The UMD CLPsych 2016 Shared Task System: Text Representation for Predicting Triage of Forum Posts about Mental Health

Meir Friedenberg, Hadi Amiri, Hal Daumé III, Philip Resnik
Department of Computer Science,UMIACS
University of Maryland, College Park
meir@terpmail.umd.edu, {hadi,hal3f,resnik}@umd.edu

Abstract

We report on a multiclass classifier for triage of mental health forum posts as part of the CLPsych 2016 shared task. We investigate a number of document representations, including topic models and representation learning to represent posts in semantic space, including context- and emotion-sensitive feature representations of posts.

1 Introduction

The 2016 CLPsych Shared Task focused on automatic triage of posts from ReachOut.com, an anonymous online mental health site for young people that permits peer support and dissemination of mental health information and guidance. Peer support and volunteer services like ReachOut, Koko,[1] and Crisis Text Line[2] offer new and potentially very important ways to help serve mental health needs, given the challenges many people face in obtaining access to mental health providers and the astronomical societal cost of mental illness (Insel, 2008). In such settings, however, it is essential that moderators be able to quickly and accurately identify posts that require intervention from trained personnel, e.g., where there is potential for harm to self or others. This shared task aimed to make progress on that problem by advancing technology for automatic triage of forum posts. In particular, the task involved prediction of categories for ReachOut posts, with the four categories, {*crisis*, *red*, *amber*, *green*}, indicating how urgently the post needs attention.

[1] itskoko.com
[2] crisistextline.org

2 Systems Overview

Following Resnik et al. (2015), the core of our system is classification via multi-class support vector machines (SVMs) with a linear kernel. We explore topic models as well as context- and emotion-sensitive representations of posts, together with baseline bag of words representations, as features for our model.

2.1 Baseline Lexical Features

We considered bag of words and bag of bigrams in conjunction with TF-IDF and binary weighting schemes of these represenations and stopword removal. Our preliminary experiments with development data suggested that binary weighted bag of words features with stopword removal were an effective baseline; we refer to this feature set simply as BOW.

2.2 Topic Models

We use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to create a 30-topic model on the entire ReachOut corpus (including labeled, unlabeled, and test data), as well as posts from the Reddit.com /r/Depression forum, yielding document (forum post) topic probability posteriors as features. The inclusion of the test data among the inputs to LDA can be thought of as a transductive approach to model generation for this shared task aiming to take maximal advantage of available data, although this would prevent post-by-post processing in a real-world setting.

Proceedings of the 3rd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 158–161,
San Diego, California, June 16, 2016.

System #	Description
1	BOW
2	BOW + Context-Sensitive Representations
3	Emotion-Sensitive Representations (Euclidean distance)
4	BOW + Topic Posteriors (LDA)
5	BOW + Topic Posteriors + Emotion-Sensitive Representations (Cosine similarity)

Table 1: System Features and Runs

2.3 Context-Sensitive Representation

We obtain context-sensitive representations of an input post by concatenating the average word embedding of the input post with its "context" information (represented by low dimensional vectors) and passing the resulting vector to a basic autoencoder (Hinton and Salakhutdinov, 2006). We obtain context vectors for posts via non-negative matrix factorization (NMF) where the disttribution of an input post over the topics in the dataset is used as its context vector. We use the pre-trained 300-dimensional word embeddings provided by `Word2Vec`.[3]

Formally, we use NMF to identify context information for input posts as follows. Given a training dataset with n posts, i.e., $\mathbf{X} \in \mathbb{R}^{v \times n}$, where v is the size of a global vocabulary and the scalar k is the number of topics in the dataset, we learn the topic matrix $\mathbf{D} \in \mathbb{R}^{v \times k}$ and a context matrix $\mathbf{C} \in \mathbb{R}^{k \times n}$ using the following sparse coding algorithm:

$$\begin{aligned} \min_{\mathbf{D},\mathbf{C}} \quad & \|\mathbf{X} - \mathbf{D}\mathbf{C}\|_F^2 + \mu\|\mathbf{C}\|_1, \qquad (1) \\ s.t. \quad & \mathbf{D} \geq 0,\ \mathbf{C} \geq 0, \end{aligned}$$

where each column in $\mathbf{C}$ is a sparse representation of an input over all topics and can be used as context information for its corressponding input post. Note that we obtain the context of test instances by transforming them according to the fitted NMF model on training data. We believe combining test and training data (as discussed above) will further improve the quality of our context vectors.

We concatenate the average word embedings and context vectors of input posts and pass them to a basic deep autoencoder (Hinton and Salakhutdinov, 2006) with three hidden layers. The hidden representations produced by the autoencoder will be used as context-sensitive representations of inputs and considered as features in our system.

[3] `code.google.com/p/word2vec`.

2.4 Emotion-Sensitive Representation

The emotion-sensitive representation of an input post is obtained by computing the distance (Euclidean distance or cosine similarity) between the average word embedding of the input post with nine categories of emotion words. The emotion categories that we consider are

> *anger*, *disgust*, *sadness*, *fear*, *guilt*, *interest*, *joy*, *shame*, *surprise*,

where each category has a designated word, e.g. "anger", and its 40 nearest neighbor words in embedding space according to Euclidean distance. For example, the category for *anger* contains "anger" along with related words like "resentment", "fury", "frustration", "outrage", "disgust", "indignation", "dissatisfaction", "discontentment", etc.[4] Using the Euclidean distance or cosine similarity between average word embedding of the input post with the embedding of each emotion word yields 311 features for the classifier, one per emotion-word category ignoring the emotion words that were removed.

2.5 Classifier Details

In our experiments we used multi-class SVM classifiers with a linear kernel. Specifically, we used the python scikit-learn module (Pedregosa et al., 2011), which interfaces with the widely-used libsvm.[5] We employed a one-vs-one decision function, and used the 'balanced' class_weight option to set class weights to be inversally proportional to their frequency in the training data.[6] All other parameters were set to their default values.

[4] We also manually verified the nearest neighbor words to ensure that they correctly represent their corresponding categories, and remove words that appear in at least two categories with opposite sentiment orientation.

[5] scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[6] One-vs-one beat one-vs-all in preliminary experimentation.

Specific feature combinations for our systems are reported in Table 1 and were selected based on development data. While our main criterion for choosing what features to use was Macro-Averaged F-Score, System 3 (emotion-sensitive representations) was selected primarily because of its superior performance on red prediction. Given the importance of red and crisis prediction in this context, we found this system interesting and consider its relative success at red prediction to be worthy of further exploration.

2.6 Data Preparation

Preprocessing: We performed the same basic preprocessing on all posts, including removing URLs and non-ascii characters, unescaping HTML, and expansion of contractions. We also lemmatized the tokens.

Data Splits: As per the suggestion in the shared task description, we set aside the last 250 posts of the training data as development data. Our primary use of the development data was in system development and selecting feature combinations. We also removed one post each from the training and development data as they did not appear to us to have significant linguistic content.

3 Results

Tables 2 and 3 show the performance of our submitted systems on development and test data respectively. Table 4 presents the effects of different feature combinations on development data performance, which we used to select our systems for submission.

Test data performance is noticeably worse for all five of our systems than development data performance. A non-negligible part of that seems to be our performace on crisis recall - the fact that there is only one crisis post in the test data set implies that when our system incorrectly labels that post an F-Score of 0 is necessarily averaged in. Evaluating why all five of our systems predict a green label for the crisis post seems like a worthwhile line of inquiry towards improving upon our system. We will conduct such experiments in the future.

F-Score\System	**1**	**2**	**3**	**4**	**5**
Green	0.82	0.85	0.78	0.83	0.82
Amber	0.57	0.54	0.42	0.54	0.51
Red	0.39	0.44	0.52	0.40	0.42
Crisis	0.33	0.25	0.24	0.33	0.35
Macro-Averaged	**0.53**	**0.52**	**0.49**	**0.52**	**0.52**
Official Score	**0.43**	**0.41**	**0.39**	**0.42**	**0.43**

Table 2: F-scores on development data. (Official Score is Macro-Averaged F-Score over crisis, red, and amber.)

F-Score\System	**1**	**2**	**3**	**4**	**5**
Green	0.83	0.87	0.84	0.83	0.85
Amber	0.41	0.5	0.33	0.43	0.48
Red	0.47	0.44	0.4	0.48	0.44
Crisis	0.00	0.00	0.00	0.00	0.00
Macro-Averaged	**0.43**	**0.45**	**0.39**	**0.44**	**0.44**
Official Score	**0.29**	**0.31**	**0.24**	**0.30**	**0.31**

Table 3: F-scores on test data. (Official Score is Macro-Averaged F-Score over crisis, red, and amber.)

Our system #3, which used Euclidean distance based emotion-sensitive representation of documents, was submitted because of its outstanding red prediction performance on development data. Given the importance of red and crisis recall in this domain, a system that perfomed particularly well in such an area seems worth exploring. Unfortunately, this red recall rate did not carry over to the test data, so it seems likely that our model simply overfit to the red data.

An examination of Table 4 suggests that it may be difficult to find features that are significantly more effective for this task than bag of words features. In particular, all of the systems listed that outperformed bag of words overall (whether on Macro-Averaged F-Score or Macro-Averaged F-Score over the amber, red, and crisis classes) seem to have done so only minimally. Interestingly, many of the feature sets did outperform bag of words on F-Score for the red class in development data, but this result does not seem to replicate in the test data.

4 Conclusions and Future Directions

In this paper we have summarized our contribution to the CLPSych 2016 shared task on triage of mental health forum posts. Our approach used class-weighted multi-class SVM classifiers with a linear kernel, and we found binary bag of words features to

Features\F-Scores	Green	Amber	Red	Crisis	Macro-Averaged	Official Score
BOW	0.82	**0.57**	0.39	0.33	**0.53**	0.43
Topics	0.77	0.37	0.34	0.24	0.43	0.32
Context Sensitive	0.80	0.43	0.28	0.32	0.46	0.34
Emotion Sensitive (Euclidean)	0.78	0.42	**0.52**	0.24	0.49	0.39
Emotion Sensitive (Cosine)	0.67	0.19	0.28	0.07	0.31	0.18
BOW + Topics	0.83	0.54	0.40	0.33	0.52	0.42
BOW + Context Sensitive	**0.85**	0.54	0.44	0.25	0.52	0.41
BOW + Emotion Sensitive (Euclidean)	0.82	0.45	0.45	0.13	0.46	0.34
BOW + Emotion Sensitive (Cosine)	0.82	0.54	0.42	**0.35**	**0.53**	**0.44**
BOW + Topics + Context Sensitive	0.84	0.52	0.44	0.25	0.51	0.40
BOW + Topics + Emotion Sensitive (Euclidean)	0.82	0.45	0.45	0.125	0.46	0.34
BOW + Topics + Emotion Sensitive (Cosine)	0.82	0.51	0.42	**0.35**	0.52	0.43
All	0.82	0.44	0.45	0.13	0.46	0.34

Table 4: Multi-class F-scores of different feature combinations on development data. (Official Score is Macro-Averaged F-Score over crisis, red, and amber.)

be reasonably effective for this task. Though topic models and context- and emotion-sensitive vector representations did not perform well independently on this task, when used to supplement bag of words features they did lead to some improvement in test data prediction.

In future work, one direction for potential improvement is the exploration of more complex topic models. In particular, our work utilized "vanilla" Latent Dirichlet Allocation, but Resnik et al. (2015) found some success in applying supervised topic modelling techniques to this domain. Furthemore, it would be interesting to introduce domain expertise into the models, whether by interactive topic modelling (Hu et al., 2014) or by providing informed priors, and seeing how that affects performance.

Another interesting direction we hope to explore is tracking changes amongst a user's posts over time. While we only used the four class labels, available sublables included "followupOk" for some amber posts and "followupWorse" for some red posts. Tracking how a user's language has changed both since the start of their time on the forum and from the start of a given thread seems likely to be able to provide useful features for classification of such cases.

Finally, the labeled data available for this task was rather limited, and while we used the unlabeled data in the creation of the topic models, our system in general focused on the labeled data. Future work might explore application of semi-supervised models, integrating both the unlabeled ReachOut data and mental health posts from other forums.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. Interactive topic modeling. *Machine learning*, 95(3):423–469.

Thomas R Insel. 2008. Assessing the economic costs of serious mental illness. *American Journal of Psychiatry*, 165(6).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Philip Resnik, William Armstrong, Leonardo Claudino, and Thang Nguyen. 2015. The University of Maryland CLPsych 2015 shared task system. *NAACL HLT 2015*, page 54.