, comprising 4,241 (pre-segmented) sentences with a total of 79,498 (pre-tokenized) words. Each sentence is annotated with any number of the 15 topic labels (as listed in Table 1). For evaluation purposes the organizers provided an additional set of 1,883 (initially unlabeled) sentences in 300 notes for held-out testing.

Our approach involves learning a collection of binary SVM classifiers, where each classifier decides whether or not a particular label should be assigned to a given sentence. The information sources explored in feature design range from simple bag-of-words features and *n*-grams over stems, to features based on syntactic dependency analysis and WordNet synonym sets. We also describe how so-called cost-sensitive learning is used for dealing with the problem of imbalanced numbers of positive and negative examples in the data.

Method

Our approach to the suicide notes labeling task utilizes a collection of *one-versus-all* automatically-learned classifiers. One-versus-all classifiers are a common solution for multi-class problems,[2] where the problem is reduced to multiple independent binary classifiers. For each label we train a linear sentence classifier using the SVM^{light} toolkit.[3]

As training data for each classifier, we use the set of all sentences annotated with the label as positive examples; the sentences in the set complement form the negative examples. We note, however, that the frequency distributions of the labels in the suicide notes vary greatly. For example, the most frequent class (INSTRUCTIONS) is applied to 19% of sentences, whereas the least frequent class (FORGIVENESS) occurs in only 0.1%. So for each classifier the negative examples will greatly outnumber positive examples. A well-known approach for improving classifier performance in the face of such skewed class distributions is the notion of *cost-sensitive learning*. In SVM^{light} this is accomplished using *unsymmetric*

Table 1. Optimal feature sets and cost-balance (*j*) parameters for each label, as observed in the development data set using ten-fold cross-validation.

Label	Features	Cost (*j*)	Prec	Rec	F_1
ABUSE	mas	50	0.17	10.00	0.33
ANGER	bos + sas	90	6.64	10.97	7.83
BLAME	bos + wns	15	17.02	27.05	19.16
FEAR	sas	5	10.00	10.00	10.00
FORGIVENESS	mas + wns	9	5.00	10.00	6.67
GUILT†	pos + wns	5	44.36	51.65	46.90
HAPPINESS/PEACEFULLNESS	bos + sas	150	19.17	21.43	18.32
HOPEFULNESS	bos	25	15.62	29.02	18.82
HOPELESSNESS†	big + bos + wns	6	54.56	55.37	54.07
INFORMATION†	dep + pos + wns	8	46.34	49.50	46.41
INSTRUCTIONS†	big + bos + dep + pos	3	69.27	66.40	67.32
LOVE†	big + bos + dep + pos	2	76.19	67.80	71.23
PRIDE	mas + wns	15	5.00	5.00	5.00
SORROW	mas + wns	5	12.33	11.36	10.37
THANKFULNESS†	bos + wns	4	69.47	69.44	67.77
micro-average (total)			46.00	54.00	49.41
micro-average†			61.09	51.71	55.81

Notes: Only the classifiers for labels marked with † are included in our final setup. While the scores listed as micro-average† are computed only for these labels, the total micro-averages are based on all labels.
Abbrevations: The feature types are: *big,* bigrams over stems; *bos,* bag-of-stems; *dep,* sentence dependency patterns; *mas,* maximum association score; *pos,* parts-of-speech; *sas,* sum of association scores; *wns,* WordNet synsets.

cost factors,[4] such that different penalties are assigned to false positives and false negatives.

Sentences are represented by a variety of features that record both surface and syntactic characteristics, as well as semantic information from external resources, as described below.

The most basic features we employ describe the surface characteristics of sentences. These include:

- The stem form of words, obtained using the implementation of the Porter Stemmer[5] in the Natural Language Toolkit[6] (eg, *happy*, *happiness*, *happily*, etc. all activate the stem feature *happi*).
- Bigrams of stems, created from pairs of stems appearing in sequence (eg, *happy days* activates the bigram feature *happi day*).
- Lexicalized part-of-speech, formed of word stems concatenated with the PoS assigned by TreeTagger.[7]

Features based on syntactic dependency analysis provide us with a method for abstracting over syntactic patterns in the data set. The data is parsed with the Maltparser system, a language-independent system for data-driven dependency parsing.[8] We train the parser on a PoS-tagged version of the Wall Street Journal sections 2–21 of the Penn treebank, using the parser and learner settings optimized for the Maltparser in the CoNLL-2007 Shared Task. The data was converted to dependencies using the Pennconverter software[9] with default settings—see Figure 1 for an example. From these dependencies we extract:

- Sentence dependency patterns; wordform, lemma, PoS of the root of the dependency graph, eg, (`leave, leave, VV`), and patterns of dependents from the (derived) root, expressed by their dependency label, (eg, `VC-OBJ-OPRD`), part-of-speech (`VV-NN-VVD`) or lemma (`leave-door-unlock`).
- Dependency triples; the set of labeled relations between each head and dependent, eg, (`will-SBJ-I`, `will-VC-leave`, `leave-OPRD-unlocked`).

We also draw on semantic information from external resources:

- Synonym set features are generated using WordNet,[10] by mapping word forms and their predicted part-of-speech to the first synset identifier (eg, the adjectives *distraught* and *overwrought* both map the the synonym set feature 00086555).
- WordNetAffect[11] is used similarly, activating features that represent emotion classes when member words are observed in sentences (eg, the words *wrath* or *irritation* both activate the WordNetAffect feature *anger*).

The final type of feature that we employ represents the degree to which each stem in a sentence is associated with each label, as estimated from the training data using the *log odds ratio*. In order to incorporate this information in the classifier, we add features that record the following for each sentence:

- The sum of the association scores of all words in a given sentence towards each label.
- Boolean features indicating which label had the maximum association score.

Model tuning

For system tuning we performed a grid search of parameters for each classifier, evaluating different permutations of feature combinations. In parallel we also tuned the unsymmetric cost factors, drawing values from logarithmic intervals. Each model configuration was tested by ten-fold cross-validation on the development data (partitioning on the note-level), and for each label we then selected the combination of feature types and cost factor that resulted in the highest F_1.

Results

The cross-validated micro-averaged results on the development data are: Precision = 46.00, Recall = 54.00, F_1 = 49.41. Table 1 lists details of the results of our model tuning procedure. We note that

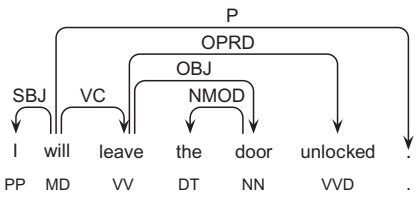

Figure 1. Example dependency representation.

the optimal configuration of features varies from label to label. However, while stems and synonym sets are often in the optimal permutation, dependency triples and features from WordNetAffect do not occur in any configuration.

We note that the unsymmetric cost factor enabled us to improve recall for many classes but this often came at a cost in terms of precision. While this typically lead to increased F_1s for individual labels, the effect on the overall micro-averaged F_1 was negative. We found that this was due to poor precision on the infrequent labels.

In the end, therefore, we only attempt to classify the six labels that we can predict most reliably—GUILT, HOPELESSNESS, INFORMATION, INSTRUCTIONS, LOVE and THANKFULNESS—and make no attempt on the remaining labels. In the development data this increases overall system performance in terms of the micro-average scores: Precision = 61.09, Recall = 51.71, F_1 = 55.81. However, it should be noted that this decision is motivated by the fact that micro-averaging is used for the shared task evaluation. Micro-averaging emphasizes performance on frequent labels, whereas macro-averaging would encourage equal performance across all labels.

Table 2 describes the performance on the held-out evaluation data when training classifiers on the entire development data set, with details on each label attempted by our setup. As described above, we only apply classifiers for six of the labels in the data set (due to the low precision observed in the development results for the remaining nine labels). We find that the held-out results are quite consistent with those predicted by cross-validation on the development data. The final micro-averaged F_1 is 54.36, a drop of only 1.45 compared to the development result.

Table 2. Held-out evaluation results.

Label	Prec	Rec	F_1
GUILT	48.72	48.72	48.72
HOPELESSNESS	55.13	56.33	55.72
INFORMATION	37.41	50.00	42.80
INSTRUCTIONS	72.14	60.99	66.10
LOVE	77.99	61.69	68.89
THANKFULNESS	50.79	71.11	59.26
micro-average	60.58	49.29	54.36

Note: The labels that are not attempted are not listed in the table (Prec = Rec = 0).

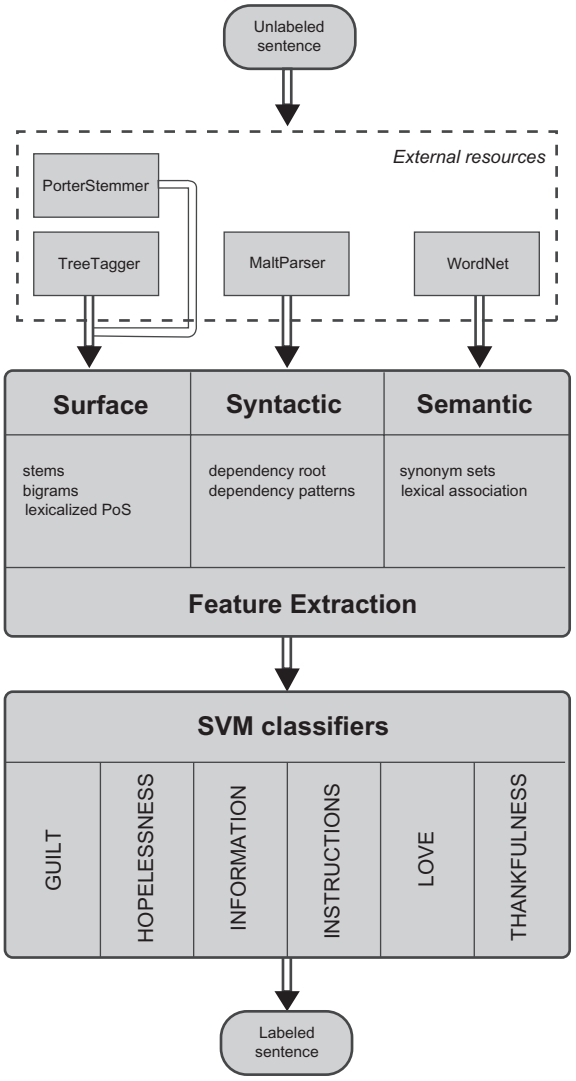

Figure 2. Final system architecture.

Conclusions

Our approach to the shared task on topic classification of sentences from suicide notes is summarized in Figure 2. Using a variety of external resources, we represented sentences using a diverse range of surface, syntactic and semantic features. We used these representations to train a set of binary support vector machine classifiers, where each classifier is responsible for determining whether or not a label applies to a given sentence. We also experimented with unsymmetric cost factors to handle problems with the skewed distribution of positive and negative examples in the data sets. We performed a grid search of hyper-parameters for each classifier to find optimal combinations of feature types and unsymmetric cost factors.

In future work we will optimize the parameters for each classifier with respect to the overall F_1 (rather than the label F_1, as described in this paper). We will also investigate how the performance for labels with few examples may be boosted by drawing information from large amounts of unlabeled text. For example, estimating the semantic similarity of words with prototypical examples of a label using measures of lexical association or distributional similarity can be informative when labeling text with sentiment or emotion.[12] We will experiment with this approach, both as a stand-alone technique, and by including its prediction in features for supervised classifiers.

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. Duan KB, Keerthi SS. Which is the best multiclass SVM method? An empirical study. In: *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, 2005.
3. Joachims T. Making large-scale SVM learning practical. In: Scholkopf B, Burges C, Smola A, editors, *Advances in Kernel Methods—Support Vector Learning*. MIT Press, 1999.
4. Morik K, Brockhausen P, Joachims T. Combining statistical learning with a knowledge-based approach—a case study in intensive care monitoring. In: *Proceedings of the* 16*th International Conference on Machine Learning*, Bled, Slovenia, 1999.
5. Porter MF. An algorithm for suffix stripping. *Program*, 1980:14(3).
6. Bird S, Loper E. NLTK: The natural language toolkit. In: *Proceedings of the ACL demonstration session*, 2004.
7. Schmid H. Probabilistic part-of-speech tagging using decision trees. In: *Proceedings of the International Conference on New Methods in Language Processing*, 1994.
8. Nivre J, Nilsson J, Hall J, Eryiğit G, Marinov S. Labeled pseudo-projective dependency parsing with Support Vector Machines. In: *Proceedings of the Conference on Computational Natural Language Learning*, 2006.
9. Johansson R, Nugues P. Extended constituent-to-dependency conversion for English. In: *Proceedings of the* 16*th Nordic Conference of Computational Linguistics*, 2007.
10. Fellbaum C, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
11. Strapparava C, Valitutti A. Wordnet-affect: an affective extension of wordnet. In: *Proceedings of the Fourth International Conference of Language Resources and Evaluation*, Lisbon, 2004.
12. Read J, Carroll J. Weakly supervised techniques for domain-independent sentiment classification. In: *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*, 2009.