A Semi-supervised Approach for the CLPsych 2016 Shared Task

Nicolas Rey-Villamizar and **Prasha Shrestha** and **Thamar Solorio**
University of Houston
nrey@uh.edu, pshrestha3@uh.edu, solorio@cs.uh.edu

Farig Sadeque and **Steven Bethard**
University of Alabama at Birmingham
farigys@uab.edu, bethard@uab.edu

Ted Pedersen
University of Minnesota, Duluth
tpederse@d.umn.edu

Abstract

The 2016 CLPsych Shared Task is centered on the automatic triage of posts from a mental health forum, au.reachout.com. In this paper, we describe our method for this shared task. We used four different groups of features. These features are designed to capture stylistic and word patterns, together with psychological insights based on the Linguistic Inquiry and Word Count (LIWC) word list. We used a multinomial naive Bayes classifier as our base system. We were able to boost the accuracy of our approach by extending the number of training samples using a semi-supervised approach, labeling some of the unlabeled data and extending the number training samples.

1 Introduction

The 2016 ACL Workshop on Computational Linguistics and Clinical Psychology (CLPsych) included a shared task focusing on classification of user posts in the mental health forum, au.reachout.com. Our system is based on two main ideas: the use of word lists that group words into psychologically meaningful categories, and a semi-supervised approach in order to increase the size of the training data. For the word list we used, the Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2007). LIWC is a psychometrically validated lexicon mapping words to psychological concepts and has been used extensively to examine language in order to understand mental health. For using some of the unlabeled data to train our system we leveraged the idea of self-training. This method consists of expanding the number of label samples from the unlabeled data by using the most confident samples, based on a pretrained system on the label data. We were able to combine these two ideas and develop a system that performs significantly better than the baselines.

2 Task Description

The 2016 CLPsych Shared Task is based on the automatic classification of user posts from an online mental health forum *ReachOut*[1] into four different categories according to how urgently the post needs a moderator's attention.

For the shared task, a corpus of posts tagged with four different categories *crisis/red/amber/green* has been provided. Table 1 describes each of the different categories. A dataset of unlabeled data was also provided. Table 2 describes the number of samples of both the labeled and unlabeled data as well as the test data.

The evaluation metric of the task is a macro-averaged F-score over *crisis*, *red* and *amber* labels. This was motivated by a system needing to get the critical cases correct.

3 System description

In our system, we used a Multinomial naive Bayes classifier together with features that aim to capture the user's cognitive processes and writing style. We used a cross-validation approach in combination with a Bayesian optimization for the parameter selection using the provided training set.

[1] http://www.au.reachout.com

Proceedings of the 3rd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 171–175,
San Diego, California, June 16, 2016.

Post's label	Description
crisis	The author (or someone they know) might hurt themselves or others (these are red instances that are of immediate importance).
red	A moderator needs to look at this ASAP and take action.
amber	A moderator needs to look at this and assess if there is enough responses and support from other or if they should reply.
green	A moderator does not need to prioritize addressing this post.

Table 1: Categories of the post

Data	Description
Train set	39 *crisis*, 110 *red*, 249 *amber*, 549 *green* posts
Test set	1 *crisis*, 27 *red*, 47 *amber*, 166 *green* posts
Unlabeled set	63797 posts

Table 2: Data distribution

3.1 Classifier

We explored different classifiers in our experimentation. Based on a cross-validation study on the training set we choose to use Multinomial naive Bayes for our final submission. We used the implementation of the *scikit-learn*[2] module (Pedregosa et al., 2011). To account for the words not present in the training vocabulary we explore the use of different smoothing parameters. Using a smoothing parameter of 1 corresponds to the classic Laplace smoothing, and values below 1 correspond to Lidstone smoothing.

3.2 Features

We used the following features in our system:

- Unigrams and bigrams of words
- Prefixes and suffixes of lengths 2, 3, 4 and 5
- Number of kudos in the post
- For each category of the LIWC word lists, we counted how many occurrences of each word in the list the post has, and we created a vector representation for each post. The LIWC 2007 word list has 64 different word categories.

The unigram and bigram features are intended to capture writing patterns of words that are associated with each label. For example, unigrams and bigrams such as *harm*, *overwhelmed*, *hurts*, and *can't handle* are usually associated with negative feelings that we want our system to be able to capture as *red* and *crisis* labels. The same happens with positive words that are more typically associated with the *green* label.

The number of kudos of the post was used to better distinguish positive posts from the others. In general, posts labeled as *green* have more kudos than the rest. Prefixes and suffixes are added since they have shown to perform well in many text classification tasks.

3.3 Parameter Optimization

We used the Bayesian optimization framework provided by SigOpt[3]. This framework is an alternative to the classic grid search approach, where parameters are explored in an exhaustive way. Table 3 describes the ranges of values explored for the classifier. We also tested the same set of parameters with a different combination of features. We found that using trigrams decreased the performance as well as using more than five character prefixes or suffixes as a feature.

Parameter	Range of values
Smooth term(α)	(1, 0.8, 0.4, 0.2, 0.1)
Class weight	exhaustive search of 10% increase for each class

Table 3: Parameter exploration for the classifier

We also explored feature selection algorithms. However in the 8-fold cross validations over the training set that we performed none of them gave us better performance than when all the features were used.

4 Self-training

Self-training is a method to expand the number of labeled samples given the high cost of labeling samples in the text processing domain (Nigam et al., 1998). We optimized our system in order to achieve the maximum possible f1-macro-average that is used as the

[2] scikit-learn.org/

[3] https://sigopt.com/

official score using an 8-fold cross validation on the training dataset. We ranked each system as the mean over all the f1-macro-average of the three classes of the 8 runs. We then ran our algorithm in the unlabeled data and selected the most confident samples for each class. The confidence was measured based on the posterior probability of the Multinomial naive Bayes classifier. In order to keep the class balanced in the same way as the training data, we selected only 100 samples in this way 4 *crisis*, 11 *red*, 26 *amber*, and 59 *green*. In our experimentation with the 8-fold cross-validation of the training set, including the samples found by self-training improved the f1-macro average of our system by 0.12. It also helped to extend the vocabulary of some of the words not present in the training samples.

5 Results

In this section, we present the results of our system. We used two baseline systems. The first baseline consists of random assignment of labels with any of the three classes *crisis*/*red*/*amber*. The second is a majority class, always predicting *amber*. The first baseline achieves a macro average f1-score of 0.11, and the second system achieves a macro average f1-score of 0.10.

5.1 Official results

Our system results are summarized in Table 5 and Table 4, and the overall official statistics of all the teams submissions are summarized in Table 6. From the precision and recall results of Table 4 we can conclude that our system was balanced in terms of achieving a similar precision and recall for each one of the classes. The system incorrectly assigned three posts a *crisis* label and was not able to predict the only *crisis* post present in the test data. This post in particular contained vocabulary not seen in the training set, which made it difficult for our system to detect it correctly, instead our system assigned it a *red* label.

Our system performed a little above the median of all the team best scores with a 0.34 official score. Our system would require an increase 0.08 in the f1-average-macro to score as the best participant. In the non-*green* vs. *green* macro f-score and the non-*green* vs. *green* accuracy we performed above the median of the team bests. It is important to mention that the selected metric is very sensitive to the *crisis* label. If the *crisis* post was labeled correctly, the official score would have increased to around 0.50.

label	precision	recall	f1-score
crisis	0.00 (0/3)	0.00 (0/1)	0.00
red	0.46 (11/24)	0.41 (11/27)	0.43
amber	0.53 (30/57)	0.64 (30/47)	0.58

Table 4: Precision, recall, and f1-score of our system for the three classes used for the official score.

Measurement	Our Score
official score (f1-macro)	**0.34**
accuracy	0.77
non-*green* vs. *green* macro f-score	0.79
non-*green* vs. *green* accuracy	0.86
random *crisis*/*red*/*amber* (f1-macro)	0.11
all *amber* (f1-macro)	0.10

Table 5: Official results of our system together with baseline 1 and 2

Measurement	min.	max.	median of team bests
official score (f1-macro)	**0.13**	**0.42**	**0.335**
accuracy	0.42	0.85	0.775
non-*green* vs. *green* macro f-score	0.58	0.87	0.77
non-*green* vs. *green* accuracy	0.60	0.91	0.85

Table 6: Official statistics of the overall results

5.2 Analysis and discussion

The most difficult part of the shared task was the highly skewed distribution of the training samples. The smallest class, *crisis*, has 39 samples and the largest class, *green*, has more than 500 samples. We assumed the distribution of each class to be representative of the distribution of the whole population. If more information can be known a priori about the class distribution, our system could be adjusted to model such a distribution. During the cross-validation study of the training samples, we found that distinguishing between the *red* and the *crisis* class was the most challenging part of the problem.

We found that sometimes even for a human it is difficult to distinguish between one or the other, given the informal language used in the online posts.

In order to understand the types of posts present in the unlabeled data, we ran our self-training algorithm multiple times to understand how it will be biased towards the classes and to get familiar with data. We found that in the forum there were some particular threads where users tend to post very negative posts. We found that many of the posts in this thread were either *crisis* or *red*. We performed a study to replace the given training sample with some of these posts and study the mean performance in an 8-fold cross validation. We found that the performance was lower. In particular, those posts were structured in a specific way, people will post something very positive, followed by something very negative. This structure of the post was very challenging for our system. Either the posts were assigned to *green* or *crisis* label depending on the data present in each fold of the cross-validation iteration.

From the gold data, we could see that most of the errors of our system were due to new vocabulary not present in the training set. We tried to account for this with the use of a smoothing parameter in the classifier but more work is needed in this respect. One way could be to train a word embedding using the unlabeled data in such a way that semantic similarities of words not present in the training samples can be modeled in the test set.

6 Related work

In the previous versions of the workshop some systems have been proposed to solve similar challenging problems using some or similar features to the ones we used in our system. In (Mitchell et al., 2015) a system was developed for quantifying the language of schizophrenia in social media based on the LIWC lexicon. This study also showed that character n-grams over specific tweets in the user's history can be used to separate schizophrenia sufferers from a control group. In (Pedersen, 2015) a system based on decision lists was developed to identify Twitter users who suffer from Depression or Post Traumatic Stress Disorder (PTSD). The features in this system are based on n-grams of up to 6 words. In this system, the usage of larger n-grams performed better than bigrams. In our experiments, we only tried with n-grams up to length 3 and found that the best performing system in the cross-validation of the training data was obtained using bigrams.

7 Conclusion

In this paper, we have briefly described our submission to the CLPsych 2016 shared task. We found that the best result was achieved when the number of label samples was expanded by using a self-training approach. We also saw that the performance of the system degraded when some challenging posts with both very positive and negative information were included. We also used a method for parameter tuning that accelerated our experimentation significantly as compared with the exhaustive grid search algorithm and we expect this to be useful for other researchers in the field.

In future work, we plan to study the use the unlabeled data to extend the vocabulary and in this way help us model words not present in the training sample. We also plan to do a more exhaustive experimentation on different algorithms to label the unlabeled data to increase the amount of training data used to train our system. Finally, we expect to study in more detail how the pattern of posts over a period of time can be used to predict the likelihood of a user to post a *crisis* or *red* kind of post.

Acknowledgments

We thank the organizers of this shared task for their effort towards building a community able to solve this challenging problem.

References

Margaret Mitchell, Kristy Hollingshead, and Glen Coppersmith. 2015. Quantifying the Language of Schizophrenia in Social Media. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 11–20. North American Chapter of the Association for Computational Linguistics.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. 1998. Learning to classify text from labeled and unlabeled documents. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, page 792799.

Ted Pedersen. 2015. Screening Twitter Users for Depression and PTSD with Lexical Decision Lists. In *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 46–53. North American Chapter of the Association for Computational Linguistics.

Fabian Pedregosa, Olivier Grisel, Ron Weiss, Alexandre Passos, and Matthieu Brucher. 2011. Scikit-learn : Machine Learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. 2007. *The Development and Psychometric Properties of LIWC2007 The University of Texas at Austin*.