Automatic Triage of Mental Health Forum Posts

Benjamin Shickel and **Parisa Rashidi**
University of Florida
Gainesville, FL
{shickelb, parisa.rashidi}@ufl.edu

Abstract

As part of the 2016 Computational Linguistics and Clinical Psychology (CLPsych) shared task, participants were asked to construct systems to automatically classify mental health forum posts into four categories, representing how urgently posts require moderator attention. This paper details the system implementation from the University of Florida, in which we compare several distinct models and show that best performance is achieved with domain-specific preprocessing, n-gram feature extraction, and cross-validated linear models.

1 Introduction

As more and more social interaction takes place online, the wealth of data provided by these online platforms is proving to be a useful source of information for identifying early warning signs for poor mental health. The goal of 2016 CLPsych shared task was to predict the degree of moderator attention required for posts on the ReachOut forum, an online youth mental health service that provides support to young people aged 14-25.[1]

Along with the analysis of forum-specific meta-information, this task includes aspects of sentiment analysis, the field of study that analyzes people's opinions, sentiments, attitudes, and emotions from written language (Liu, 2012), where several studies have explored the categorization and prediction of user sentiment in social media platforms such as Twitter (Agarwal et al., 2011; Kouloumpis et al., 2011; Spencer and Uchyigit, 2012; Zhang et al., 2011). Other studies have also applied sentiment analysis techniques to MOOC discussion forums (Wen et al., 2014) and suicide notes (Pestian et al., 2012), both highly relevant to this shared task.

Our straightforward approach draws from successful text classification and sentiment analysis methods, including the use of a sentiment lexicon (Liu, 2010) and Word2Vec distributed word embeddings (Mikolov et al., 2013), along with more traditional methods such as normalized n-gram counts. We utilize these linguistic features, as well as several hand-crafted features derived from the meta-information of posts and their authors, to construct logistic regression classifiers for predicting the status label of ReachOut forum posts.

2 Dataset

As part of the shared task, participants were provided a collection of ReachOut forum posts from July 2012 to June 2015. In addition to the textual post content, posts also contained meta-information such as author ID, author rank/affiliation, post time, thread ID, etc. A training set of 947 such posts was provided, each with a corresponding moderator attention label (*green*, *amber*, *red*, or *crisis*). An additional 65,024 unlabeled posts was also provided. The test set consisted of 241 unlabeled forum posts.

3 System

In this section, we describe the implementation details for our classification system. In short, our relatively straightforward approach involves selecting and extracting heterogenrous sets of features for

[1] https://au.reachout.com/

Proceedings of the 3rd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 188–192,
San Diego, California, June 16, 2016.

Name	Type	Description
View Count	Numeric	The number of times the post was viewed.
Kudos Count	Numeric	The number of kudos given to the post.
Reply Count	Numeric	The number of posts which were made in reply to the current post.
# Replying Authors	Numeric	The number of unique authors replying to the current post.
Board Name	Categorical	Which of the 25 subforums (boards) the post was made in.
Reply Status	Binary	Whether the current post is a reply or a new post.
Thread Size	Numeric	The number of total posts involved in the current post's thread.
Sibling Count	Numeric	The number of *other* posts replying to the same post that the current post is replying to.
Total Post Count	Numeric	The total number of posts made by the current author.
Total View Count	Numeric	The total number of views for posts made by the current author.
Total Kudos Count	Numeric	The total number of kudos given to posts created by the current author.
Mean View Count	Numeric	The average number of views for posts created by the current author.
Mean Kudos Count	Numeric	The mean number of kudos given to posts created by the current author.
Rank	Categorical	The forum "ranking" of the current author.
Affiliation	Binary	Whether the current author is a member of the ReachOut forum staff.
Board Fraction	Numeric	The fraction of the current author's total posts that were made in the current post's subforum.

Table 1: List of attributes extracted for each post. The upper half of the table contains attributes unique to the post itself, while the lower half contains attributes derived from the post's author.

each post, which are then used to train separate logistic regression classifiers for predicting the moderator attention label. We report results for each model individually, and experiment with various classifier ensembles. Results were obtained following a randomized hyperparameter search and 10-fold cross-validation process.

For clarity, we subdivide our features into two categories: post attributes and text-based features. We only extracted features for the 947 posts in the labeled training set; however, several of our features were historical in nature, utilizing information from the entirety of the unlabeled dataset of 65,024 posts.

3.1 Attribute Features

As a starting point for classifying posts as *green*, *amber*, *red*, or *crisis*, we began by examining several attributes of each post and its corresponding author.

Many of our attribute features were immediately available from the raw dataset, and required no further processing. A small sample of these statistics include the post's view count, kudos count, author rank, and in which subforum the post is located.

We also incorporated historical attributes that were derived from the entirety of the unlabeled dataset. These include items such as thread size, mean author kudos/views, number of unique reply authors, etc. Our full list of post attributes is shown in Table 1.

3.2 Text Features

Each post in the dataset was associated with two sources of free text - the subject line and the body content. Since the post content itself is what moderators themselves look to when deciding whether action should be taken, we speculated that these features were of the greatest importance. We applied several text-based feature extraction techniques, and began with an in-depth preprocessing phase.

3.2.1 Preprocessing

Since the textual information of each post was formatted as raw HTML, our first preprocessing step involved converting the post content to plain text. During this process, we replaced all user mentions (i.e., @user) with a special string token. We also built a map of all embedded images, of which the majority were forum-specific emoticons, and replaced occurrences in the text with special tokens denoting which image was used. We performed a similar technique for links, replacing each one with a special link identifier token. Finally, in an effort to reduce noise in the text, we removed all text contained within <BLOCKQUOTE> tags, which typically contained text that a post is replying to. After

these conversions, we stripped all remaining HTML tags from each post, resulting in plain-text subject and body content.

While examining the corpus, we also noticed the frequent presence of text-based emoticons, such as ':)' and '=('. We employed the use of an emoticon sentiment lexicon[2], which maps text-based emoticons to either a positive or negative sentiment, to convert each textual emoticon to one of two special tokens denoting the corresponding emoticon's polarity. We manually annotated 12 additional emoticons that were not present in the pre-existing lexicon.

Since we found the subject and body text to be highly related, we concatenated these texts into a single string per post. In an effort to further reduce noise in the text, we examined the subject line of each post, and if it was of the form "Re: ..." and contained the same subject text of the post it was replying to, we discarded the subject line.

Finally, we finished our preprocessing phase with several traditional techniques, including converting all text to lowercase and removing all punctuation. We also converted non-unicode symbols to their best approximation. Due to experimental feedback, we did not remove traditional stop words, as doing so decreased classifier performance for this domain.

3.2.2 N-Gram Features

The majority of our text features are derived from traditional n-gram extraction methods. Given the large amount of unlabeled posts in the dataset, we trained our text vectorizers on the entire corpus (minus the test set posts). After constructing a vocabulary of n-grams occurring in the corpus, we counted the number of each n-gram occurring in each post's text, and normalized them by term-frequency inverse-document frequency (tf-idf). Following initial feedback, our n-gram methods employed normalized unigram counts.

3.2.3 Sentiment Lexicon Features

Because a primary goal of the shared task was to gauge the mental state of posting authors, we borrowed a basic technique from sentiment analysis and utilized a pre-existing sentiment lexicon[3], which contains a list of words annotated as *positive* or *negative*. We count the number of occurrences of both *positive* and *negative* words in the text of each post.

3.2.4 Embedding Features

Since the amount of unlabeled text was so large relative to the labeled posts, we sought to learn a basic language model from past forum discussions. Our word embedding features are based on the recent success of Word2Vec[4] (Mikolov et al., 2013), a method for representing indidivual words as distributed vectors. Our specific implementation utilized Doc2Vec[5] (Le and Mikolov, 2014), a related method for computing distributed representations of entire documents. Our model used an embedding vector size of 400 and a window size of 4. After training the Doc2Vec model on the entire corpus of post text (minus test posts), we computed a 400-dimensional vector for the text of each training post.

3.2.5 Topic Modeling Features

As a final measure to incorporate the abundance of unlabeled text in the dataset, we trained a custom Latent Dirichlet Allocation (LDA) (Blei et al., 2003) model with 20 topics on the entire corpus of post text (minus test posts). LDA is a popular topic modeling technique which groups words into distinct topics, assigning both word-topic and topic-document probabilities. Once trained, we used our LDA model to predict a topic distribution (i.e, a 20-dimensional vector) for the text of each post.

4 Results

After extracting features for each of the 947 posts in the training set, we trained a separate logistic regression classifier on each source of text features, plus one trained on all of the attribute-based features. Because we hypothesized that the content of the replies to a particular post could be indicative of the nature of the post itself, for each set of text features we trained an additional model on the concatenated text of all direct reply posts only, ignoring the text of the post itself.

For each model, we performed a randomized hyperparameter search in conjunction with a 10-fold cross-validation step based on macro-averaged F1

[2]http://people.few.eur.nl/hogenboom/files/EmoticonSentimentLexicon.zip

[3]https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

[4]https://code.google.com/archive/p/word2vec

[5]https://radimrehurek.com/gensim/models/doc2vec.html

Feature Set	Accuracy	F1	Green vs. Non-Green Accuracy	Green vs. Non-Green F1
Post Attributes	0.76	0.72	0.78	0.66
Sentiment Lexicon	0.71	0.64	0.76	0.64
N-Grams (Post)	**0.83**	**0.82**	**0.90**	**0.88**
N-Grams (Replies)	0.73	0.68	0.80	0.72
Doc2Vec (Post)	0.74	0.70	0.80	0.72
Doc2Vec (Replies)	0.72	0.65	0.76	0.62
LDA (Post)	0.73	0.67	0.78	0.70
LDA (Replies)	0.71	0.63	0.78	0.66

Table 2: Classification results on the test set using a single logistic regression model trained on each set of features. (Post) denotes features extracted from each post itself, while (Replies) indicates that features were extracted from only replies to the post.

Label	Precision	Recall	F1
Green	0.91	0.95	0.93
Amber	0.59	0.72	0.65
Red	0.90	0.33	0.49
Crisis	0.00	0.00	0.00
Average	**0.84**	**0.83**	**0.82**

Table 3: Detailed classification results for our final model. No *crisis* labels were predicted, resulting in metrics of 0.0; however, the test set only included a single *crisis* post. Average reported metrics consider the support of each label.

Green	Amber	Red	Crisis
<E0>	(@user)	worse	cant
awesome	phone	feeling	anymore
<E1>	anxious	<E2>	life
hope	talk	empty	dont
love	not	sick	screwed
proud	school	hate	negative
amazing	think	family	f**k
fun	going	hospital	unsafe
favourite	help	scared	intense
first	feeling	s**t	die

Table 4: Top 10 features per label via the largest per-class feature coefficients of our final model. From an informal inspection, there appears to be a clear trend in the polarity of the word lists from *green* posts to *crisis* posts. **Notation:** <E0> = emoticon with alt text 'Smiley Happy', <E1> = emoticon with alt text 'Smiley Very Happy', <E2> = emoticon with alt text 'Smiley Sad', (@user) = special token for any user mention.

score. Results for each feature set are shown in Table 2, where it is clear that the model trained on n-grams of the post text (subject + body) performs the best across all metrics. We show a more detailed breakdown of this model's performance in Table 3, which includes per-label metrics.

4.1 Discussion

Given the relatively small amount of labeled data, it comes as no surprise that the traditional n-gram approach performs better than the more complex text-based methods. Because our vectorizers and vocabulary were trained on the full corpus of unlabeled and training posts before fine-tuning predictions on the test posts, this model is able to capture trends in word usage across all four labels.

We sought to combine the models shown in Table 2 with various ensemble methods, but found that no combination of classifiers trained on heterogeneous feature sets produced better results than the straightforward n-gram technique. Thus, the simplest text-based method proved also to have the best performance, a benefit for deploying such a system.

To gain better insight into our best-performing model, we show the top 10 features per label in Table 4, obtained by inspecting the model coefficients of the fully-trained logistic regression classifier. Here (aside from the *Amber* label, which is a bit more ambiguous, as expected), there is a clear distinction and trend in the type of language used between posts of different labels.

5 Conclusion

In this paper, we detailed our system implementation for the CLPsych 2016 shared task. We compared several types of models and feature sets, and showed the benefit of combining rigorous preprocessing with straightforward n-gram feature extraction and a simple linear classifier. Additionally, using the entire corpus of forum text, we identified several discriminative features that can serve as a launching point for future studies.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of Twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38.

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

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 11)*, pages 538–541.

Qv Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1188–1196.

Bing Liu. 2010. *Sentiment Analysis and Subjectivity*. 2 edition.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations (ICLR)*, pages 1–12.

John Pestian, John Pestian, Pawel Matykiewicz, Brett South, Ozlem Uzuner, and John Hurdle. 2012. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5(1):3–16.

James Spencer and Gulden Uchyigit. 2012. Sentimentor: Sentiment analysis of Twitter data. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 56–66.

Miaomiao Wen, Diyi Yang, and Cp Rosé. 2014. Sentiment analysis in MOOC discussion forums: What does it tell us? In *Proceedings of Educational Data Mining*, pages 1–8.

Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. 2011. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Technical report.